SPERRY UNIVAC
COMPUTER SYSTEMS

PUBLICATIONS RELEASE

Operating System/3 (OS/3)

Menu Services

Concepts and Facilities

UP-9317

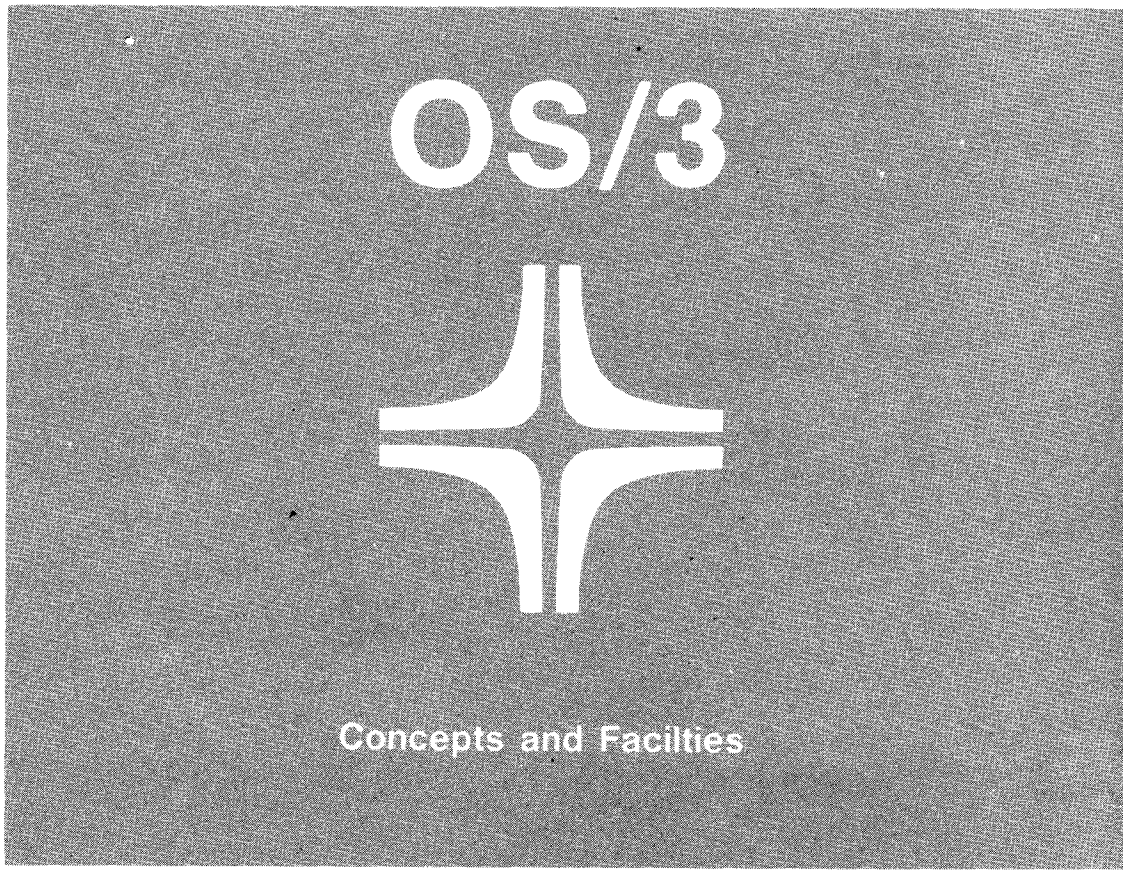
This Library Memo announces the release and availability of "SPERRY UNIVAC® Operating System/3 (OS/3) Menu Services Concepts and Facilities", UP-9317.

This is the initial release of the menu services concepts and facilities manual. It is intended for the OS/3 programmer and system administrator. It discusses the concepts behind menu use; how to create, modify, or display menus using the menu generator; and how to integrate menus with user programs.

Additional copies may be ordered by your local Sperry Univac representative.

LIBRARY MEMO ONLY	LIBRARY MEMO AND ATTACHMENTS	THIS SHEET IS
Mailing Lists BZ, CZ and MZ	Mailing Lists A00,A30,B00,B30,18,18U,19,19U,20,20U, 21,21U,28U,29U,75,75U,76 and 76U (Cover and 135 pages)	Library Memo for UP-9317
		RELEASE DATE: September, 1982

Menu Services



This document contains the latest information available at the time of preparation. Therefore, it may contain descriptions of functions not implemented at manual distribution time. To ensure that you have the latest information regarding levels of implementation and functional availability, please consult the appropriate release documentation or contact your local Sperry Univac representative.

Sperry Univac reserves the right to modify or revise the content of this document. No contractual obligation by Sperry Univac regarding level, scope, or timing of functional implementation is either expressed or implied in this document. It is further understood that in consideration of the receipt or purchase of this document, the recipient or purchaser agrees not to reproduce or copy it by any means whatsoever, nor to permit such action by others, for any purpose without prior written permission from Sperry Univac.

Sperry Univac is a division of the Sperry Corporation.

This document was prepared by Systems Publications using the SPERRY UNIVAC UTS 400 Text Editor. It was printed and distributed by the Customer Information Distribution Center (CIDC), 555 Henderson Rd., King of Prussia, Pa., 19406.

PRINTED IN U.S.A.

PAGE STATUS SUMMARY

ISSUE: UP-9317
RELEASE LEVEL: 8.0 Forward

Part/Section	Page Number	Update Level
Cover/Disclaimer		
PSS	1	
Preface	1, 2	
Contents	1 thru 4	
1	1 thru 8	
2	1 thru 17	
3	1, 2 2a 3 thru 29	
4	1, 2 2a 3, 4 4a 5 thru 10 10a 11 thru 14 14a 15 thru 19	
5	1, 2 2a 3 thru 6	
6	1 thru 17	
Appendix A	1 thru 4	
Appendix B	1, 2	
Appendix C	1 thru 13	
Index	1 thru 5	
User Comment Sheet		

Part/Section	Page Number	Update Level

Part/Section	Page Number	Update Level

All the technical changes are denoted by an arrow (→) in the margin. A downward pointing arrow (↓) next to a line indicates that technical changes begin at this line and continue until an upward pointing arrow (↑) is found. A horizontal arrow (→) pointing to a line indicates a technical change in only that line. A horizontal arrow located between two consecutive lines indicates technical changes in both lines or deletions.

Preface

This manual is part of the Sperry Univac library dealing with Operating System/3 (OS/3). It describes the concept behind OS/3 workstation menus, as well as how to create and process them.

This manual is intended for programmers and system administrators who are interested in expanding the range of menu services by creating their own. For programmers, this manual explains how to create and maintain menus and integrate them with other user software, most notably job control streams and user programs written in COBOL, FORTRAN, RPG II, and the basic assembly language (BAL). For the system administrator, the manual shows how menus can be used to train operators and promote program and system security. We explain how to use menus in an appendix to this manual, but that information is of use mostly to workstation operators and consequently can be found in other manuals.

Menu services have been designed for ease of use. Apart from a general knowledge of OS/3 programming, you only need to be familiar with job control and interactive commands. Some knowledge of screen format use and creation is helpful, but not absolutely necessary.

Throughout this manual, we refer to other manuals in the OS/3 library. These include the current versions of:

- Interactive services commands concepts and facilities, UP-8845

  Explains all the workstation commands. In particular, it explains how to call menus from a workstation in system mode. This is the publication to which workstation operators most commonly refer.

- Screen format services concepts and facilities, UP-8802

  Explains in detail how the screen format generator can create and modify workstation screen formats. This, in turn, explains some of the features of menu services and shows how to create screen formats for use in your menus.

- Job control user guide, UP-8065

  Explains how to create and modify job control streams.

- Consolidated data management concepts and facilities, UP-8825

  Explains how consolidated data management handles input to a user program. This and the four language manuals that follow provide background information on integrating menus with user programs.

- Consolidated data management macroinstructions user guide, UP-8826

  Gives a detailed explanation of input/output for user programs written in BAL.

- Report Program Generator (RPG II) user guide/programmer reference, UP-8067

  Gives a detailed explanation of input/output for user programs written in RPG II.

- FORTRAN IV programmer reference, UP-8474

  Gives a detailed explanation of input/output for user programs written in FORTRAN IV.

- 1974 ANS COBOL programmer reference, UP-8613

  Gives a detailed explanation of input/output for user programs written in COBOL.

Contents

3. CREATING A MENU

4. MODIFYING A MENU

5. DISPLAYING A MENU

6. USING MENUS WITH USER PROGRAMS

APPENDIXES

FIGURES

TABLES

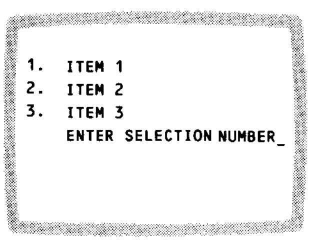
1. Introduction

1.1. MENUS – AIDS TO WORKSTATION OPERATORS

Menus – a tool for familiarizing operators with OS/3

If your OS/3 system uses workstations, you know that an operator works faster and with more accuracy when he is familiar with the tasks he performs on his workstation. Familiarity comes most easily when the operator can relate workstation use to other everyday tasks, and it comes fastest when the system itself can teach a new operator or help a more experienced operator brush up on his work. OS/3 provides several aids to familiarize operators with workstation use, and among these are *menus.*

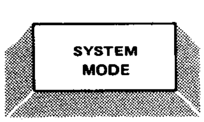
List of items



Like any other kind of menu, OS/3 menus are lists of items from which to choose; usually these represent workstation commands, programs, or functions within programs. Menu items are numbered so that when the system displays a menu on the operator's workstation, he selects an item by entering its item number in the blanks provided on the menu screen and pressing the XMIT key. (On some terminals, this key is marked TRANSMIT. Throughout this manual, we'll use XMIT to refer to either the XMIT or TRANSMIT key.)

Menus are *interactive:* that is, a menu displays a screen of data and then waits for the operator to make a selection. Then, depending on the response, the menu may call a job, execute an interactive command, or even call another menu. In most cases, the menu eventually returns to the screen and asks the operator to make another selection.

How to call menus



The operator can use menus on a workstation either in system mode or with a user program in workstation mode. When the workstation is in system mode, the operator calls a menu with this command:

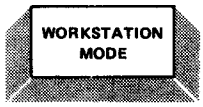
```
MENU [ [menuname] [ , { filename,vsn              } ] ]
                        { SYSFMT [ , { RES } ]    }
                        {            { vsn }      }
```

This command has three optional positional parameters:

`menuname`
: Specifies the name of a menu the workstation operator wants to call. The operator may omit this parameter to get a standard system menu, supplied by Sperry Univac, that acquaints new operators with OS/3; if he omits this parameter, he must also omit the two following parameters.

`filename`
: Specifies the name of MIRAM library file containing the menu to be called. If the operator omits this parameter, the file is assumed to be system format file YFMT.

`vsn`
: Specifies the volume serial number of the disk on which the menu resides. The default value, the system resident volume RES, applies only if the filename parameter is YFMT. For any other file name, the operator must specify this parameter even if it is RES.



Besides calling menus directly in system mode, a workstation operator can call them indirectly from a user program when the workstation is in workstation mode. In this mode, a menu is treated like an input/output device, such as a diskette drive or card reader. Menus appear the same in either mode, but a menu written for use with a user program is not usually called from system mode, and vice versa.

A payroll menu example

Let's see how a workstation operator uses a menu. We'll assume that the operator maintains a payroll system with such common functions as printing payroll checks, adding new employees to the payroll, backing up and restoring payroll file, and so on. For this type of application, the system may display, for the operator's use, a menu like the one in Figure 1–1.

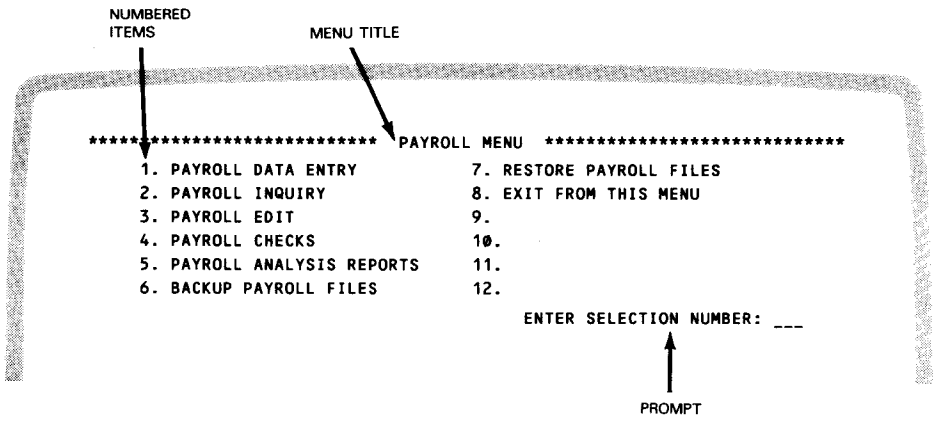


Figure 1–1. A PAYROLL Menu

Parts of a menu

We've labeled the various parts of the menu in Figure 1-1. These include:

- The menu title
- One or more numbered items, each a short narrative describing what happens if it is chosen
- A prompt, which is a 1- to 3-character field into which the operator types his response to the menu. The response can take one of these forms:
  - **n** - which means the operator selects item #n
  - **?n** - which displays a help screen (Figure 1-2) for item #n
  - **?** - which displays a help screen for the menu as a whole

  After typing in a response, the operator must press XMIT to actually send it to the system.

Choosing an item

An experienced operator who knows, for example, that he wants to get information from the payroll file by using PAYROLL INQUIRY can do so simply by entering the number of that item - 2 - in the underscored part of the menu after ENTER SELECTION NUMBER. After entering the item number, he then presses XMIT:

```
**************************** PAYROLL MENU ****************************
     1. PAYROLL DATA ENTRY          7. RESTORE PAYROLL FILES
     2. PAYROLL INQUIRY             8. EXIT FROM THIS MENU
     3. PAYROLL EDIT                9.
     4. PAYROLL CHECKS              10.
     5. PAYROLL ANALYSIS REPORTS    11.
     6. BACKUP PAYROLL FILES        12.
                                  ENTER SELECTION NUMBER: 2__
```

XMIT

(PAYROLL INQUIRY begins.)

We've just seen that by pressing two keys, the number *2* and the XMIT key, the operator can call whatever function on the menu he wants to run. After PAYROLL INQUIRY finishes, control of the workstation returns to the menu so that the operator sees this screen once again:

```
**************************** PAYROLL MENU ****************************
     1. PAYROLL DATA ENTRY          7. RESTORE PAYROLL FILES
     2. PAYROLL INQUIRY             8. EXIT FROM THIS MENU
     3. PAYROLL EDIT                9.
     4. PAYROLL CHECKS              10.
     5. PAYROLL ANALYSIS REPORTS    11.
     6. BACKUP PAYROLL FILES        12.
                                    ENTER SELECTION NUMBER: ___
```

Calling a help screen

There's more to using menus than just calling programs. An inexperienced operator who's unsure, say, about whether to call BACKUP PAYROLL FILES or RESTORE PAYROLL FILES can get quick assistance from the system itself by means of *help screens*. To get more information about BACKUP PAYROLL FILES, for instance, the operator enters a question mark followed by the item number for that function:

```
**************************** PAYROLL MENU ****************************
     1. PAYROLL DATA ENTRY          7. RESTORE PAYROLL FILES
     2. PAYROLL INQUIRY             8. EXIT FROM THIS MENU
     3. PAYROLL EDIT                9.
     4. PAYROLL CHECKS              10.
     5. PAYROLL ANALYSIS REPORTS    11.
     6. BACKUP PAYROLL FILES        12.
                                    ENTER SELECTION NUMBER: ?6_
```

Upon pressing the XMIT key, the operator gets a workstation screen that looks something like Figure 1–2.

```
                        PAYROLL BACKUP HELP SCREEN
THIS PAYROLL FUNCTION COPIES ALL PAYROLL DISK FILES ONTO TAPE TO GUARD AGAINST
THE ACCIDENTAL LOSS OF VITAL PAYROLL DATA. WHEN YOU RUN THIS FUNCTION, THE
SYSTEM ASKS YOU TO MOUNT THE TAPE VOLUMES THAT WILL RECEIVE THE PAYROLL DATA.
IF SOMETHING DOES HAPPEN TO ERASE THE MOST CURRENT PAYROLL DATA ON THE SYSTEM,
YOU CAN USE THE PAYROLL RESTORE FUNCTION TO COPY THE TAPE DATA BACK TO THE
SYSTEM.
                                                PRESS XMIT TO CONTINUE -
```

Figure 1–2. A Help Screen

The help screen in Figure 1-2 contains information about item 6. When the operator finishes reading the screen, he signals the system by pressing the XMIT key. The menu then reappears, ready for another entry. If he wishes, the operator can ask to see the help screen for RESTORE PAYROLL FILES or actually run the payroll function that he wants to run. (Sometimes help for an item appears as a sequence of help screens. After reading each help screen, the operator presses XMIT to call the next screen.)

A summary help screen

An even more inexperienced operator, uncertain about using the menu itself, can call a summary help screen to describe the entire menu. He does this by entering a question mark by itself – with no item number – in the prompt:

```
**************************** PAYROLL MENU ****************************
     1. PAYROLL DATA;ENTRY          7. RESTORE PAYROLL FILES
     2. PAYROLL INQUIRY             8. EXIT FROM THIS MENU
     3. PAYROLL EDIT                9.
     4. PAYROLL CHECKS              10.
     5. PAYROLL ANALYSIS REPORTS    11.
     6. BACKUP PAYROLL FILES        12.
                                  ENTER SELECTION NUMBER: ?__
```

Upon pressing XMIT, the operator gets a help screen that may look like Figure 1-3.

```
                        PAYROLL MENU HELP SCREEN
THIS MENU LISTS SEVERAL PAYROLL-RELATED TASKS. CHOOSE THE TASK YOU WANT TO RUN,
ENTER ITS NUMBER IN THE PROMPT (THE AREA AFTER ''ENTER SELECTION NUMBER''), AND
PRESS THE XMIT KEY. AFTER THE TASK FINISHES RUNNING, THIS MENU WILL REAPPEAR.
IF YOU HAVE ANY QUESTIONS ABOUT A SPECIFIC MENU ITEM, ENTER A QUESTION MARK
FOLLOWED BY THE ITEM NUMBER AND PRESS XMIT. TO GET THIS SCREEN, SIMPLY ENTER A
QUESTION MARK THEN PRESS XMIT.
                                  PRESS XMIT TO CONTINUE -
```

Figure 1-3. A Summary Help Screen

Just as with the help screen in Figure 1-2, the operator presses XMIT to signal when he's finished reading this screen, causing the menu to reappear. In fact, the only difference between the help screens in Figures 1-2 and 1-3 is how and when the operator calls each. Figure 1-2 describes only item 6 in the PAYROLL menu. Since there may be help screens for every other item in the menu too, the operator has to specify for which item number he wants help (?6 in Figure 1-2, for example). On the other hand, the menu has a single summary help screen to explain the menu as a whole, called simply by entering "?" in the prompt.

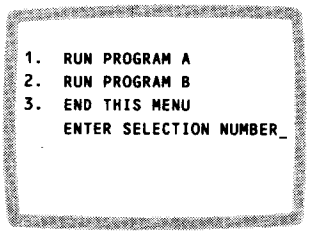
Leaving a menu

It may seem that we've locked the workstation operator into an endless loop because the menu reappears after every item or help screen finishes. We haven't. In fact, we've provided the operator with an exit from the loop in item 8, EXIT FROM THIS MENU. If a user program called this menu, as is likely, choosing this item causes the menu to disappear and gives workstation control back to the program. (And as an alternative to items like item 8, the operator can press function key F15 to exit from a menu at any time, whether the menu was called by a program or by the MENU command.)

Menu features

You can see from our PAYROLL menu example that menu services offer you and your workstation operators these features:

- You can simplify the procedure by which workstation operators run jobs. This means less confusion, especially for operators not yet acquainted with OS/3 job control.
- You can structure the tasks workstation operators do by listing them on menus. This way, operators perform only those tasks they need to, which can also aid in your system's security.
- With menus, the system itself can teach workstation operators the tasks they are to perform. The combination of multiple-choice menus, help screens, and the system's ability to keep track of several workstations simultaneously allows each operator to proceed at his own pace. By shifting much of the burden of teaching new operators to the system, menu services also free supervisors from having to deal with every operator question as it arises.



To summarize, then, a menu makes a workstation screen act like menus found in everyday life. It lists on a single screen all the actions available to the operator. Operator responses need only a few keystrokes. If the operator needs to know more about the menu or any of its items, it furnishes help screens for on-the-spot information.

You've seen how easy it is to use a menu. In the sections to follow, we'll show you how to create menus and how to integrate them into your other programs and jobs.

1.2. WORKSTATION DISPLAY CONVENTIONS

Throughout this manual, we display menus and other workstation screens in this format:

Menus and screens

SCREEN 3-3

```
                                                          MG0101S
                    MENU MODULE OUTPUT INFORMATION
WHAT NAME DO YOU WANT TO GIVE THE MENU MODULE?  ENTER BELOW
MENU MODULE NAME: ________

WHAT FILE LBL SHOULD THE MENU MODULE BE WRITTEN TO?  ENTER BELOW
FILE LBL: ____________________________________________

WHAT VSN SHOULD THE MENU MODULE BE WRITTEN TO?  ENTER BELOW
VSN: ______
```

When we want to show you where and how to enter data on a workstation screen, we use **reverse print**:

Data entry

SCREEN 4-12

```
                                                          MG2202S
ENTER ITEM NUMBER WHICH CORRESPONDS TO ACTION(S) TO BE ADDED.
ENTER ITEM #: 2
```

THIS IS AN ENTRY YOU WOULD MAKE.

Sometimes we want to highlight some part of a screen; one way we do this is with **shading**:

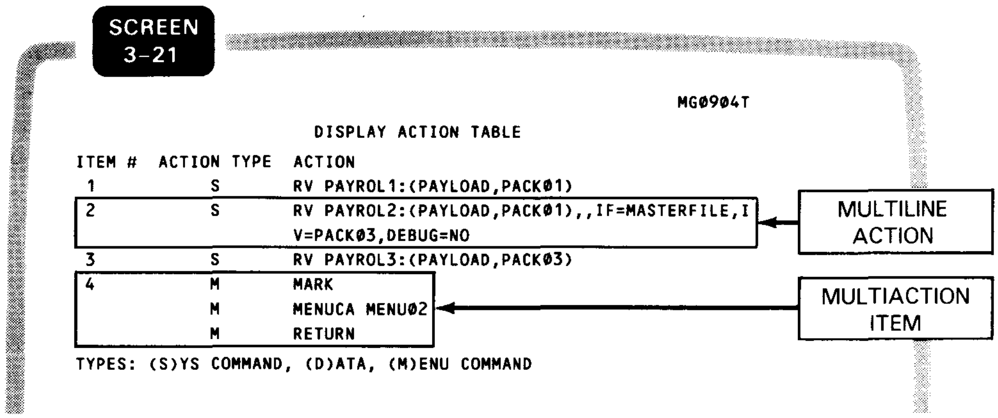
Highlighting by shading

SCREEN 3-23

```
                                                          MG0401S
                CREATE ACTION TABLE FOR EXPERIENCED USER
ITEM #   ACTION TYPE  ACTION (IF GREATER THAN 44 CHARACTERS
(1-99) (S, D, OR M)         CONTINUE IN NEXT ACTION FIELD.)
 --          -       ---------------------------------------------
 --          -       ---------------------------------------------
 --          -       ---------------------------------------------
 --          -       ---------------------------------------------
 --          -       ---------------------------------------------
 --          -       ---------------------------------------------
 --          -       ---------------------------------------------
 --          -       ---------------------------------------------
```

Another way of highlighting is with boxes and callouts:

Highlighting with boxes



2. Making Menus Work for You

Now that you've seen how to use menus from a workstation and some of the advantages of menu use, we'll begin to show you how to create your own menus. You create menus interactively with the menu generator (MENUGEN). We'll show you how to use MENUGEN starting with Section 3. But first, we'll show you how OS/3 handles menus, how menus interact with other menus, and how to make menus do what you want them to do.

2.1. HOW DOES OS/3 HANDLE MENUS?

Comparing menus with user programs

To see how menus work, we can compare them with user programs. Both consist of sequences of instructions and related data that make the system carry out operations according to the wishes of a programmer. Carrying the comparison a little further, remember that a user program usually resides in a disk library file as a load module generated by a language processor (such as COBOL or RPG II) and the linkage editor. When run, it's loaded into main storage, from which it carries out its instructions through the processor hardware. The sum total of these instructions and data is a program that does a job, such as printing a report or updating a data file. Figure 2-1 summarizes the process.

Programs and load modules

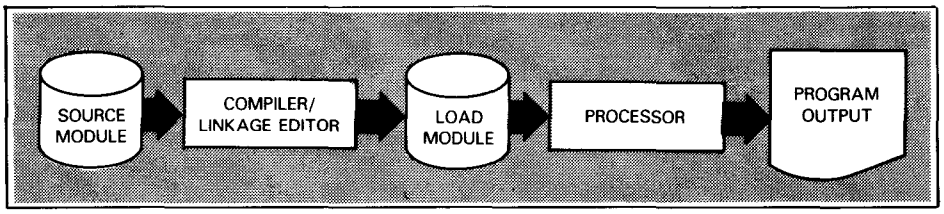


Figure 2-1. Creating and Using a Program

Menus and menu modules

Now think of a menu as a module created by MENUGEN and stored in another disk library file, as in Figure 2–2.

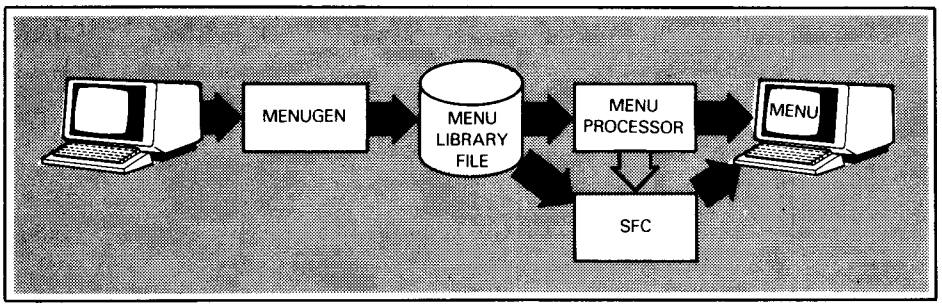


Figure 2–2. Creating and Using a Menu

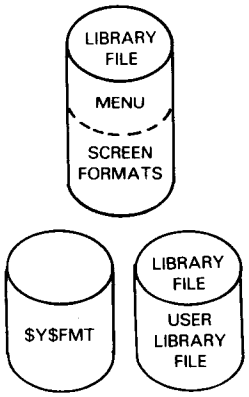
As we've said, MENUGEN creates and stores a menu as a module in its library file. When called, a menu is loaded into main storage and given control of the menu processor, a system program that displays menu screens, accepts operator replies, and acts on them according to the instructions contained in the menu.

The differences between programs and menus

There are differences between programs and menus. Unlike the processor in Figure 2–1, the menu processor is a software component, one that is capable of handling menus from more than one workstation simultaneously. Menus are more static than programs because a menu screen stays on a workstation until the operator makes a response, however long it takes. And while some menus work with user programs, many others work with interactive services, independent of any program.

The screen format coordinator

Another component in Figure 2–2 is the screen format coordinator (SFC). Some of the screens used in menus are stored in a library file as screen formats. When the menu processor needs one of these screens, it calls upon the SFC to retrieve the corresponding screen format module from the file and prepare it for display on the workstation.



For a menu, the same library file contains both the menu module and all associated screen formats. Most menus use the MIRAM system format file YFMT. You can maintain your own MIRAM file of menus, but with restrictions we'll explain when necessary.

2.2. MENU LEVELS - MAKING MENUS CALL OTHER MENUS

With menu services, you can move back and forth between menus. You may already have seen this if you've tried out the MENU workstation command. When you enter MENU (without any operand), you see screen 2-1:

SCREEN 2-1

```
                    SYSTEM MENU

          1. RUN A JOB                    3. END MENU
          2. PERFORM A SYSTEM FUNCTION    4. LOGOFF

FOR HELP ON A PARTICULAR ITEM NUMBER, ENTER A QUESTION MARK
FOLLOWED BY THE ITEM NUMBER (?N). HELP FOR THE ENTIRE DISPLAY
CAN BE ACQUIRED BY ENTERING A QUESTION MARK (?) THEN PRESSING
TRANSMIT.

                                ENTER SELECTION NUMBER: ___
```

Selecting item 2, you get a second menu screen, screen 2-2:

SCREEN 2-2

```
              SYSTEM FUNCTION MENU                    OS314

1. *DATA UTILITIES                    7. *EDITOR
2. *BASIC                             8. JCL DIALOG
3. *SCREEN FORMAT GENERATOR           9. SYSGEN DIALOG
4. *MENU GENERATOR                   10. HARDWARE UTILITIES (HU)
5. *RPG EDIT                         11. RETURN TO SYSTEM MENU
6. *DDP
* OPTIONAL SOFTWARE. CONTACT YOUR SPERRY UNIVAC MARKETING OFFICE
                     ENTER SELECTION NUMBER:___
```

Most of these items call system functions and utilities. But one exception is item 11. If you select that item, RETURN TO SYSTEM MENU, that's exactly what the menu processor will do by redisplaying the SYSTEM MENU on the screen:

```
                    SYSTEM MENU

          1. RUN A JOB                    3. END MENU
          2. PERFORM A SYSTEM FUNCTION    4. LOGOFF

FOR HELP ON A PARTICULAR ITEM NUMBER, ENTER A QUESTION MARK
FOLLOWED BY THE ITEM NUMBER (?N). HELP FOR THE ENTIRE DISPLAY
CAN BE ACQUIRED BY ENTERING A QUESTION MARK (?) THEN PRESSING
TRANSMIT.

                                ENTER SELECTION NUMBER: ___
```

Menu levels and chains

We've already said that the menu processor can handle menus from several different workstations at once. We now see that the menu processor can also handle more than one menu for any individual workstation. Of course, only one menu screen at a time can appear on the workstation. But a menu can call another menu, which then overlays it. From this, we can imagine building a *chain* of menus, one calling another, each occupying a menu *level* (Figure 2–3).

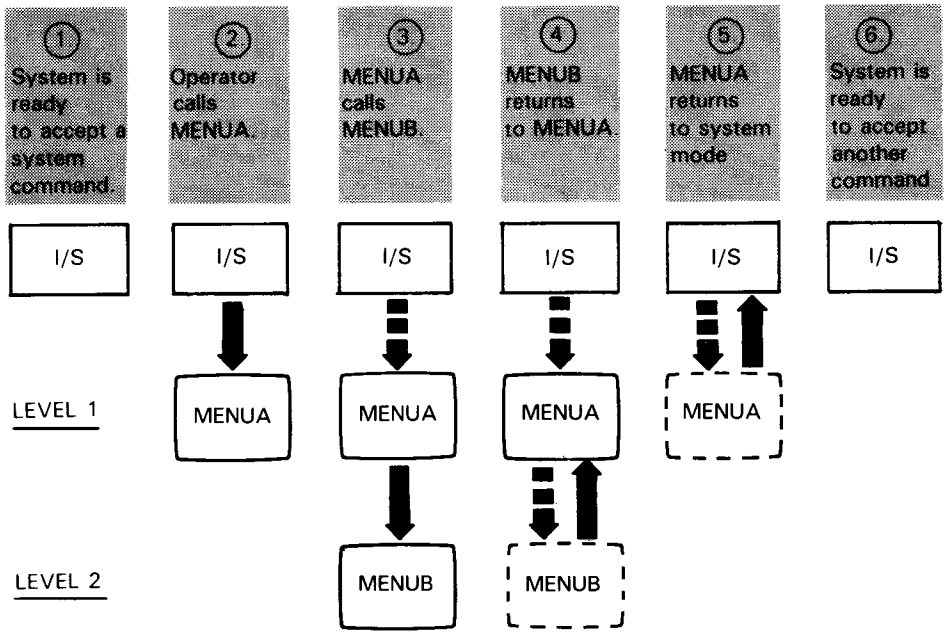


Figure 2–3. Chaining Menus Together

Assume that menus MENUA and MENUB reside in system library file YFMT. Figure 2–3 shows in six steps how you can create and use menu chains. Use these steps, numbered with the circled keys ① to ⑥, as you read through the following:

① The operator puts his workstation in system mode, preparing to call a menu.

② The operator calls MENUA by keying in MENU MENUA. MENUA now appears on his screen. From now until ⑤, the menu processor has control of the workstation.

③ MENUA calls MENUB. The MENUB screen now overlays MENUA, but MENUA remains in the chain, thanks to the menu processor. This makes step ④ possible.

④ MENUB returns to MENUA. The MENUA screen now overlays the MENUB screen, and MENUB disappears from the menu chain (dotted MENUB box).

⑤ MENUA returns to interactive services since it was a system command (MENU) that called MENUA in the first place. The MENUA screen disappears.

⑥ By putting the workstation in system mode, the operator can now enter other system commands.

The idea of menu levels becomes important when you establish menu chains. We established menu levels implicitly in Figure 2-3 by drawing MENUA in steps 2 to 5 always at the same level; likewise MENUB in steps 3 and 4. In the left margin of Figure 2-3, we make levels more explicit by naming them level 1 and level 2. Whenever one menu calls another, we'll show the connection by drawing the called menu below the calling menu. Sometimes we'll show level numbers too. We'll assign the first menu called from a user program or the MENU command to level 1, a menu called from level 1 to level 2, a menu called from level 2 to level 3, and so on, as necessary. There is no limit to the number of levels a menu chain may have.

On the way up or down the chain, each menu may provide the user with other options. In fact, a menu may be programmed to call any of several other menus, depending on the choice made. This lets you expand a menu chain into a menu tree, such as the one in Figure 2-4.

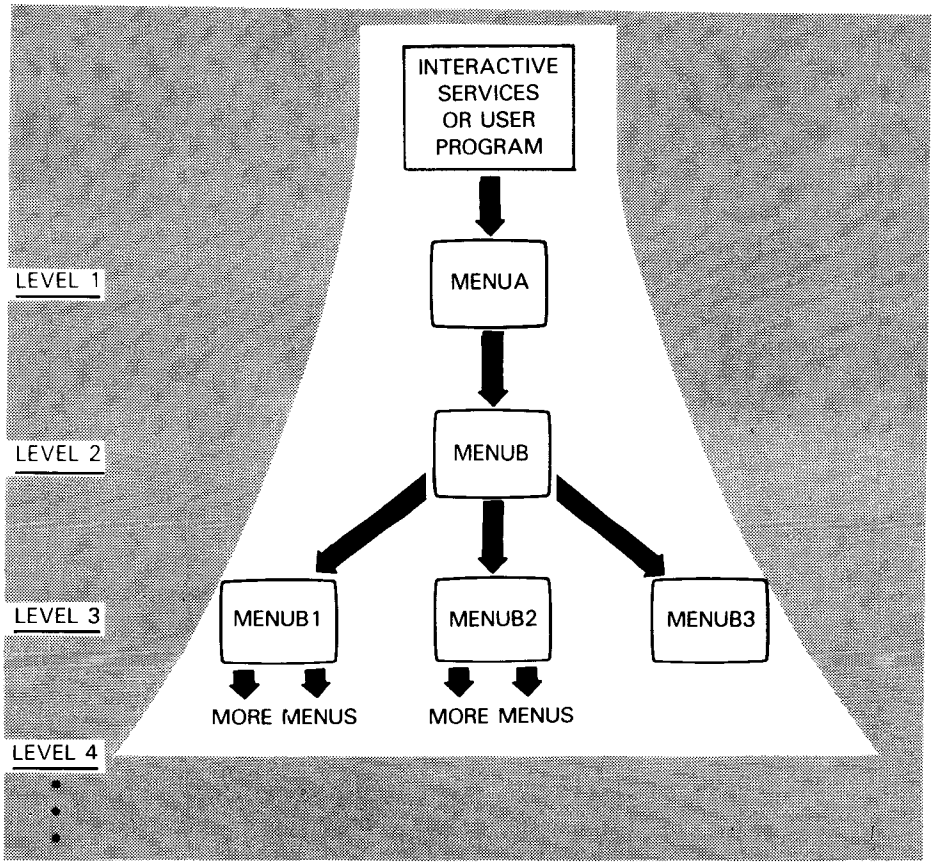


Figure 2-4. A Tree of Menus

We base Figure 2–4 on the 2-level menu chain in Figure 2–3. Here, we expand the chain to three levels by showing that MENUB can call any one of three other menus MENUB1, MENUB2, or MENUB3. Whichever is called, that menu occupies level 3. It may later return to MENUB just as MENUB returns to MENUA in Figure 2–3. Or it may in turn call a level 4 menu from among a number of menus available.

Before moving on to action tables, we should make one more point about menu chains: you don't create a menu explicitly as, say, a level 2 menu. It only becomes one because of the sequence of menus that have gone before it on the workstation. The same menu may be a level 1 menu in one chain and a level 3 menu in another chain. It's always a good idea to draw a menu chain first, like the ones in Figures 2–3 and 2–4, before creating any of its menus.

2.3. ACTION TABLES – MENU PROGRAMS

The action table – a collection of actions

Actions – how a menu carries out instructions

How are menus programmed to do what we've said they can do? The answer lies in a part of each menu that is hidden from the workstation operator – the action table. Loaded in main storage when a menu is called, the action table is a collection of actions that are linked to all the items in the menu. When you choose an item, the menu processor picks up that item's corresponding action or actions and executes them. If we compare a menu module to a user program load module the way we did in Figure 2–1, we can think of the action table as the menu's program, and its actions as individual program instructions.

Each menu has its own action table to which it is bound. Thus, when you go from one menu to another, control of the menu processor goes from the first menu's action table to the second menu's. But control information kept by the menu processor allows control to return to the first menu whenever it's called for.

Action types

An action can be one of three types:

System commands

- A *system command* like EDT, RV, or LOGOFF. Any workstation command that the operator can enter through the keyboard (up to 72 bytes long) can be an action. For this type of action, the menu processor sends the command to interactive services just as if the operator had entered it through the keyboard. For more information on system commands, their uses, and their formats, see the interactive services commands and facilities user guide/programmer reference (current version). You can use system commands in menus called either by the MENU command or by a user program.

Menu function commands

- A *menu function* command. These commands, unique to menu processing, enable menus to call each other and to carry out related functions. It is in response to these commands that the menu processor assigns menus to different levels and causes menu processing to jump from one menu level to another, changing menu screens in the process. These commands can be present in menus called either by a user program or by the MENU command. These are listed in 2.4.

Input data

- *Input data* for a user program. This type of action can be present only in menus called by a user program. In this case, the action is simply a string of data up to 72 bytes long that the menu processor sends to the input program every time its corresponding item is selected. From the point of view of the program itself, it "reads" the data just as if the menu were an input device, like a card reader or diskette drive. We'll tell you more about inputting data from menus in Section 6.

Linking action to menu items

Each action contains information telling the menu processor which item the action is associated with. When an item has more than one action associated with it, other information in the action tells the menu processor which action to take first, which second, and so on. Figure 2–5 shows how the menu processor uses this information when an operator makes a menu selection.

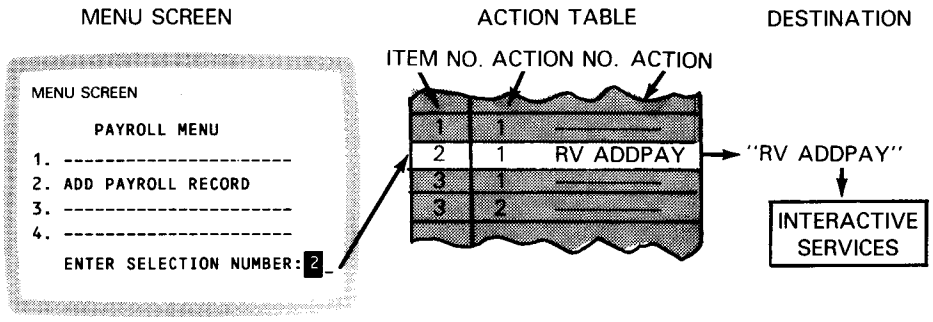


Figure 2–5. How the Action Table Works

When the operator chooses item 2 of PAYROLL MENU in Figure 2–5, the menu processor searches the action table for any action linked to item 2. Finding one (the command RV ADDPAY), the menu processor pulls it out of the action field and sends it to interactive services for execution.

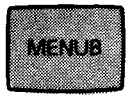
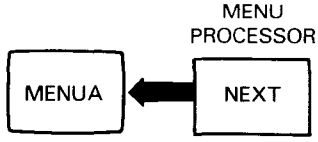
Multiple system commands Note that if there were a second RV command linked to item 2, the menu processor would send it immediately after the first RV. In fact, both of the jobs called by these commands could run at the same time. This illustrates an important point about certain system commands: If multiple commands are linked to a single item, the menu processor deals with them all, one after the other. It does not wait for one command to finish before continuing on to the next. Be careful therefore not to call two or more jobs using the same resource (such as a diskette drive) or to run two jobs if one depends entirely on the successful completion of the other.

NOTE:

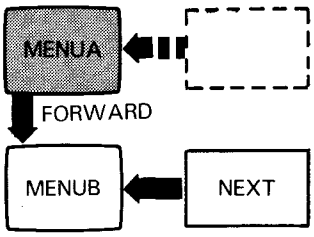
Multiple actions that are menu function commands are executed consecutively, not all at once like system commands. Each action is executed only when the preceding action has been fully processed.

2.4. MENU FUNCTION COMMANDS

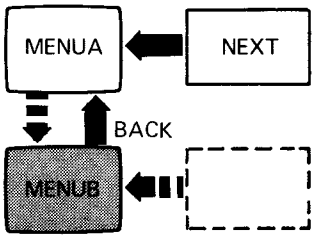
This subsection deals with menu function commands and their use. To explain them, we'll use some concepts that we've already presented, and some we haven't. Those we haven't are shown in the following list of symbols, each with an explanation next to it. When we get to explaining the menu commands, we'll assemble these symbols into more complex diagrams that should help make clear what each command does.

Symbol	Explanation
	The menu currently appearing on the workstation
	A menu in the menu chain, but not the one currently appearing on the workstation (It may also indicate a screen called from a menu.)
	The menu processor. It contains the next action to be executed and always points left to the current menu – here, MENUA. (Usually, the current menu is the one displayed on the operator's workstation, but not always; see the SCREEN and DISPLAY commands.)

(continued)



The action of one menu calling another. Here, MENUA calls MENUB. Note that the menu processor box moves from MENUA to MENUB. As a result, we say that control moves *forward* in the chain. The menu screen for MENUB overlays the MENUA screen. MENUA and MENUB are on different menu levels.



The action of one menu returning workstation control to another. Here, MENUB is returning control to MENUA. Note that the menu processor box moves from MENUB *back* to the menu that originally called it, MENUA. The menu screen for MENUA overlays the MENUB screen. MENUA and MENUB are on different menu levels.

Keeping these terms in mind, we can describe the menu function commands:

The MARK command

MARK Use this command to set an indicator in the current menu that is used as a guidepost by the RETURN command (see later discussion). The indicator remains set for the duration of menu processing. The MARK function has no other effect on the current menu. Figure 2-6 shows one example of this command.

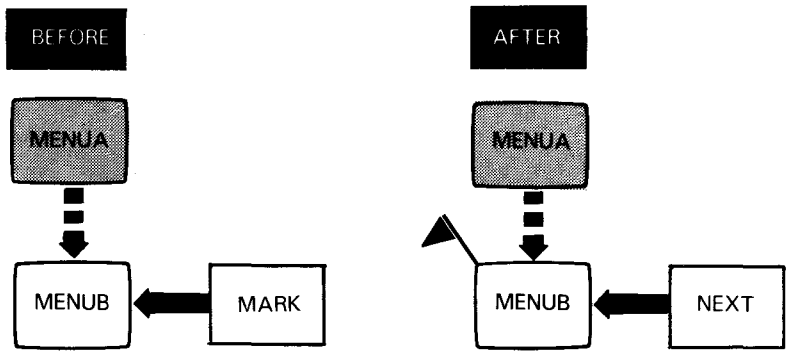


Figure 2-6. An Example of the MARK Command

The menu chain in Figure 2-6 first shows menu MENUB just after it's been called by MENUA and before it executes a MARK command. Upon execution of that command, an indicator within MENUB is set (small flag). The word *NEXT* inside the box indicates that the menu processor is about to process the next action after MARK.

The RETURN command

RETURN Use this command to move back through the menu chain. You may use it in tandem with a previously issued MARK command to backtrack to another menu in the chain, or by itself to end menu processing altogether. In response to this command, the menu processor leaves the current menu and travels back through the menu chain, skipping over intervening menus, until either of these events occurs:

- It finds another menu previously marked by the MARK command, as in Figure 2–7. In this case, the menu processor stops traveling and gives the marked menu control.

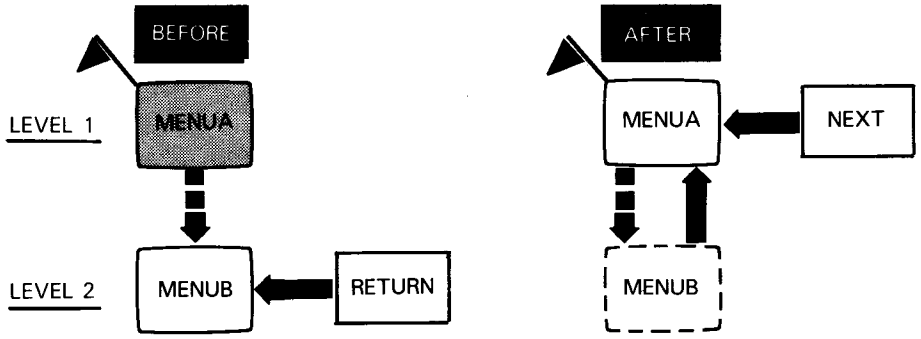


Figure 2–7. Example 1 of the RETURN Command

In Figure 2–7, the menu chain starts with two menus, MENUA and MENUB. Earlier, two things had happened: MENUA executed a MARK command on itself (hence the flag); then it called MENUB. Now the menu processor executes a RETURN command. Menu control travels back up through the menu chain and stops at MENUA because it was previously marked. As the menu processor leaves MENUB, it breaks the link between menus, forcing MENUB out of the menu chain.

- It leaves the menu chain altogether without finding any marked menus, as in Figure 2–8. In this case, menu processing ends, and control returns to the software component calling menu services in the first place, whether a user program or interactive services.

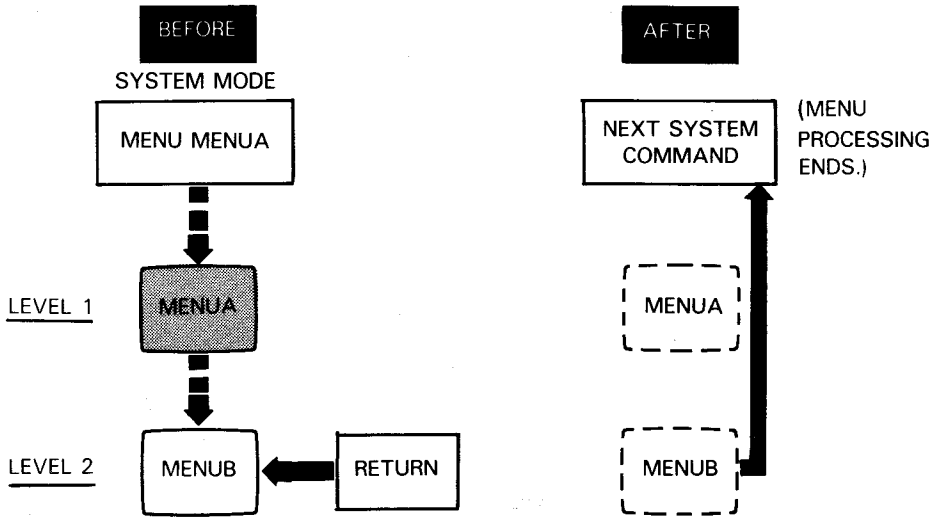


Figure 2–8. Example 2 of the RETURN Command

In Figure 2–8, MENUB is about to execute a RETURN command. Earlier, the workstation operator called menu MENUA, which in turn called MENUB. Neither MENUA nor MENUB has had a MARK command executed. Now, the RETURN command causes the menu processor to search back through the menu chain for a menu marked by the MARK command. MENUA doesn't satisfy this requirement so the menu processor goes all the way back to interactive services. As a result, menu processing ends, the workstation goes blank, and the operator is free to enter another system command.

The MENUCALL command

MENUCALL menuname This command calls the menu specified by menuname and displays it on the operator's workstation, overlaying the previous menu. You specify this command by MENUCA. In calling the new menu, the menu processor establishes it one level forward of the current menu. Figure 2–9 shows one way you can use this command:

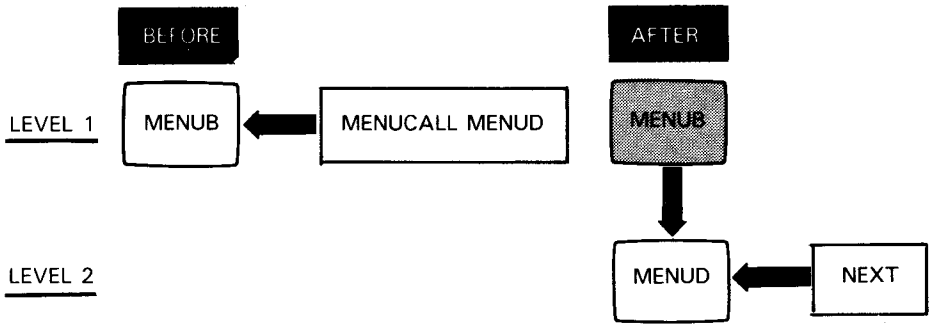


Figure 2–9. An Example of the MENUCALL Command

In Figure 2-9, we first see MENUB just as the menu processor is about to execute action MENUCALL MENUD. The MENUCALL command retrieves menu MENUD, places it at the next menu level, links it to MENUB, and gives control to the action table associated with MENUD. The shading on MENUB also indicates that its screen is completely overlaid by the MENUD screen.

The BACK command

BACK This command moves the menu processor back one level in the menu chain. If the current menu was called by another menu, that menu reappears. If the current menu was called by a user program or the MENU command, menu processing ends and the workstation screen goes blank. Two examples of BACK use follow, the first in Figure 2-10:

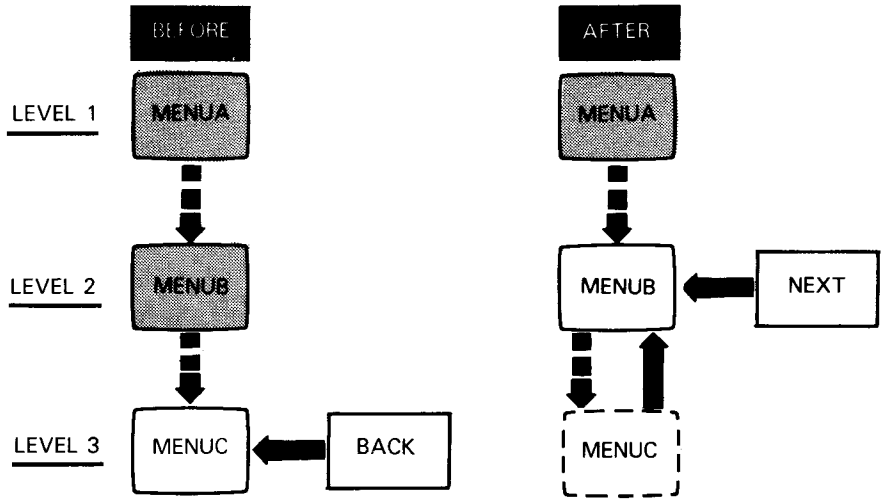


Figure 2-10. Example 1 of the BACK Command

In Figure 2-10, a 3-level menu chain has been built, with the current menu being the level-3 menu MENUC, for which the menu processor is about to execute a BACK command. As a result of the BACK command, the menu processor moves one level back in the chain to MENUB. Also, the link from MENUB to MENUC is broken, forcing MENUC out of the menu chain.

The second BACK example, in Figure 2-11, shows one way you can terminate menu processing.

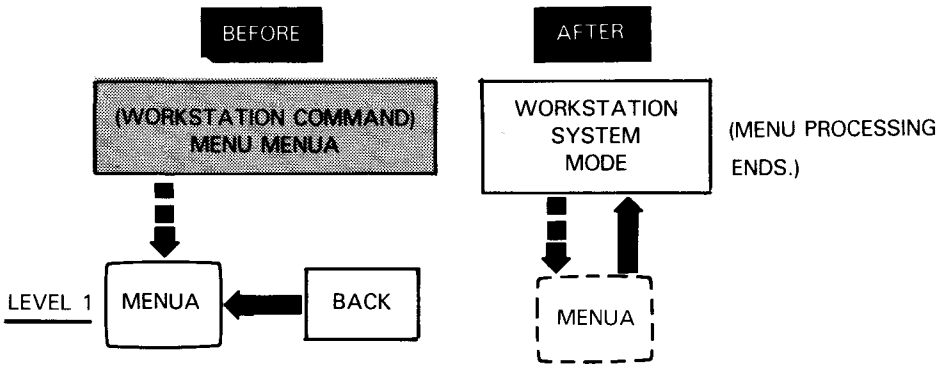


Figure 2–11. Example 2 of the BACK Command

Menu MENUA, having previously been called by the workstation command MENU MENUA, is the current menu. Now the menu processor executes a BACK command within MENUA's action table. The command causes the menu processor to move one level back in the chain, but not to another menu. Because MENUA had been called from outside menu processing in the first place, the BACK command backs out of menu processing completely, returning control of the workstation to interactive services.

The MENU OVERLAY command

MENU OVERLAY menuname Like MENUCALL, this command (written as MENUOV) calls another menu and gives it control. But unlike MENUCALL, the menu processor does not place the new menu at a new level. Instead, it replaces (overlays) the current menu *at its level*. Figure 2–12 shows an example of how it is used.

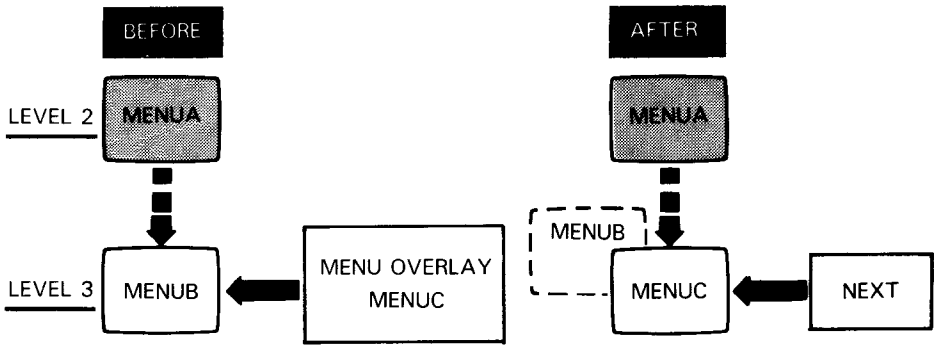


Figure 2–12. An Example of the MENU OVERLAY Command

In Figure 2–12, the current menu is MENUB, which had earlier been called from MENUA. MENUB occupies level 3 as the menu processor prepares to execute a MENU OVERLAY command. The MENU OVERLAY MENUC command calls MENUC and gives it control at the same level MENUB occupied, in effect replacing MENUB in the menu chain.

Using screen formats with menus

The next two menu function commands both depend on the capability OS/3 gives you of creating screen formats. Screen formats resemble a form with *blanks* in which the operator places input data. In fact, their use of a common task - filling in the blanks - makes screen formats, along with menus, a powerful tool for increasing productivity in workstation operators. If you already know how to create and use screen formats, the SCREEN and DISPLAY menu function commands can give you added flexibility in your menus by allowing you to use screen formats directly.

The SCREEN command

SCREEN format-name Use this command to input data to a user program. It obtains the screen format given by format-name, displays it on the workstation screen (overwriting the menu screen), and relays back to the program all data that the operator enters on the screen. Then, provided there are no more actions to execute, the menu screen reappears.

Inputting data with screen formats

Why use screen formats? After all, you can program menus to do the same thing screen formats do: send input data to a program requesting it. The difference lies in where the data comes from. An input-type menu action returns the same string of data every time its corresponding item is selected. But what if you expect the input data to vary? What if you want the workstation operator to enter the data himself instead of letting the menu do it for him? In this case, the SCREEN command can give you and your workstation operators more flexibility.

Figure 2-13 shows an example of SCREEN use.

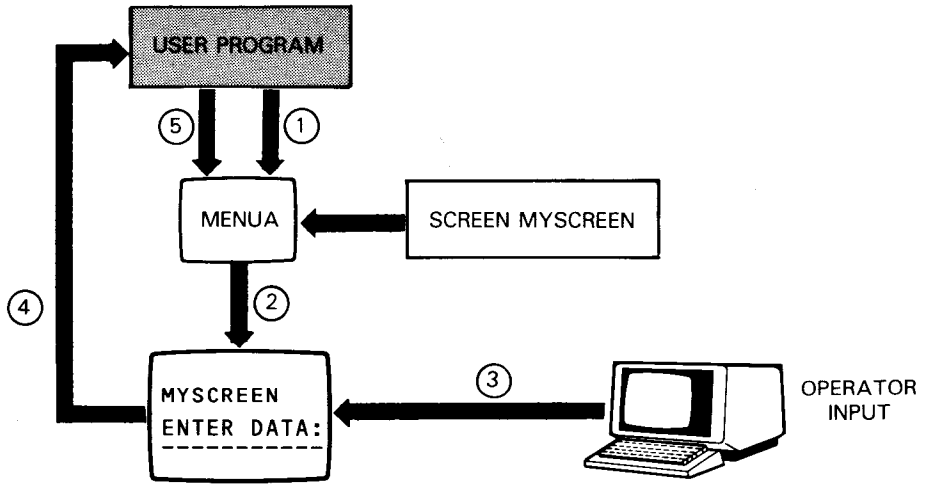


Figure 2-13. An Example of the SCREEN Command

This example is a little more involved than the ones you've seen up to now, so we'll go through it step by step. Each step in the figure corresponds to the following steps:

① A user program needing input data calls menu MENUA. In response to one selection in the menu, the menu processor is about to execute a SCREEN MYSCREEN command.

② The SCREEN command uses the SFC to retrieve screen format MYSCREEN from a format library file. MYSCREEN overlays MENUA on the workstation screen and waits for operator input.

③ The operator keys in the data as requested by MYSCREEN and then presses XMIT.

④ The menu processor sends the keyed-in data of step 3 back to a buffer in the user program, making it available for program use.

⑤ Upon completion of step 4, menu MENUA reappears.

Just as with input data actions, the program calling this screen acts as if the screen were an input device. But now, the data returned to the program depends entirely on what the operator keys in and is not preset the way input actions are. The only restriction on screens is that they may contain only input fields, not output or bidirectional fields.

If you've had any experience with screen formats, you know that you can call a screen format directly from a program without having to go through a menu. But say that an operator is faced with the need to choose from among several different formats. A menu can help in that choice by listing the available formats as items on the menu screen, with each item linked to a corresponding SCREEN command within the action table.

The DISPLAY command

DISPLAY format-name Use the DISPLAY command to display the screen format given by format-name on the operator's workstation. The command retrieves the format from a library file and displays it, overlaying the menu screen. When the operator finishes reading the screen, he presses XMIT. Then, barring any further actions, the menu screen reappears.

Unlike SCREEN, this command works only with formats containing fixed fields displayed purely for the operator's information. Therefore, formats used with DISPLAY cannot have any input, output, or bidirectional fields. Figure 2-14 is an example of how to use the DISPLAY command.

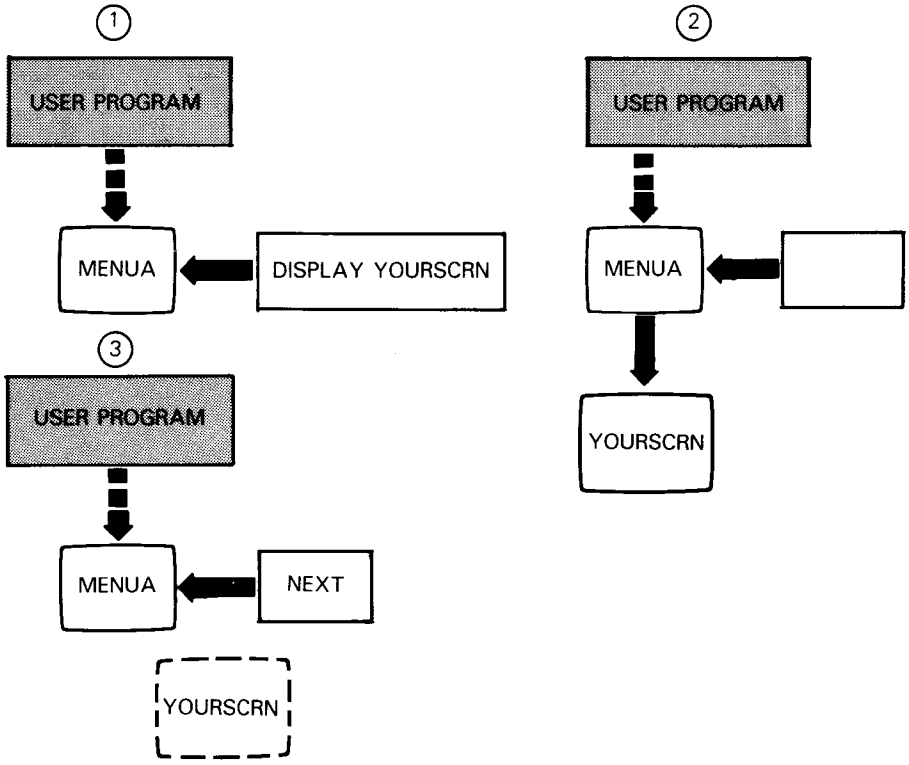


Figure 2–14. An Example of the DISPLAY Command

In Figure 2–14, ① menu MENUA is executing a DISPLAY YOURSCRN command. ② The command calls upon the SFC to retrieve screen YOURSCRN from a format library file and display it for the operator to read. When finished, ③ the operator presses XMIT, which signals the SFC to return screen control to MENUA.

What happens after an action has finished?

When the menu processor has finished executing any of these commands, it takes one of the following actions, depending on the command and its current position in the menu chain:

MARK SCREEN DISPLAY

- If the command is MARK, SCREEN, or DISPLAY, control returns to the action table of the current menu. Then, if more actions remain for the selected item, they are executed. If no more actions remain, the menu screen reappears and prompts the operator for another entry.

RETURN BACK

- If the command is RETURN or BACK, the menu processor leaves the current menu and moves back through the menu chain. If it finds another menu farther back in the chain that satisfies the requirements of the command, the menu's action table gets control of the menu processor, and its screen gets control of the workstation. If no menu is found, menu processing ends and workstation control returns to the software component that originally called it, whether interactive services or a user program.

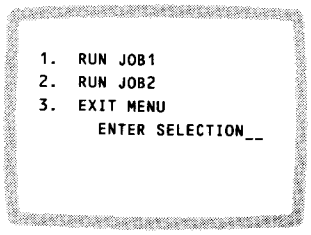
MENUCALL
MENU OVERLAY

- If the command is MENUCALL or MENU OVERLAY, the menu processor leaves the current menu, just as with RETURN or BACK. In this case, though, the menu processor finds a new menu, giving it control of the screen and its action table control of the menu processor. In the additional case of MENU OVERLAY, the new menu diplaces the old one in the menu chain.

2.5. SUMMARY

We've explained in this section the concepts you need to write and use your own menus.

You can think of a menu as a program with input and output. The input is the number of a menu item the operator selects from the menu's list of available items



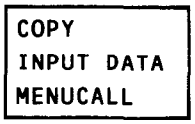
The output varies: it can be a workstation command

a string of input data

or a menu function command

```
MENUCALL
```

Each of these is a menu action, and all actions for all items in a menu are contained in the menu's action table



Menu function commands let you expand the range of services a menu provides by calling other menus, linking menus together in menu chains, and calling additional screen format services if desired.

3. Creating a Menu

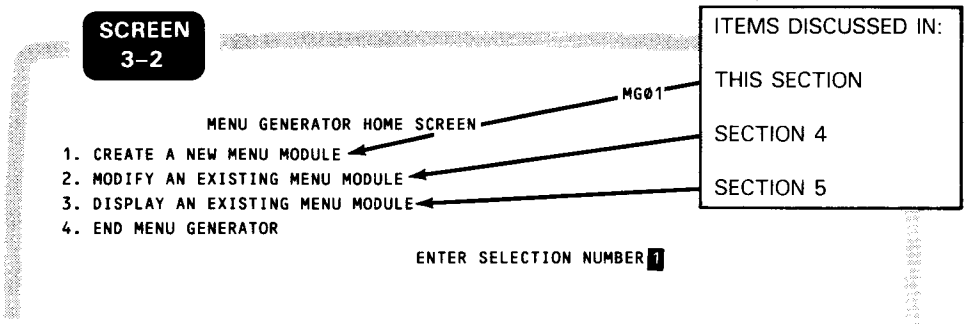
Creating menus with the menu generator (MENUGEN)

Now that you have a basic knowledge of how a menu works, we can tell you how to create it. You create a menu with the menu generator (MENUGEN), a software product that asks you questions about the new menu and uses your replies to build it. Once you've finished using the generator, your menu is ready for use whether you planned to call it with the MENU command or from a user program.

The menu generator uses interactive services to help make it an easy-to-use product. Much of its ease of use lies in its own use of menus to help you choose the characteristics of the menu you are creating. You can choose, for example, among different menu screen sizes, help screen sizes, or action table entries. And for all these menus, Sperry Univac provides help screens to answer any questions that may pop up during menu creation. Before reading on, though, make sure you know how to use a menu. If you need a reminder, go back to Section 1 for a quick but complete summary of menu use.

To begin creating a menu, first call the menu generator. Put your workstation in system mode, enter the workstation command MENUGEN as in screen 3-1, and then press XMIT:

You then see a menu screen labeled MENU GENERATOR HOME SCREEN (screen 3-2):



The home screen

We call screen 3-2 the home screen because MENUGEN always returns to this screen after completing a function. As you can see, there are four choices, three of them functions dealing with menus. We discuss items 2 and 3 in Sections 4 and 5, respectively. Item 4 is your choice when you want to terminate MENUGEN and return your workstation to system mode. But for now, what you want is item 1, so you enter that item number in screen 3-2 and press XMIT.

The create function of MENUGEN

NOTES:

1. *Almost every screen in MENUGEN has a number in the upper right-hand corner, like MG01 in screen 3-2. You won't need to concern yourself about these screen numbers because they identify individual MENUGEN screens to Sperry Univac personnel.*

2. *You can end MENUGEN at any time by pressing function key F15. This is useful when, for example, you decide you don't want to finish a menu that you're working on. Or you may want to quit working on a menu now but return to it later. When you press the F15 key, two things happen:*

- *You see a screen telling you that MENUGEN has terminated.*
- *You get one or more system messages issued by MENUGEN. To see them, put your workstation in SYSTEM MODE. While most of these are for information only, one message asks you if you want to save the menu module you were working on when you terminated MENUGEN. You can save an unfinished menu now and finish it in a later MENUGEN session using its modify operation (Section 4). If you choose not to save the menu, it is lost.*

The four stages of menu creation

You create a menu in four stages, always in this order:

1. Identifying a menu
2. Creating its menu screen
3. Creating its action table
4. Creating its help screens

We'll discuss these, also in order.

3.1. IDENTIFYING A MENU

Before you can specify the details of a menu, you have to tell MENUGEN three things:

1. the name of the menu, which is the name MENUGEN gives the library module containing the menu;

2. the name of the library file in which you want the menu stored (called the *destination file*); and

3. the volume serial number of the disk volume on which the destination file resides.

Specify all this by using screen 3-3, which you see on the screen after you reply to screen 3-2:

SCREEN 3-3

```
                                                    MG0101S
                MENU MODULE OUTPUT INFORMATION
WHAT NAME DO YOU WANT TO GIVE THE MENU MODULE?  ENTER BELOW
MENU MODULE NAME: ________

WHAT FILE LBL SHOULD THE MENU MODULE BE WRITTEN TO?  ENTER BELOW
FILE LBL: ______________________________________________

WHAT VSN SHOULD THE MENU MODULE BE WRITTEN TO?  ENTER BELOW
VSN: ______
```

Specifying the menu name

You must specify a menu module name, which must be one to eight characters long beginning with an alphabetic character. The file and volume serial number parameters may be optional, depending on these guidelines:

Guidelines for the destination file

- If you don't specify a file name, the menu module goes in the system file YFMT.

- If you don't specify a volume serial number, the menu module goes on the system resident volume (SYSRES, usually specified as RES).

- If you plan to call your menu from a user program, you can store it in any existing MIRAM file on any disk volume. When the program is run, you'll use job control to direct menu services to retrieve your menu from that volume and file.

3.2. CREATING OR DEFINING THE MENU SCREEN

Creating the menu screen

Once you've told MENUGEN where to store the new menu module, you're ready to create the menu screen. Actually, you have the choice of either creating a new menu screen entirely from scratch or directing your menu to retrieve an existing menu screen that you created by using another interactive product, the screen format generator. You make your choice by using screen 3-4:

SCREEN 3-4

```
                                                    MG02

                          CREATE MENU SCREEN

1. USE THE DEFAULT MENU SCREEN TEMPLATE.
2. USE THE USER SUPPLIED MENU SCREEN.

                          ENTER SELECTION NUMBER: __
```

Let's look at each item in screen 3-4.

3.3. USING A MENUGEN MENU SCREEN TEMPLATE

Using a menu screen template

To create your menu screen with MENUGEN, choose item 1 in screen 3-4. By making this choice, you direct MENUGEN to display a template, a skeleton screen containing numbered blanks that you fill in with the menu title and the items you want to appear on the menu. You can think of a template as a menu screen layout like the one in screen 3-5:

SCREEN 3-5

```
                                                    MG0301S

                          MENU SCREEN TEMPLATE

INSERT MENU SCREEN TITLE BELOW:
------------------------------------------------------------
1. ____________________________  7. ____________________________
2. ____________________________  8. ____________________________
3. ____________________________  9. ____________________________
4. ____________________________  10.____________________________
5. ____________________________  11.____________________________
6. ____________________________  12.____________________________
```

The template shown in screen 3-5 is one of several predefined templates available with MENUGEN; we'll show you in a moment what they are and how to call them. For now, let's assume you want to create the menu screen of Figure 1-1. You would fill in screen 3-5 as follows:

```
                                                              MG0301S
                          MENU SCREEN TEMPLATE
INSERT MENU SCREEN TITLE BELOW:
                           *** PAYROLL MENU ***________________
1. PAYROLL DATA ENTRY__________  7.  RESTORE PAYROLL FILES______
2. PAYROLL INQUIRY_____________  8.  EXIT FROM THIS MENU________
3. PAYROLL EDIT________________  9.  ___________________________
4. PAYROLL CHECKS______________  10. ___________________________
5. PAYROLL ANALYSIS REPORTS____  11. ___________________________
6. BACKUP PAYROLL FILES________  12. ___________________________
```

You then press XMIT to send the filled-in screen to MENUGEN. When the menu is later called, it appears like this:

```
                          *** PAYROLL MENU ***
1. PAYROLL DATA ENTRY              7. RESTORE PAYROLL FILES
2. PAYROLL INQUIRY                 8. EXIT FROM THIS MENU
3. PAYROLL EDIT                    9.
4. PAYROLL CHECKS                  10.
5. PAYROLL ANALYSIS REPORTS        11.
6. BACKUP PAYROLL FILES            12.
                                   ENTER SELECTION NUMBER: ___
```

Choosing a template

Screen 3-5 is one of several predefined templates available with MENUGEN. All templates look basically alike, providing blanks for a menu title and a number of menu items. They differ only in the number of items displayed and in the overall size of the screen.

When you request a template by transmitting screen 3-4, screen 3-6 appears to ask you what size template you want:

SCREEN 3-6

```
                                                    MG03

          WHAT SIZE SCREEN IS DESIRED?

1. UNIVERSAL (12X64)                5. 12X80
2. CURRENT                          6. 16X80
3. 16X64                            7. 24X80
4. 24X64

                    ENTER SELECTION NUMBER: __
```

Use screen 3-6 to specify the template size you want. This also determines the size of the resulting menu screen. Except for item 2, each item describes a menu screen template in terms of number of items multiplied by the length of the menu's title line. (The length of menu items doesn't vary: it's always 29 characters.)

MENUGEN won't let you create a menu screen that's larger than your workstation screen. If you try, MENUGEN simply redisplays screen 3-6 so you can specify another template. If you don't know your screen size, choose item 2, CURRENT. It automatically displays the largest template that can fit on your screen.

Available template sizes

Screens 3-7a to 3-7f show what template sizes are available:

12X64 SIZE TEMPLATE

SCREEN 3-7a

```
                                                    MG0301S

                        MENU SCREEN TEMPLATE
INSERT MENU SCREEN TITLE BELOW:
------------------------------------------------------------
1. ____________________________ 7. ____________________________
2. ____________________________ 8. ____________________________
3. ____________________________ 9. ____________________________
4. ____________________________ 10. ____________________________
5. ____________________________ 11. ____________________________
6. ____________________________ 12. ____________________________
```

SCREEN 3-7b

16X64 SIZE TEMPLATE

```
                                                    MG0303S

                        MENU SCREEN TEMPLATE

INSERT MENU SCREEN TITLE BELOW:
----------------------------------------------------------------

1. ____________________________ 9. ____________________________
2. ____________________________ 10. ____________________________
3. ____________________________ 11. ____________________________
4. ____________________________ 12. ____________________________
5. ____________________________ 13. ____________________________
6. ____________________________ 14. ____________________________
7. ____________________________ 15. ____________________________
8. ____________________________ 16. ____________________________
```

SCREEN 3-7c

24X64 SIZE TEMPLATE

```
                                                    MG0304S

                        MENU SCREEN TEMPLATE

INSERT MENU SCREEN TITLE BELOW:
----------------------------------------------------------------

1. ____________________________ 13. ____________________________
2. ____________________________ 14. ____________________________
3. ____________________________ 15. ____________________________
4. ____________________________ 16. ____________________________
5. ____________________________ 17. ____________________________
6. ____________________________ 18. ____________________________
7. ____________________________ 19. ____________________________
8. ____________________________ 20. ____________________________
9. ____________________________ 21. ____________________________
10. ____________________________ 22. ____________________________
11. ____________________________ 23. ____________________________
12. ____________________________ 24. ____________________________
```

12X80 SIZE TEMPLATE

SCREEN 3-7d

```
                                                          MG0305S
                        MENU SCREEN TEMPLATE
INSERT MENU SCREEN TITLE BELOW:
--------------------------------------------------------------------
1. _____________________________ 7.  _____________________________
2. _____________________________ 8.  _____________________________
3. _____________________________ 9.  _____________________________
4. _____________________________ 10. _____________________________
5. _____________________________ 11. _____________________________
6. _____________________________ 12. _____________________________
```

16X80 SIZE TEMPLATE

SCREEN 3-7e

```
                        MENU SCREEN TEMPLATE
INSERT MENU SCREEN TITLE BELOW:
--------------------------------------------------------------------
1. _____________________________ 9.  _____________________________
2. _____________________________ 10. _____________________________
3. _____________________________ 11. _____________________________
4. _____________________________ 12. _____________________________
5. _____________________________ 13. _____________________________
6. _____________________________ 14. _____________________________
7. _____________________________ 15. _____________________________
8. _____________________________ 16. _____________________________
```

24X80 SIZE TEMPLATE

SCREEN 3-7f

```
                                                          MG0307S
                        MENU SCREEN TEMPLATE
INSERT MENU SCREEN TITLE BELOW:
--------------------------------------------------------------------
1.  _____________________________ 13. _____________________________
2.  _____________________________ 14. _____________________________
3.  _____________________________ 15. _____________________________
4.  _____________________________ 16. _____________________________
5.  _____________________________ 17. _____________________________
6.  _____________________________ 18. _____________________________
7.  _____________________________ 19. _____________________________
8.  _____________________________ 20. _____________________________
9.  _____________________________ 21. _____________________________
10. _____________________________ 22. _____________________________
11. _____________________________ 23. _____________________________
12. _____________________________ 24. _____________________________
```

Upon transmitting a filled-in template to MENUGEN, you've finished creating the menu screen. MENUGEN now prompts you to create the action table to be linked to the menu as described in 3.5.

3.4. USING A USER-DEFINED MENU SCREEN

Using an existing menu screen

To use a previously defined screen format as your menu screen, you choose item 2 in screen 3-4. Then, when your menu is called, the menu processor determines that this screen resides outside the menu module. The menu screen can be in YFMT or in another menu library file if that's where your menu is located. Wherever the screen is, the menu processor retrieves and displays it as your menu screen. Figure 3-1 outlines the process for the case where both menu module and menu screen module reside on a user library file.

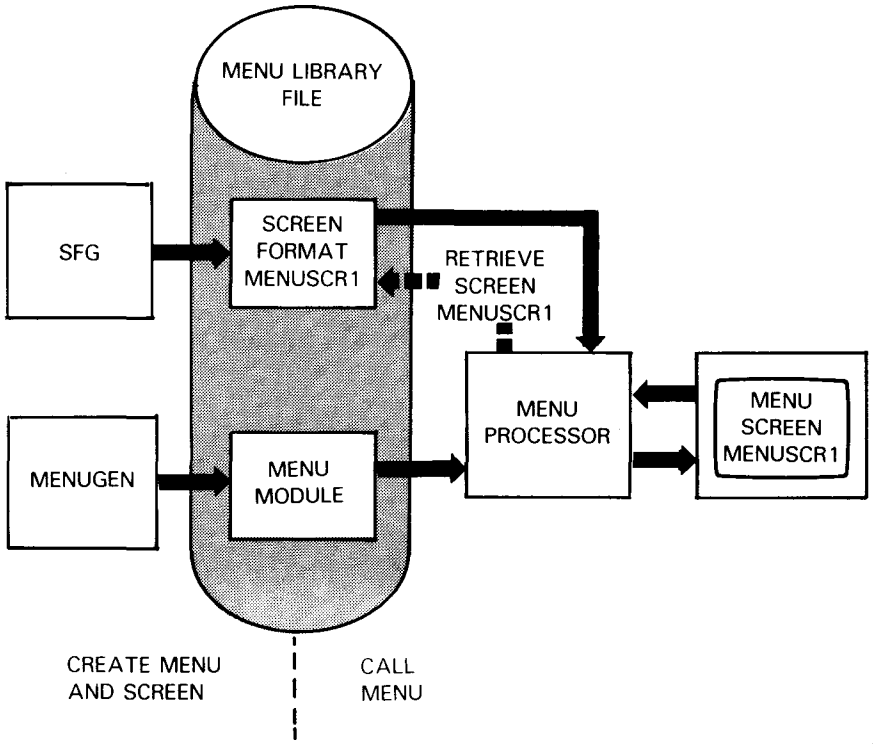


Figure 3-1. Creating and Using an SFG-created Menu Screen

The advantages of existing menu screens

Flexible format

We make this feature available because, as you gain experience with menus, menu layouts other than those available with MENUGEN templates may suggest themselves. You might want to use fewer than 12 items, specify items longer than 29 characters, or display additional information on the menu screen, to name just three examples.

Flexible programming

There are other advantages to the use of existing help screens. You can reuse an old screen in a new menu. And because you can create menu screens separately from menus, you can create your menu concentrating on its basic functions and leaving menu screen details to the screen format generator.

Specifying where to find SFG-created screens

To help you integrate an existing menu screen with your menu, MENUGEN lets you display the menu screen at certain points in the menu creation process, asking you each time what file and volume the screen resides on. (For the screen to be acceptable as a menu screen, you must create it following certain guidelines listed in Appendix B.)

NOTE:

If you use an existing screen format as your menu screen, it must be stored either in YFMT or in the same library file as the menu module if a library file is specified.

When you call MENUGEN and choose item 2 of screen 3-4, you see screen 3-8:

SCREEN 3-8

```
                                                                MG0202S
                              WHAT IS YOUR MENU SCREEN NAME?
INSERT BELOW THE NAME YOU GAVE THE SCREEN WHEN YOU CREATED
IT USING THE SCREEN FORMAT GENERATOR:
MENU SCREEN NAME: ________
```

Fill in screen 3-8 with the name of the screen you want as the menu screen for your new menu. When the menu is called, the screen format module must reside either in the same file as the menu module or in YFMT. If it doesn't, you can copy it to the menu library file or YFMT with the MIRAM librarian or the COPY system command. The MIRAM librarian is described in the current version of the system service programs user guide. The COPY command is described in the current version of the interactive services concepts and facilities manual.

When you transmit screen 3-8 with the screen name filled in, you've finished defining the menu screen. MENUGEN now prompts you to create the action table to be linked to the menu.

3.5. CREATING THE ACTION TABLE

As we've said, a menu's action table contains the data and commands that the menu processor passes to the rest of the system in response to each item on the menu. Since we discuss elsewhere how to use actions, we confine ourselves here to the mechanics of specifying them by using MENUGEN.

After you've completed the menu screen as specified in 3.2, screen 3-9 appears:

SCREEN 3-9

```
                                                          MG04
                        CREATE ACTION TABLE
1. CREATE ACTION TABLE (EXPERIENCED USER).
2. CREATE ACTION TABLE (INEXPERIENCED USER).
3. DISPLAY MENU SCREEN ASSOCIATED WITH THIS ACTION TABLE.
4. OUTPUT MENU MODULE TO FILE.
5. END CREATE OPERATION.
                                    ENTER SELECTION NUMBER: __
```

The two modes of creating action tables

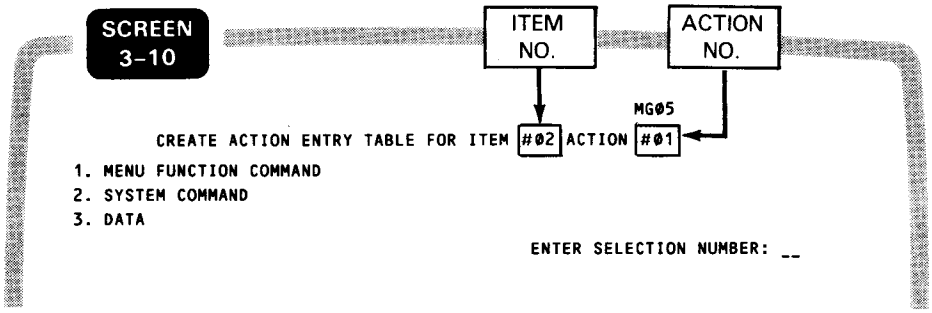
As you can see, MENUGEN has two ways of creating an action table. One (item 2) is for the new MENUGEN user. If you are one, the INEXPERIENCED USER option guides you step by step as you create actions, making extensive use of menus and help screens.

As useful as the INEXPERIENCED USER option is to a new user, MENUGEN has another option: action table creation for the EXPERIENCED USER (item 1). You get fewer menus and help screens with this option because MENUGEN assumes you know exactly how to create your desired action table. In return for fewer screens though, this option works faster than the other and generally becomes a more natural way of programming action tables for the experienced user.

We discuss these two options first in 3.6 and 3.11, respectively. We defer mention of the other screen 3-9 options to 3.12.

3.6. CREATING AN ACTION TABLE FOR THE INEXPERIENCED USER

Choose item 2 of screen 3-9 if you are an inexperienced MENUGEN user. This MENUGEN option guides you step by step as you specify the actions that make up your action table. In place of screen 3-9, you see screen 3-10:

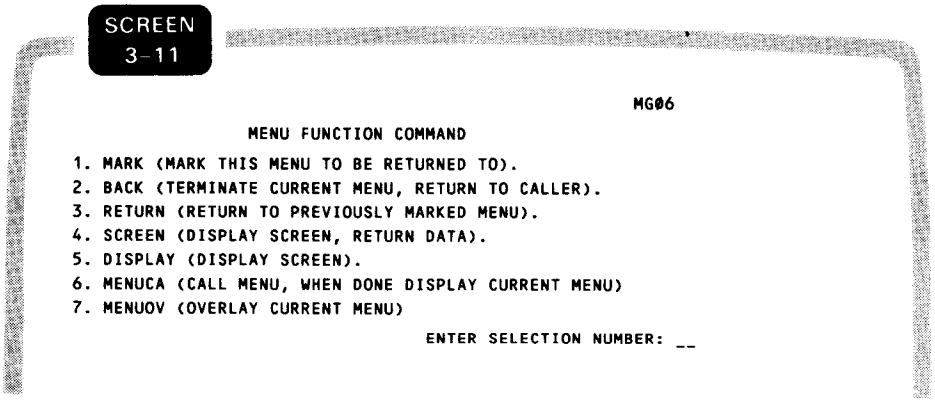


Specifying the current item number

Note that the title line of screen 3-10 indicates the number of the item for which you are currently creating an action, and also the number of the action when you're creating more than one action for that item. This information helps you keep track of your progress. You may recognize in screen 3-10 the three types of actions we discussed in Section 2: menu function commands, system commands, and input data. We'll discuss each of these in turn.

3.7. SPECIFYING A MENU FUNCTION COMMAND

If you wish to specify a menu function command, select item 1 in screen 3-10. MENUGEN responds by displaying screen 3-11:



Screen 3-11 presents all seven menu function commands. You simply decide which one you want to specify, enter its number in the prompt, and transmit the screen to MENUGEN. Table 3-1 summarizes these commands.

Table 3-1. Menu Function Commands

Command	Function
MARK	Sets a mark in the current menu
BACK	Causes menu to give up workstation control to menu that called it or, if not a menu, to user program or interactive services
RETURN	Causes menu processor to backtrack through menu chain either until it leaves the menu chain completely or until it finds a menu for which the MARK command has been executed
SCREEN format-name	Calls a screen format given by format-name and gives it control of the workstation until operator transmits it, when calling menu gets control back. Screen may be used to input data.
DISPLAY format-name	Calls a screen format given by format-name for display only
MENUCALL menu-name (written MENUCA)	Calls menu specified by menu-name and gives it control of the workstation one level beyond calling menu's level
MENU OVERLAY menu-name (written MENUOV)	Calls menu specified by menu-name and gives it control of the workstation at calling menu's level

Specifying additional information for some menu commands

Of these commands, four require additional information that you provide as follows:

```
4. SCREEN (DISPLAY SCREEN, RETURN DATA).
```

To completely specify the SCREEN command, you have to include a screen format name. If you choose item 4 of screen 3-11, MENUGEN asks you for a screen name with screen 3-12:

SCREEN 3-12

```
                                                    MG0604S

                    SCREEN MENU FUNCTION COMMAND

ENTER SCREEN NAME: ________
```

Enter the screen name and transmit the screen to MENUGEN to finish specifying the SCREEN menu function command. MENUGEN then prompts you for more actions, as described in 3.10.

5. DISPLAY (DISPLAY SCREEN).

To completely specify the DISPLAY command, you have to include a screen format name. If you choose item 5 of screen 3-11, MENUGEN asks you for a screen name with screen 3-13:

SCREEN 3-13

```
                                                    MG0605S

              DISPLAY MENU FUNCTION COMMAND

ENTER SCREEN NAME: ________
```

Enter the screen name and transmit the screen to MENUGEN to finish specifying the DISPLAY menu function command. MENUGEN then prompts you for more actions, as described in 3.10.

6. MENUCA (CALL MENU, WHEN DONE DISPLAY CURRENT MENU)

To completely specify the MENU CALL command, you have to include a menu name. If you choose item 6 of screen 3-11, MENUGEN asks you for a menu name with screen 3-14:

SCREEN 3-14

```
                                                    MG0606S

              MENU CALL MENU FUNCTION COMMAND

ENTER SCREEN NAME: ________
```

Enter the menu name and transmit the screen to MENUGEN to finish specifying the MENU CALL menu function command. MENUGEN then prompts you for more actions, as described in 3.10.

7. MENUOV (OVERLAY CURRENT MENU)

To completely specify the MENU OVERLAY command, you have to include a menu name. If you choose item 7 of screen 3-11, MENUGEN asks you for a screen name with screen 3-15:

SCREEN 3-15

```
                                                    MG0607S

          MENU OVERLAY MENU FUNCTION COMMAND

ENTER SCREEN NAME: ________
```

Enter the menu name and transmit the screen to MENUGEN to finish specifying the DISPLAY menu function command. MENUGEN then prompts you for more actions, as described in 3.10.

3.8. SPECIFYING A SYSTEM COMMAND

Listing and specifying system commands

If you want to specify a system command as an action, select item 2 in screen 3-10. MENUGEN responds by displaying screen 3-16:

SCREEN 3-16

```
                                                    MG07

                SYSTEM COMMANDS (PART1)

1  ALLOCATE     8  COMMENT     15  ENTER      22  HOLD
2  ASK          9  CONNECT     16  ERASE      23  IN
3  BASIC        10 COPY        17  FILE       24  LOGON
4  BEGIN        11 DDP         18  FREE       25  LOGOFF
5  BRKPT        12 DELETE      19  FSTATUS    26  MENU
6  CANCEL       13 DISPLAY     20  GO         27  ENTER ACTION
7  CHANGE       14 EDT         21  HELP       28  NEXT MENU

                   ENTER SELECTION NUMBER: ___
```

Screen 3-16 is only a partial list of system commands you can specify as menu actions. To see the rest of the list, choose item 28, NEXT MENU. You then see screen 3-17:

SCREEN 3-17

```
                                                        MG08
                    SYSTEM COMMANDS (PART2)
29  OCL         36  RECOVER     43 SI          50  ENTER ACTION
30  OV          37  REMARK      44 SDU         51  PREVIOUS MENU
31  PAUSE       38  RESUME      45 SMU
32  PRINT       39  RUN         46 STATUS
33  PTSTAT      40  RV          47 STOP
34  PUNCH       41  SCREEN      48 TELL
35  REBUILD     42  SCHED       49 VTOC
                    ENTER SELECTION NUMBER: ___
```

You can go back to screen 3-16 at any time by choosing item 51, PREVIOUS MENU. Using items 28 and 51, you can move back and forth between the two screens as you build your action table. The other entries you can make on screens 3-16 and 3-17 include:

- Items 1-26 (screen 3-16) and items 29-49 (screen 3-17)

Displaying help screens for commands

These items are OS/3 system commands. Choosing one displays a help screen explaining the command. When you finish reading the help screen, press XMIT to return to screen 3-16 or 3-17, whatever screen it was from which you called the help screen.

- Item 27 (screen 3-16) and item 50 (screen 3-17)

Entering commands

When you decide what command you want, enter it as a menu action by choosing either item 27 or item 50. You'll then see screen 3-18:

SCREEN 3-18

```
                                                        MG0727S
ENTER SYSTEM COMMAND BELOW:
--------------------------------------------------------------------------------
```

Using screen 3-18, key in the system command exactly as you want it to be sent to the system when that action is executed. (The command can't exceed 72 characters in length.) When you press XMIT, MENUGEN creates an action from the command and then prompts you for more actions, as described in 3.10.

3.9. SPECIFYING INPUT DATA

If you want to specify a string of input data to be sent to a user program, you create that type of action by selecting item 3 of screen 3–10. You then see screen 3–19:

SCREEN 3-19

```
                                                    MG0503S
                        DATA ACTION ENTRY
ENTER DATA BELOW:
--------------------------------------------------------------------------
```

Screen 3–19 consists of a 72-character field into which you enter the data exactly as it is to be sent to the user program. When you send this screen to MENUGEN, it creates the action and then prompts you for more actions, as described in 3.10.

3.10. MORE ACTIONS?

Each time you create an action, whether input data, system command, or menu function command, you see screen 3–20:

SCREEN 3-20

```
                                                    MG09
                        NEXT ACTION
1. ANOTHER ACTION FOR SAME ITEM #    5. DISPLAY MENU SCREEN
2. ACTION FOR NEXT ITEM #            6. WRITE MENU MODULE TO FILE
3. ACTION TABLE IS COMPLETE          7. END CREATE OPERATION
4. DISPLAY ACTION TABLE
                                  ENTER SELECTION NUMBER: __
```

Your choices in screen 3–20 include:

1. ANOTHER ACTION FOR SAME ITEM #

More actions for the current item

If you have more actions to add to the menu item you're currently specifying, choose this item. It redisplays screen 3–10 for you to enter another action. The ITEM field in screen 3–10 remains the same, but the ACTION field is increased by 1. If for example you previously specified action 3 for an item, screen 3–10 tells you that the next action you create will be the fourth action for that same item.

If creating multiple actions for the same item, especially system command actions, remember our earlier note that these actions are performed consecutively. If two actions are RV commands, for example, they will be sent to the run processor just as if you yourself keyed in one RV command right after the other.

2. ACTION FOR NEXT ITEM #

Actions for the next item

If you want to move on to specifying actions for your next item, choose this item. Like item 1, it redisplays screen 3-10 for you to enter another action. But in this case, the ITEM field is increased by 1 to the next item's number, and the ACTION field is reset to 1, indicating the first action for this item.

3. ACTION TABLE IS COMPLETE

Ending the action table

If you have no more actions to specify, choose this item. It stores the actions you've already entered and causes MENUGEN to move to the next stage of menu creation, help screens, as described in 3.13.

4. DISPLAY ACTION TABLE

Displaying the action table

If you want to display your menu's action table, say, to check your progress or to look for errors, choose this item. The screen that appears resembles screen 3-21 except for individual actions:

SCREEN 3-21

```
                                                    MG0904T
                        DISPLAY ACTION TABLE
ITEM #  ACTION TYPE  ACTION
 1          S        RV PAYROL1:(PAYLOAD,PACK01)
 2          S        RV PAYROL2:(PAYLOAD,PACK01),,IF=MASTERFILE,I
                     V=PACK03,DEBUG=NO
 3          S        RV PAYROL3:(PAYLOAD,PACK03)
 4          M        MARK
            M        MENUCA MENU02
            M        RETURN
TYPES: (S)YS COMMAND, (D)ATA, (M)ENU COMMAND
```

MULTILINE ACTION

MULTIACTION ITEM

Screen 3-21 is simply a list of all actions contained in your menu. Each line represents an action, showing the number of the item to which the action is linked, the type of action it is (explained on the last line of the screen), and its text. Note that some actions take up more than one line, such as the action for item 2.

After you're finished reading the screen, press XMIT. If your actions take up more than one screen, MENUGEN displays each screen and waits until you press XMIT before going on to the next screen. After the last screen has been displayed, you see screen 3-20 again.

NOTE:

The last action you entered won't be displayed until after you've finished creating the action table. Also, if you've created only one action so far, MENUGEN will skip over displaying the action table, returning you directly to screen 3-20.

5. DISPLAY MENU SCREEN

Displaying the menu screen

If you want to see the menu screen for your menu, choose this item. The menu screen then appears. If you're supplying the menu screen from outside MENUGEN, by using the SFG, MENUGEN first asks you where to find the menu by using screen 3-22:

SCREEN 3-22

```
                                                        MG9901S

                    WHERE IS YOUR MENU SCREEN

TO DISPLAY YOUR MENU SCREEN. THE MENU GENERATOR MUST BE
TOLD WHERE IT IS. ENTER THE FILE NAME (LBL). AND THE VSN
ON WHICH THE SCREEN RESIDES. THE DEFAULTS ARE PRINTED NEXT
TO THE INPUT FIELDS. EACH WILL BE USED IF ITS ASSOCIATED
FIELD IS LEFT BLANK.

FILE LBL: _____________________________________________ ($Y$FMT)
FILE;VSN: ______ (RES)
```

Fill in the file identifier and volume serial number of the file where the menu screen currently resides, whether it's your menu library or your secondary file, and then press XMIT. MENUGEN uses this information to retrieve and display the menu screen. When you've finished reading the menu screen, press XMIT to return to screen 3-20.

6. WRITE MENU MODULE TO FILE

Making a backup copy of the menu

You may want to back up the work you've put into creating the menu by putting a copy of what you've done so far in the destination file. That's what happens when you choose this item. If necessary, you can later use the MENUGEN modify operation to change or add to the menu, as described in Section 4. After the backup operation finishes, screen 3-20 reappears.

7. END CREATE OPERATION

Writing a menu to the destination file and going to the home screen

If you want to end the MENUGEN create operation at this time, choose this item. It writes your menu to the menu module on the destination file and then displays screen 3-2, the MENUGEN home screen.

3.11. CREATING AN ACTION TABLE FOR THE EXPERIENCED USER

Choose item 1 of screen 3-9 if you are an experienced MENUGEN user. In response to this selection, MENUGEN displays screen 3-23:

SCREEN 3-23

```
                                                          MG0401S
            CREATE ACTION TABLE FOR EXPERIENCED USER
ITEM #  ACTION TYPE  ACTION (IF GREATER THAN 44 CHARACTERS
(1-99)  (S, D, OR M)        CONTINUE IN NEXT ACTION FIELD.)
 --         -       --------------------------------------------
 --         -       --------------------------------------------
 --         -       --------------------------------------------
 --         -       --------------------------------------------
 --         -       --------------------------------------------
 --         -       --------------------------------------------
 --         -       --------------------------------------------
 --         -       --------------------------------------------
```

Action type:
- ***S*** *System command*
- ***D*** *Data*
- ***M*** *Menu function command*

Filling in an action table

Each line of screen 3-23 is an action with three fields that you fill in:

1. the item number on your menu screen to which this action is linked;
2. the type of action it is (S for a system command, D for input data, or M for a menu function command); and
3. the action itself.

To specify more than one action for an item, use the same action number for consecutive actions. The action field is 44 characters long, but you can specify longer actions by keying in the remainder on the action field of the next line, being sure to leave its item number and action type fields blank. Screen 3-24 shows an example of a filled-out screen 3-23:

SCREEN 3-24

```
                                                          MG0401S
          CREATE ACTION TABLE FOR EXPERIENCED USER
ITEM #  ACTION TYPE  ACTION (IF GREATER THAN 44 CHARACTERS
(1-99) (S, D, OR M)        CONTINUE IN NEXT ACTION FIELD.)
 1      S            RV PAYROL1:(PAYLOAD,PACK01)----------------
 2      S            RV PAYROL2:(PAYLOAD,PACK01),,IF=MASTERFILE,I
 --     -            V=PACK03,DEBUG=NO--------------------------
 3      S            RV PAYROL3:(PAYLOAD,PACK03)----------------
 4      M            MARK---------------------------------------
 4      M            MENUCA MENU02------------------------------
 4      M            RETURN-------------------------------------
 --     -            -------------------------------------------
```

As you can see in screen 3-24, items 1 and 3 represent single-action items. Item 2 shows how you can extend an action from one action field to the next. For item 4, three actions have all been specified, each keyed to the item number field.

When you've completed filling in screen 3-23, or when you fill up all of its available fields, transmit the screen. Screen 3-25 then appears:

SCREEN 3-25

```
                                                          MG13
              EXPERIENCED USER ACTION TABLE (CONT'D)
1. CREATE MORE ACTIONS          4. DISPLAY MENU SCREEN
2. ACTION TABLE IS COMPLETE     5. WRITE MENU MODULE
3. DISPLAY ACTION TABLE         6. END CREATE OPERATION
                                 ENTER SELECTION NUMBER: __
```

Your choices in screen 3-25 include:

1. CREATE MORE ACTIONS

More actions to specify

If you have more actions to specify than screen 3-23 can hold, choose this item. It stores the actions you've already entered and asks you to specify more by redisplaying screen 3-23.

2. ACTION TABLE IS COMPLETE

No more actions to specify

If you have no more actions to specify, choose this item. It stores the actions you've already entered and prompts you for your menu's help screens, as described in 3.13.

3. DISPLAY ACTION TABLE

Displaying the action table

If you want to see a complete list of the actions you've created so far, choose this item. MENUGEN displays the list in a format similar to that of screen 3-21. If your actions take up more than one screen, MENUGEN displays each screen and waits until you press XMIT before going on to the next screen. After the last screen has been displayed, you see screen 3-25 again.

NOTE:

The last action you entered won't be displayed until after you've finished creating the action table. Also, if you've created only one action so far, MENUGEN will skip over displaying the action table, returning you directly to screen 3-25.

4. DISPLAY MENU SCREEN

Displaying the menu screen

If you want to see the menu screen for your menu, this item causes it to appear. If you're supplying the menu screen from outside MENUGEN, by using the SFG, MENUGEN first asks you where it resides, with screen 3-22, and then uses that information to retrieve and display the menu screen. When you've finished reading the menu screen, press XMIT to return to screen 3-25.

5. WRITE MENU MODULE

Making a backup copy of the menu

You may want to back up the work you've put into creating the menu by putting a copy of what you've done so far in the destination file. That's what happens when you choose this item. If necessary, you can use the MENUGEN modify operation later to change or add to the menu, as described in Section 4. After the backup operation finishes, you see screen 3-25 again.

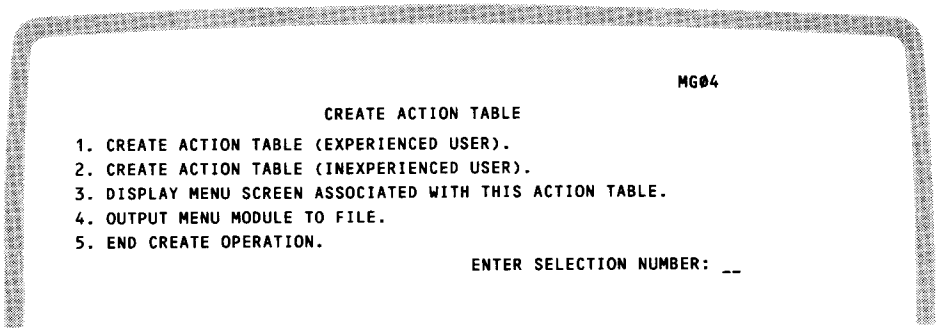
6. END CREATE OPERATION

Writing a menu to the destination file and going to the home screen

If you want to end the MENUGEN create operation at this time, choose this item. It copies the menu to the menu module on the destination file and then displays screen 3-2, the MENUGEN home screen.

3.12. OTHER ACTION TABLE FUNCTIONS

Besides creating action tables, screen 3-9 has other options. Let's look at screen 3-9 once again:



We've already discussed items 1 and 2. The other three choices are:

3. DISPLAY MENU SCREEN ASSOCIATED WITH THIS ACTION TABLE.

Displaying the menu screen

If you want to see the menu screen associated with your menu, choose this item. The menu screen then appears. If you're supplying the menu screen from outside MENUGEN, using the SFG, MENUGEN first asks you where it resides, with screen 3-22, and then uses that information to retrieve and display the menu screen. When you've finished reading it, press XMIT to return to screen 3-9.

4. OUTPUT MENU MODULE TO FILE.

Making a backup copy of the menu

If you want to make a backup copy of the work you've done so far, this item writes a copy of the menu to the destination file and then causes screen 3-9 to reappear.

5. END CREATE OPERATION

Writing a menu to the destination file and going to the home screen

If you want to end the MENUGEN create operation at this time, choose this item. It copies the menu to the menu module on the destination file and then displays screen 3-2, the MENUGEN home screen.

3.13. CREATING HELP SCREENS

Now that you've created your menu's screen and action table, you can proceed to this last stage of menu creation, specifying its help screens. Unlike the menu screen and action table, help screens aren't a necessary part of a functioning menu. In fact, once you've created its action table, your menu is ready to use. But help screens are a big part of the philosophy behind menus, which is to make them easy to use. Therefore, we've provided you the capability to create help screens to go with all your menus.

Two ways of creating help screens

One feature of help screens that we'll discuss in more detail is how and when you create them. There are two ways of specifying help screens:

Creating help screens with the MENUGEN command

1. You can create a help screen by using MENUGEN itself. In fact, you can create up to four help screens for a single item. In use, the operator reads one help screen and then presses XMIT to call the next.

Creating help screens with the SFG

2. You can direct MENUGEN to retrieve a screen format that you've previously created by using the screen format generator (SFG). The screen must already reside in the destination file for MENUGEN to successfully retrieve it. With this method, you can specify as many help screens as you want for a single item. As with MENUGEN-created help screens, the operator reads each of these screens and then presses XMIT to call the next screen.

The menu processor handles these help screens as it does a menu screen that lies outside your menu (3.4). The advantages over MENUGEN-created screens are the same, too: you can reuse an old help screen in a new menu, and you can design the menu in such a way as to leave the details of a help screen to the SFG.

NOTE:

If you use existing screen formats as help screens, they must be stored either in YFMT or in the same library file as the menu module if a file is specified.

You can use both SFG-created and MENUGEN-created help screens within the same menu, and even within the same item.

Help screen creation automatically begins after you've indicated to MENUGEN that you've finished creating your action table. At this point you see screen 3-26:

SCREEN 3-26

ITEM NO.

```
                                                         MG10
                CREATE HELP SCREEN FOR ITEM #00
1. USE DEFAULT HELP SCREEN TEMPLATE   5. DISPLAY ACTION TABLE
2. USE AN EXISTING HELP SCREEN        6. WRITE MENU MODULE
3. NO HELP FOR ITEM                   7. HELPS ARE COMPLETE, END CREATE
4. DISPLAY MENU SCREEN
                                      ENTER SELECTION NUMBER: __
```

Displaying the current item

Note that the title line of screen 3–26 indicates the item number for which you are currently specifying a help screen. This helps you keep track of your progress. In screen 3–26, item 0 (#00) indicates that you are currently creating the summary help screens for your menu. Subsequent help screens are linked with item 1 (#01), item 2 (#02), and so on. Let's look at each item in screen 3–26:

1. USE DEFAULT HELP SCREEN TEMPLATE

If you want to create a help screen with MENUGEN, choose this item. You create a help screen much as you do a menu screen, by filling in a template that MENUGEN supplies you. After choosing item 1, you see screen 3-27:

SCREEN 3-27

```
                                                 MG11
          WHAT SIZE HELP SCREEN TEMPLATE#
1. UNIVERSAL (12X64)              5. 12X80
2. CURRENT                        6. 16X80
3. 16X64                          7. 24X80
4. 24X64
                 ENTER SELECTION NUMBER: __
```

Choosing and filling in a template

Use screen 3–27 to specify the template size you want, which determines the size of the resulting help screen. Except for item 2, each item describes a help screen template in terms of the number of lines multiplied by the characters per line.

As with menu screens, MENUGEN won't let you create a help screen that's larger than your workstation screen. If you choose a larger template, MENUGEN redisplays screen 3-27 so you can specify another template. If you don't know your screen size, choose item 2, CURRENT. It automatically displays the largest template that can fit on your screen.

When a template appears, you enter the help screen text exactly as you want it to appear and then transmit the filled-in template to MENUGEN.

After you've transmitted the screen, you see screen 3-28, which asks you if you want to specify more help screens:

SCREEN 3-28

```
                                                   MG12
              CREATE HELP SCREEN (CONT'D)
 1. ANOTHER HELP SCREEN FOR SAME ITEM
 2. HELP SCREEN FOR SAME ITEM
 3. HELP SCREENS ARE COMPLETE, END CREATE OPERATION
                                 ENTER SELECTION NUMBER: __
```

Your choices in this menu include:

`1. ANOTHER HELP SCREEN FOR SAME ITEM`

More help screens for the current item

If you have another help screen to add to the current item, choose this item. It redisplays screen 3-26, ready for you to add another help screen, with the ITEM field unchanged.

`2. HELP SCREEN FOR SAME ITEM`

Help screens for the next item

If you want to move on to the next item in your menu, choose this item. It redisplays screen 3-26, ready for you to add another help screen, but this time with the ITEM field increased by 1.

`3. HELP SCREENS ARE COMPLETE. END CREATE OPERATION`

Writing menu to the destination file and going to the home screen

If you want to end the MENUGEN create operation at this time, choose this item. It writes your menu to the menu module on the destination file and then displays screen 3-2, the MENUGEN home screen.

NOTE:

The following items (2 to 7) refer to screen 3-26.

2. USE AN EXISTING HELP SCREEN

Specifying an SFG-created screen for a help screen

If you want to specify an SFG-created screen as your next help screen, choose this item. MENUGEN needs to know the name of the screen you want it to retrieve, and so it displays screen 3-29:

SCREEN 3-29

```
                                                    MG1002S

        WHAT IS THE HELP SCREEN NAME?

INSERT NAME: ________
```

Fill in this screen with the name of the screen you want to become your help screen and then press XMIT. Screen 3-28 then appears, asking you if you want to add more help screens; you respond to that screen as described in item 1 of screen 3-28.

3. NO HELP FOR ITEM

Specifying the NO HELP message

If you don't want to provide any help screens for an item, choose this item. When the menu is later executed, an operator attempting to call a help screen for the item sees this screen:

```
                                                    MG1003S

NO HELP PROVIDED FOR THIS ITEM

                                    PRESS TRANSMIT TO CONTINUE
```

This screen also appears by default when you don't specify any help screen at all. For example, if you terminate MENUGEN after specifying help screens for only the first three items of a 6-item menu, this screen automatically appears if you ask for a help screen for items 4, 5, or 6.

4. DISPLAY MENU SCREEN

Displaying the menu screen

If you want to see the menu screen for your menu, this item causes it to appear. If your menu screen was created by SFG, MENUGEN first asks you where it resides, by using screen 3-22, before retrieving and displaying it. When you've finished reading it, press XMIT to get to screen 3-26.

5. DISPLAY ACTION TABLE

Displaying the action table

If you want to see a complete list of your menu's action table, choose this item. MENUGEN displays the list in a format similar to that of screen 3-21. If your actions take up more than one screen, MENUGEN displays each screen and waits until you press XMIT before going on to the next screen. After the last screen has been displayed, screen 3-26 appears.

6. WRITE MENU MODULE

Making a backup copy of the menu

If you want a backup copy of the work you've done so far on the menu, choose this item. It puts a copy of your menu in the destination file, which you can later complete if necessary by using the MENUGEN modify function. Screen 3-26 then appears.

7. HELPS ARE COMPLETE. END CREATE

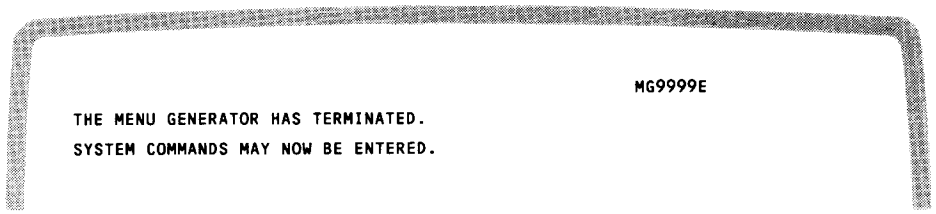
Writing a menu to the destination file and going to the home screen

If you want to end the MENUGEN create operation at this time, choose this item. It writes your menu to the menu module on the destination file and then displays screen 3-2, the MENUGEN home screen.

3.14. TERMINATING MENUGEN

What you see when MENUGEN ends

When you've created and stored your menu, you terminate MENUGEN by selecting item 4 of the MENUGEN home screen. You then see this screen:



This screen indicates that MENUGEN has given workstation control back to interactive services. You are now ready to use your new menu, although you can call MENUGEN again at any time to modify or display it as you wish.

4. Modifying a Menu

Besides creating your own menus, you can easily modify them by using MENUGEN. You may want to modify a working menu to fit the changing needs of your operators, or you may want to complete a new menu that you created partially with the MENUGEN create operation. In either case, MENUGEN allows you to change a menu's menu screen, action table, or help screens. Additionally, you can replace the menu screen or help screens with screens created by the screen format generator (SFG).

The MENUGEN scratch file

To avoid wiping out menus unintentionally, all changes go on a scratch file. Only when you specifically ask it to does MENUGEN copy this file, with your changes, to your destination file.

Like the create operation, the MENUGEN modify operation itself makes extensive use of menus and help screens. We recommend that you familiarize yourself, if you haven't already done so, with the procedure for creating a menu, as explained in Section 3.

The MENUGEN home screen

Before you begin modifying a menu, you have to display the MENUGEN home screen (screen 3-2). If you're already in MENUGEN, it's easy because that screen is the common returning point for all major MENUGEN operations (modify included). If not, recall that the MENUGEN workstation command executes MENUGEN, which displays screen 3-2 as its first screen. Let's look at the home screen once again (screen 4-1):

SCREEN 4-1

```
                                                        MG01

                    MENU GENERATOR HOME SCREEN

1. CREATE A NEW MENU MODULE
2. MODIFY AN EXISTING MENU MODULE
3. DISPLAY AN EXISTING MENU MODULE
4. END MENU GENERATOR

                                        ENTER SELECTION NUMBER 2
```

The four stages of menu modification

As you can see, item 2 of screen 4–1 is the modify operation. Like menu creation, you modify a menu in four stages:

1. Identifying the existing menu to be modified
2. Modifying its menu screen
3. Modifying its action table
4. Modifying its help screens

You have to identify an existing menu to be modified. After you've identified it, you can move about the menu, modifying any or all of its parts in no particular order. All changes go to that menu until you return to the MENUGEN home screen.

NOTE:

You can end MENUGEN at any time by pressing function key F15. This is useful when, for example, you decide you don't want to finish a menu that you're working on. Or you may want to quit working on a menu now but return to it later. When you press the F15 key, two things happen:

1. *You see a screen telling you that MENUGEN has terminated.*
2. *You get one or more system messages issued by MENUGEN. To see them, put your workstation in SYSTEM MODE.*

4.1. IDENTIFYING A MENU

To modify a menu, you tell MENUGEN:

Menu name

1. The name of the menu to be modified

The file and volume name

2. The file and volume on which the menu module resides

The secondary file name

3. The name of a secondary file

A secondary file

Specifying a secondary file is an optional step. To understand what the secondary file is for, recall that there are two ways of providing a menu screen: either by keying it in with MENUGEN or by creating a separate screen format outside of your menu and having the menu use that format as its menu screen. The same choice exists for help screens, to be part of the menu module or to lie outside of it. The MENUGEN modify operation gives you the options of displaying your menu's help and menu screens even if they aren't included in your menu module. To find them, MENUGEN has the capability of searching up to two files. The first is always the YFMT system file on your SYSRES volume. If a screen isn't there, MENUGEN then searches the secondary file if you choose to specify one. This 2-step search makes it possible, in some cases, to keep a menu in one file and to keep selected screens, not at all attached to the menu, in an entirely different file.

Identifying a menu and menu file

First, to let you specify the menu to be modified, MENUGEN displays screen 4-2:

SCREEN 4-2

```
                                                    MG0102S
                    MENU MODULE ACCESS INFORMATION
WHAT IS THE NAME, FILE LBL AND VSN OF THE MENU MODULE
THAT YOU WANT TO MODIFY?  ENTER BELOW.
MENU MODULE NAME: ________
FILE LBL: _____________________________________________
VSN: ______
```

Identifying a secondary file

Fill in screen 4-2 with the name of the menu you want to modify and the name and volume serial number of its library file. Next, MENUGEN asks you to specify your secondary file by using screen 4-3:

SCREEN 4-3

```
                                                    MG0102S1
                        OPEN SECONDARY FILE
IF YOU WANT TO DISPLAY YOUR MENU SCREEN AND/OR HELP
SCREEN(S) THAT WERE CREATED BY THE SCREEN FORMAT GENERATOR,
THEN THE FILE THAT IS USED TO SEARCH FOR THEM IS $Y$FMT
ON THE RES PACK. A SECONDARY FILE CAN ALSO BE SEARCHED
IF YOU SUPPLY THE REQUESTED INFORMATION BELOW. BY
JUST PRESSING TRANSMIT (DON'T SUPPLY THE INFORMATION)
ONLY $Y$FMT ON THE RES PACK WILL BE USED.
FILE LBL: _____________________________________________
VSN: ______
```

If you transmit screen 4-2 without filling in any of the blanks, MENUGEN assumes that your secondary file is YFMT. If so, YFMT becomes the only file that MENUGEN searches for external screens. Table 4-1 summarizes which files MENUGEN searches for your menu and your SFG-created menu/help screens.

Table 4–1. Files Searched by MENUGEN, Modify Operation

SFG-created Screen File (screen 4–3)	Menu File (screen 4–2)	
	YFMT	User File 1
YFMT	Menu: YFMT SFG screens: YFMT alone	Menu: User file 1 SFG screens: YFMT alone
User file 2	Menu: YFMT SFG screens: YFMT then user file 2	Menu: User file 1 SFG screens: YFMT then user file 2. (User file 1 can be the same as user file 2.)

Once you've transmitted this screen, MENUGEN begins the menu modify operation by displaying screen 4–4:

Modify options

SCREEN 4–4

```
                                                    MG25
                    MODIFY OPTIONS
1. MODIFY MENU SCREEN
2. MODIFY ACTION TABLE
3. MODIFY HELP SCREEN
4. END MODIFY OPERATION - WRITE MODIFIED MENU MODULE TO FILE
5. END MODIFY OPERATION - DON'T WRITE MODIFIED MODULE TO FILE
                                    ENTER SELECTION NUMBER: __
```

Your choices in screen 4–4 include:

1. MODIFY MENU SCREEN

Modifying the menu screen

Choose this item if you want to modify your menu screen. In response, MENUGEN displays screen 4–6 and menu screen modification proceeds as we describe in 4.2.

2. MODIFY ACTION TABLE

Modifying the action table

Choose this item if you want to modify your action table. In response, MENUGEN displays screen 4–8 and action table modification proceeds as we describe in 4.3.

3. MODIFY HELP SCREEN

Modifying the help screens

Choose this item if you want to modify your help screens. In response, MENUGEN displays screen 4–18 and help screen modification proceeds as we describe in 4.8.

4. END MODIFY OPERATION - WRITE MODIFIED MODULE TO FILE

Writing a modified menu to a file and going to the home screen

Choose this item if you've finished modifying your menu. In response, MENUGEN copies your modified menu from the MENUGEN scratch file to a file you designate by using screen 4–5:

SCREEN 4-5

```
                                                    MG2504S
        MODIFIED MENU MODULE OUTPUT INFORMATION
ENTER THE NAME, FILE LBL AND VSN OF THE MODIFIED MENU MODULE
  FILE LBL DEFAULTS TO $Y$FMT
  VSN DEFAULTS TO RES
NAME: ________
FILE LBL: _____________________________________________
VSN: ______
```

Fill in screen 4–5 with the name under which you want your menu written to its destination file, the name of the file, and its volume serial number. This feature allows you to create copies of a menu that differ in some way from the original. Of course, if you want to overwrite the original menu, you specify its menu name, file name, and volume serial number. After it finishes writing the menu, MENUGEN returns to the MENUGEN home screen.

5. END MODIFY OPERATION - DON'T WRITE MODIFIED MODULE TO FILE

Going to the home screen without writing to a destination file

Choose this item if you want to end the modify operation without writing your modified menu to the destination file. MENUGEN returns to the MENUGEN home screen, the contents of your menu scratch file are lost, and no changes are made to any menu on any other file.

4.2. MODIFYING THE MENU SCREEN

You can modify your menu screen in several ways, depending on how you originally created it:

- If you originally created the menu screen with MENUGEN, as part of the menu, you can either keep it while changing its contents or replace it with another screen created separately by the SFG.
- If you originally created the menu screen with SFG, you can replace it with either another SFG-created screen format or a new menu screen created by MENUGEN within the modify operation. You can't use MENUGEN to change the contents of a SFG-created screen; only SFG itself can do that.
- You can even create an entirely new menu screen.

When you indicate to MENUGEN that you want to modify your menu screen, you see screen 4–6:

SCREEN 4-6

```
                                                        MG21
                    MODIFY MENU SCREEN OPTIONS
1. CHANGE TEXT OF MENU GENERATOR CREATED MENU SCREEN
2. CREATE MENU SCREEN (SCREEN FORMAT GENERATOR)
3. CREATE MENU SCREEN (MENU GENERATOR)
4. DISPLAY MENU SCREEN
5. END MODIFY MENU SCREEN
                                    ENTER SELECTION NUMBER: __
```

Your choices in screen 4–6 include:

1. CHANGE TEXT OF MENU GENERATOR CREATED MENU SCREEN

Changing a menu screen created by the MENUGEN command

If you want to change the text of a menu screen created by MENUGEN, you choose this item. MENUGEN displays the text of the menu screen exactly as it appears in use. At this point, you can use the cursor positioning and text handling (INSERT, DELETE, ERASE, etc) keys on your workstation to make any changes you want directly on the displayed screen. When you've finished your changes, you position your workstation cursor *after the last character on the screen* and then transmit the screen to MENUGEN, which writes it to the scratch file with all the changes you made. After this, MENUGEN asks you, through screen 4–4, if you want to change anything else in your menu.

2. CREATE MENU SCREEN (SCREEN FORMAT GENERATOR)

Replacing a menu screen created by the SFG

If you want to replace the menu screen with one you created with SFG, choose this item. You can do this whether the original screen was created by MENUGEN or by SFG. The new screen can reside on either YFMT or your secondary file. When you choose this item, you see screen 4–7:

SCREEN 4-7

```
                                                    MG2102S

                        REPLACE MENU SCREEN

ENTER THE NAME OF THE MENU SCREEN WHICH IS TO REPLACE
THE CURRENT MENU SCREEN.

INSERT NAME: ________
```

Simply enter the name of the screen format that you want to replace the current menu screen. Remember that while this format need not exist at the time you modify your menu with MENUGEN, it has to be present, in the same file as the menu module, at the time you actually use the menu. After you transmit screen 4–7, MENUGEN asks you, through screen 4–4, if you want to change anything else in your menu.

3. CREATE MENU SCREEN (MENU GENERATOR)

Creating a new menu screen

If you want to create an entirely new menu screen by using MENUGEN, choose this item. MENUGEN asks you what size screen you want by displaying screen 3–6 and then displays the template of that size onto which you enter your menu screen text. To refresh your memory on how this is done, refer to 3.3. After you enter the screen text and transmit the screen, MENUGEN asks you, through screen 4–4, if you want to change anything else in your menu.

4. DISPLAY MENU SCREEN

Displaying the menu screen

If all you want to do is look at the menu screen, choose this item. It displays the menu screen without allowing you to make any changes. When you finish reading it and press XMIT, screen 4–6 returns.

5. END MODIFY MENU SCREEN

Changing another part of a menu

If you don't want to modify or display your menu screen, choose this item. It simply causes MENUGEN to ask you, through screen 4–4, if you want to change anything else in your menu.

4.3. MODIFYING THE ACTION TABLE

To modify an action table, choose item 2 of screen 4–4. You then see screen 4–8:

SCREEN 4–8

```
                                                    MG22

              MODIFY ACTION TABLE OPTIONS

1. CHANGE EXISTING ACTION(S)
2. ADD ACTION(S) TO AN ITEM WITH NO ACTIONS
3. DELETE ALL ACTIONS FOR AN ITEM
4. DISPLAY ACTION TABLE
5. DISPLAY MENU SCREEN
6. END MODIFY ACTION TABLE

                                  ENTER SELECTION NUMBER: __
```

We'll discuss each of these choices in turn.

4.4. CHANGING EXISTING ACTIONS

If you want to change existing actions in your action table, choose item 1 of screen 4–8. You can change the actions for one item at a time; MENUGEN asks you which item by displaying screen 4–9:

SCREEN 4–9

```
                                                    MG22Ø1S
ENTER THE ITEM NUMBER WHICH CORRESPONDS TO
THE ACTION OR ACTIONS THAT YOU WANT TO CHANGE.
ENTER ITEM NUMBER: 1
```

How an action table is displayed for changes

Enter the number of the item whose actions you want to change, as we did in screen 4–9 by specifying item 1. In response, you see screen 4–10:

SCREEN 4–10

```
  CHANGE OR ADD ACTIONS FOR ITEM Ø1 ON THIS SCREEN (MG22Ø1S1)
  ORDER THEM CHRONOLOGICALLY. TRANSMIT TO GET NEXT 8 ACTIONS.
ACTION TYPE                  ACTION
     S       RV PAYROL1:(PAYLOAD,PACKØ1)_________________
     S       RV PAYROL2:(PAYLOAD,PACKØ1),,IF=MASTERFILE__
     -       ----------------------------------------
     -       ----------------------------------------
     -       ----------------------------------------
     -       ----------------------------------------
     -       ----------------------------------------
     -       ----------------------------------------
     ENTER 'C' TO CANCEL ALL CHANGES TO ITEM _
```

Of course your screen will differ. Screen 4–10 is just an example of how MENUGEN displays actions to be changed. It lists every action currently linked to your current item. Let's say you want to change the second action; just key in the change directly on the screen:

```
  CHANGE OR ADD ACTIONS FOR ITEM Ø1 ON THIS SCREEN (MG22Ø1S1)
  ORDER THEM CHRONOLOGICALLY. TRANSMIT TO GET NEXT 8 ACTIONS.
ACTION TYPE                  ACTION
     S       RV PAYROL1:(PAYLOAD,PACKØ1)_________________
     S       RV PAYROL2:(PAYLOAD,PACKØ1),,IF=TRANSFILE __
     -       ----------------------------------------
     -       ----------------------------------------
     -       ----------------------------------------
     -       ----------------------------------------
     -       ----------------------------------------
     -       ----------------------------------------
     ENTER 'C' TO CANCEL ALL CHANGES TO ITEM _
```

Make the changes you want and then position the cursor after the last change on the screen. When you press XMIT, MENUGEN applies your changes to the action table.

NOTES:

1. *If you change your mind and don't want to make the changes you've typed in, enter 'C' on the last line and press XMIT. You then return to screen 4–8.*

2. *Remember that the only three allowable entries for ACTION TYPE in screen 4–10 are S for a system command, D for input data, and M for a menu function command.*

What happens after you transmit your action table changes depends on how many more actions remain for your item:

More actions to change?

- If there are more actions, MENUGEN displays them for you to make any changes you wish. This procedure is repeated until MENUGEN reaches the last action for your item.

- If there are no more actions, you see screen 4–11:

SCREEN 4-11

```
                                                    MG2201S2

        CHANGE AN ACTION

THERE ARE NO MORE ACTIONS FOR THIS ITEM.
DO YOU WANT TO ADD SOME?   (Y)ES,(N)O _
```

Screen 4–11 gives you the option of adding new actions to the list of those currently linked to your item. Your reply to this screen can be:

- **Y**, to indicate that you want to add new actions. MENUGEN then displays a screen that looks like screen 4–10 but contains eight blank lines. You enter your new actions on these lines and press XMIT; MENUGEN adds them to the list of existing actions in the order you entered them. After transmitting the screen, you see screen 4–11 again.

- **N**, to indicate that you don't want to add new actions or that you've finished adding them. You then return to screen 4–8.

4.5. ADDING ACTIONS

Adding new actions

If you want to add actions to an item that currently has no actions, choose item 2 of screen 4–8. You can add actions to one item at a time; MENUGEN asks you which item by displaying screen 4–12:

SCREEN 4-12

```
                                                    MG2202S
ENTER ITEM NUMBER WHICH CORRESPONDS TO ACTION(S) TO BE ADDED.
ENTER ITEM #: 2
```

Enter the number of the item for which you want to add items, as we did in screen 4–9 by specifying item 2. In response, you see screen 4–13:

SCREEN 4–13

```
                                                    MG2202S1

    ADD ACTIONS FOR ITEM 02 ON THIS SCREEN.
    ORDER THEM CHRONOLOGICALLY.
ACTION TYPE                  ACTION
    _       ---------------------------------------------
    _       ---------------------------------------------
    _       ---------------------------------------------
    _       ---------------------------------------------
    _       ---------------------------------------------
    _       ---------------------------------------------
    _       ---------------------------------------------
    _       ---------------------------------------------
    ENTER 'C' TO CANCEL ALL ADDITIONS TO ITEM _
```

Screen 4–13 displays blank lines that you fill in with new actions. If an action takes up more than 44 characters, you continue it in the ACTION field of the next line, taking care to leave that line's ACTION TYPE field blank.

NOTE:

If you've keyed in new actions on screen 4–13 and then decide you don't want to use them after all, move the cursor to the last line of the screen, enter C, and press XMIT. This wipes out the changes you entered and returns you to screen 4–8.

After you finish adding actions and transmit the screen, you get screen 4–14:

SCREEN 4–14

```
                                                    MG2202S2

                        ADD ACTIONS
    DO YOU WANT TO ADD MORE ACTIONS TO THIS ITEM?
                              (Y)ES, (N)O  _
```

More actions to add?

Use this menu if you want to add any more actions or signal MENUGEN that you've finished adding actions. If you answer Y, screen 4–13 reappears to let you specify another set of actions for the current item; this is useful when the number of actions for an item exceeds one screen. If you answer N, screen 4–8 returns.

4.6. DELETING EXISTING ACTIONS

Deleting actions

If you want to delete existing actions from your action table, choose item 3 of screen 4–8. With this option, you delete all the actions for a specified item.

NOTE:

If you want to delete only some actions from an item, use the change option (4.4) and blank out the items you want deleted.

You can delete actions for one item at a time; MENUGEN, through screen 4–15, asks you which item:

SCREEN 4-15

```
                                                    MG2203S

   ENTER ITEM NUMBER WHICH CORRESPONDS TO
   ACTION OR ACTIONS TO BE DELETED
   ENTER ITEM NUMBER: 2

IF YOU DON'T WANT TO DELETE ACTIONS ENTER 'N' HERE _
```

Enter the number of the item whose actions you want to change, as we did in screen 4–15 by specifying item 2. In response, you see screen 4–16:

SCREEN 4-16

```
                                                    MG2203S1

                        DELETE ACTION
ALL ACTIONS FOR ITEM 02 HAVE BEEN DELETED.
DELETE ALL ACTIONS FOR ANOTHER ITEM? - (Y)ES, (N)O
```

More actions to delete?

Screen 4–16 is simply an informational screen that tells you MENUGEN has deleted all actions for the specified item. It also asks you if you want to delete the actions for any other item (reply Y) or return to screen 4–8 upon finishing the delete operation (reply N).

4.7. OTHER ACTION TABLE FUNCTIONS

Three other options shown on screen 4–8 are available to you. In order of appearance, these include:

```
4. DISPLAY ACTION TABLE
```

Displaying the action table

Choose this item if you want to display the action table without modifying it in any way. This item lists the action table in a format similar to that of screen 4–17:

SCREEN 4-17

```
                                                    MG0904T
ITEM #  ACTION TYPE  ACTION
 1           S       RV PAYROL1:(PAYLOAD,PACK01)
             S       RV PAYROL2:(PAYLOAD,PACK01),,IF=MASTERFILE



TYPES: (S)YS COMMAND, (D)ATA, (M)ENU COMMAND
```

We've borrowed the action table from screen 4–10 to show you how it would be displayed here. The screen itself is the same as the one you'd see if you were creating the menu. The legend at the bottom of the screen shows the three familiar action types.

After you've finished reading the screen, press XMIT. If your actions take up more than one screen, press XMIT after reading each screen to get the next screen. After you've responded to the last screen, you see screen 4–8 again.

```
5. DISPLAY MENU SCREEN
```

Displaying the menu screen

If you want to see the menu screen, choose this item. After you've finished reading the screen, press XMIT to return to screen 4–8.

```
6. END MODIFY ACTION TABLE
```

Changing another part of a menu

If you are finished modifying your action table, choose this item. MENUGEN then returns you to screen 4–4.

4.8. MODIFYING HELP SCREENS

With the modify operation of MENUGEN, you can change any help screen.

NOTE:

When asked to specify a help screen by the number of its item and you want the summary help screen, specify item 00.

To modify your menu's help screens, choose item 3 of screen 4–4. You then see screen 4–18:

SCREEN 4-18

```
        MODIFY HELP SCREEN(S) FOR AN ITEM          MG23
1. ADD SCREEN FORMAT GENERATOR (SFG) CREATED HELP SCREEN
2. ADD MENU GENERATOR (MG) HELP SCREEN (TO BE CREATED NOW)
3. DELETE ALL HELP SCREENS FOR AN ITEM
4. DELETE A SPECIFIC HELP SCREEN FOR AN ITEM
5. CHANGE TEXT OF EXISTING HELP SCREEN THAT WAS CREATED BY MG
6. DISPLAY HELP SCREEN
7. END MODIFY HELP SCREENS
                                  ENTER SELECTION NUMBER: __
```

Your choices for screen 4–18 include:

1. ADD SCREEN FORMAT GENERATOR (SFG) CREATED HELP SCREEN

Adding an SFG-created help screen

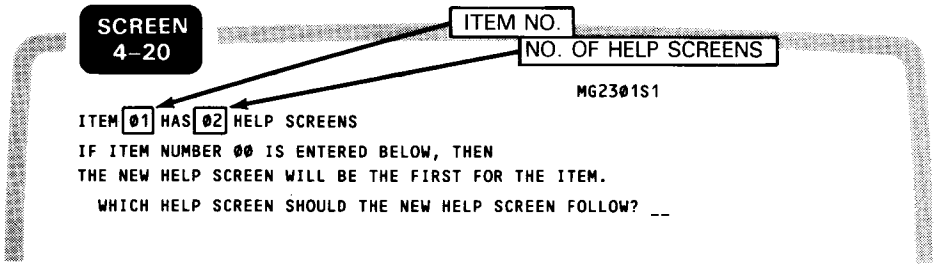
Choose this item to add a help screen created by SFG to an item. To let you specify which help screen, MENUGEN displays screen 4–19:

SCREEN 4-19

```
                                                    MG2301S
ADD HELP SCREEN CREATED BY THE SCREEN FORMAT GENERATOR
WHICH ITEM SHOULD HELP SCREEN BE ADDED TO? __
```

Fill in screen 4–19 with the number of the item to which you wish to add a help screen and then press XMIT. If the item already has existing help screens, you then see screen 4–20. If no help screens currently exist, MENUGEN skips to screen 4–21.

SCREEN 4–20



The first line of screen 4–20 shows you how many help screens are present for your current item. Below that, enter the number of the help screen you want your new help screen to follow and then press XMIT. For example, you'd enter 00 if the new screen is the first in the series, 01 if it follows the first screen, 02 if it follows the second screen, and so on.

After screen 4–19 or screen 4–20 appears, MENUGEN displays screen 4–21 for you to specify the name of the screen format to be used as your new help screen:

SCREEN 4–21

```
                                                    MG1002S

WHAT IS HELP SCREEN NAME?
  INSERT NAME: ________
```

In the space provided in screen 4–21, enter the name of the screen format you want as your new help screen and then press XMIT. MENUGEN then returns to screen 4–18.

2. ADD MENU GENERATOR (MG) HELP SCREEN (TO BE CREATED NOW)

Adding a MENUGEN-created help screen

Choose this item if you want to add a help screen by using MENUGEN. In this operation, you first specify when the new help screen is to appear in the menu and then enter its text directly on your workstation. If help screens already exist for the item, you can tell MENUGEN to insert the new help screen in any position relative to them. To begin with, MENUGEN asks you for which item you want to add the screen, using screen 4–22:

SCREEN 4–22

```
                                                    MG2301S

ADD HELP SCREEN CREATED BY THE MENU GENERATOR
WHICH ITEM SHOULD HELP SCREEN BE ADDED TO? __
```

Enter an item number in screen 4–22 and then press XMIT. If the item already has help screens, MENUGEN displays screen 4–23, to let you specify exactly where you want your new screen inserted. If no help screens exist, MENUGEN skips to screen 4–24.

SCREEN 4-23

```
                                                    MG2302S1

ITEM 01 HAS 02 HELP SCREENS.
IF 00 IS ENTERED BELOW, THEN
THE NEW HELP SCREEN WILL BE THE FIRST FOR THE ITEM.
   WHICH HELP SCREEN SHOULD THE NEW HELP SCREEN FOLLOW? __
```

The first line of screen 4–23 shows you how many help screens are present for your current item. Below that, enter the number of the help screen you want your new help screen to follow and then press XMIT. For example, you'd enter 00 if the new screen is the first in the series, 01 if it follows the first screen, 02 if it follows the second screen, and so on.

After screen 4–22 or 4–23 has been transmitted, you create the help screen. The procedure is similar to the one described in 3.13; screen 4–24 appears to ask you what size screen you want:

SCREEN 4-24

```
                                                    MG03

          WHAT SIZE SCREEN IS DESIRED?

1. UNIVERSAL (12X64)              5. 12X80
2. CURRENT                        6. 16X80
3. 16X64                          7. 24X80
4. 24X64

                  ENTER SELECTION NUMBER: __
```

After you choose a screen size and transmit screen 4–24, a template appears in which you enter the text of the help screen. After positioning the cursor beyond the last character of the help screen, press XMIT to send the screen to MENUGEN for inclusion in your menu. You then return to screen 4–18.

3. DELETE ALL HELP SCREENS FOR AN ITEM

Deleting all help screens for an item

Choose this item if you want to delete all help screens for a particular item. MENUGEN, through screen 4–25, asks you which item:

SCREEN 4-25

```
                                                    MG2303S

DELETE ALL HELP SCREENS
ENTER THE ITEM NUMBER TO WHICH THE DELETE IS TO BE DONE. __

IF YOU DON'T WANT TO DELETE ANY HELP SCREENS, ENTER 'N' HERE _
```

NOTE:

On this screen and on some screens to follow, you can choose not to delete any help screens by moving the cursor to the last line and keying in N where the screen tells you to. Upon transmitting the screen, you then return to screen 4–18.

Fill in screen 4–25 with an item number and then press XMIT. You then see screen 4–26:

SCREEN 4-26

```
                                        MG2303S3
   DELETE HELP SCREEN
ALL HELP SCREENS FOR ITEM 01
    HAVE BEEN DELETED


PRESS TRANSMIT TO CONTINUE
```

Screen 4–26 is a purely informational screen that tells you all the help screens for the specified item have been deleted. Press XMIT to return to screen 4–18.

4. DELETE A SPECIFIC HELP SCREEN FOR AN ITEM

Deleting a specific help screen for an item

Choose this item when you want to delete a specific help screen from those linked to a menu item, when that item has more than one help screen. MENUGEN first displays screen 4–27:

SCREEN 4-27

```
                                                MG2304S
DELETE SPECIFIC HELP SCREEN
ENTER THE ITEM NUMBER TO WHICH THE DELETE IS TO BE DONE. __

IF YOU DON'T WANT TO DELETE ANY HELP SCREENS ENTER 'N' HERE _
```

Fill in screen 4–27 with the number of the item your help screen is linked to. When you press XMIT, MENUGEN displays screen 4–28:

SCREEN 4-28

```
                                                MG23034S1
ITEM 01 HAS 02 HELP SCREENS
WHICH HELP SCREEN DO YOU WANT TO DELETE? __

IF YOU DON'T WANT TO DELETE ANY HELP SCREENS ENTER 'N' _
```

The first line of screen 4–28 tells you how many help screens are linked to the specified item. Below that, enter the number of the help screen you want deleted (enter 01 for the first screen, 02 for the second screen, etc.) and then press XMIT. MENUGEN deletes the help screen and then returns you to screen 4–18.

```
5. CHANGE TEXT OF EXISTING HELP SCREEN THAT WAS CREATED BY MG
```

Changing a MENUGEN-created help screen

Choose this item to change the text of a help screen created by MENUGEN. In this operation, you first specify which help screen you want to change and then make the changes directly on the screen itself when MENUGEN displays it. To begin with, through screen 4–29, MENUGEN asks you for which item you want to change a help screen:

SCREEN 4-29

```
                                                    MG2305S
CHANGE TEXT OF MENU GENERATOR CREATED HELP SCREEN
WHICH ITEM CONTAINS THE HELP SCREEN TO BE CHANGED. __
```

Fill in screen 4–29 with the number of the item whose help screen or screens you want to change and then press XMIT. MENUGEN displays the help screen for your inspection. Make the changes you want directly on the screen, position the cursor after the last changed character on the screen, and then press XMIT. This updates the help screen.

If there are more help screens linked to the item, you next see screen 4–30; if not, you see screen 4–31:

SCREEN 4-30

```
          CHANGE HELP SCREEN TEXT        MG27
1. CHANGE THE NEXT HELP SCREEN FOR ITEM Ø1
2. CHANGE THE HELP SCREENS FOR ANOTHER ITEM
3. END CHANGE HELP SCREEN
               ENTER SELECTION NUMBER __
```

For this screen, answer:

1 to change the next help screen for the item.

2 to return to screen 4–29 if you want to change the help screen for another item.

3 to return to screen 4–18.

SCREEN 4-31

```
                                        MG28
          CHANGE HELP SCREEN
1. CHANGE HELP SCREENS FOR ANOTHER ITEM
2. END CHANGE HELP SCREEN
                    ENTER SELECTION NUMBER __
```

For this screen, answer:

1 to return to screen 4–29 if you want to change the help screen for another item.

2 to return to screen 4–18.

NOTE:

MENUGEN allows you to change only those help screens that were created by MENUGEN. If it finds that a help screen is a screen format, it displays an error message. In that case, you press XMIT to continue. If there are more help screens for the item you're working on, you see screen 4–30. If there aren't any more help screens for the item, you see screen 4–31. (You can change a help screen that was created as a screen format by using the screen format generator.)

6. DISPLAY HELP SCREENS

Displaying help screens

If all you want to do is display help screens, choose this item. MENUGEN then asks you, with screen 4–32, which help screen to display:

SCREEN 4-32

```
                                        MG2306S
DISPLAY HELP SCREENS
ENTER ITEM # WHOSE HELP SCREENS YOU WANT TO DISPLAY. __
```

Fill in screen 4–32 with the number of the item whose help screen or screens you want displayed. After you transmit screen 4–32, MENUGEN displays the help screen and waits until you press XMIT to return to screen 4–18. If there are two or more help screens for an item, press XMIT after each screen to display the next screen in the series.

7. END MODIFY HELP SCREENS

Changing another part of a menu

When you've finished modifying help screens, choose this item. In response, MENUGEN asks you, through screen 4–34, if you want to change anything else in your menu.

5. Displaying a Menu

The third MENUGEN operation available to you is the display operation. With it, you can display any or all parts of an existing menu. You can also direct MENUGEN to display the menu on your workstation screen, print a copy of it on the system printer, or output it to both devices.

Like the create and modify operations, the MENUGEN display operation makes extensive use of menus and help screens. To help you understand what this operation does, we recommend first that you be familiar with the procedure for creating a menu, as explained in Section 3.

NOTE:

You can end MENUGEN at any time by pressing function key F15. This is useful when, for example, you decide you don't want to finish a menu that you're working on. Or you may want to quit working on a menu now but return to it later. When you press the F15 key, two things happen:

1. *You see a screen telling you that MENUGEN has terminated.*

2. *You get one or more system messages issued by MENUGEN. To see them, put your workstation in SYSTEM MODE.*

Before you display a menu, you have to display the MENUGEN home screen (screen 3–2). If you're already in MENUGEN, it's easy because that screen is the common returning point for all major MENUGEN operations (display included). If not, recall that the MENUGEN workstation command executes MENUGEN, which displays screen 3–2 as its first screen. Let's look at the home screen once again (screen 5–1):

The home screen

SCREEN 5-1

```
                                                        MG01

                MENU GENERATOR HOME SCREEN

1. CREATE A NEW MENU MODULE
2. MODIFY AN EXISTING MENU MODULE
3. DISPLAY AN EXISTING MENU MODULE
4. END MENU GENERATOR

                                      ENTER SELECTION NUMBER 3
```

Choosing the display operation

As you can see, item 3 of screen 5–1 is the display operation. To begin that operation, you must identify the menu you want to display by using screen 5–2:

Identifying a menu and a menu file

SCREEN 5-2

```
                                                        MG0103S

              WHERE IS MENU MODULE?

ENTER REQUESTED INFORMATION BELOW. IF THE FILE LBL
IS NOT SUPPLIED, THEN $Y$FMT IS USED. IF THE VSN IS NOT
SUPPLIED, THEN RES IS USED. THE MODULE NAME MUST BE SUPPLIED.

MENU MODULE NAME: ________

FILE LBL:  ---------------------------------------------

VSN: ______
```

Identifying a secondary file

Fill in the name of the menu, along with the file and volume in which it resides, and then transmit the screen. MENUGEN then asks you if you want to specify a secondary file. To understand what the secondary file is for, recall that there are two ways of providing a menu screen: either by keying it in with MENUGEN or by creating a separate screen format outside of your menu and having the menu use that format as its menu screen. The same choice exists for help screens – to be part of the menu or to lie outside of it. The MENUGEN display operation lets you display your menu's help and menu screens even if they aren't included in your menu module. To find them, MENUGEN has the capability of searching up to two files. The first is always the YFMT system file on your SYSRES volume. When a screen isn't there, MENUGEN then searches the secondary file if you choose to specify one. You specify your secondary file with screen 5–3:

SCREEN 5–3

```
                                                        MG0103S1

                        OPEN SECONDARY FILE

IF YOU WANT TO DISPLAY YOUR MENU SCREEN AND/OR HELP
SCREEN(S) THAT WERE CREATED BY THE SCREEN FORMAT GENERATOR,
THEN THE FILE THAT IS USED TO SEARCH FOR THEM IS $Y$FMT
ON THE RES PACK. A SECONDARY FILE CAN ALSO BE SEARCHED
IF YOU SUPPLY THE REQUESTED INFORMATION BELOW. BY
JUST PRESSING TRANSMIT (DON'T SUPPLY THE INFORMATION)
ONLY $Y$FMT ON THE RES PACK WILL BE USED.

FILE LBL: _____________________________________________

VSN: ______
```

If you transmit the screen without filling in any of the blanks, MENUGEN assumes that your secondary file is YFMT. If YFMT is your secondary file, it becomes the only file that MENUGEN searches. Table 5–1 summarizes which files MENUGEN searches for your menu and your SFG-created menu/help screens.

Table 5-1. Files Searched by MENUGEN, Display Operation

SFG-created Screen File (screen 5-3)	Menu File (screen 5-2)	
	YFMT	User File 1
YFMT	Menu: YFMT SFG screens: YFMT alone	Menu: User file 1 SFG screens: YFMT alone
User file 2	Menu: YFMT SFG screens: YFMT then user file 2	Menu: User file 1 SFG screens: YFMT then user file 2. (User file 1 can be the same as user file 2.)

Identifying output devices

Before you see the menu, there's one more question for MENUGEN to ask you: Where do you want the output directed? Screen 5–4 asks:

SCREEN 5-4

```
                                                    MG30
      WHERE SHOULD MENU MODULE BE DISPLAYED TO?
1. WORKSTATION ONLY
2. PRINTER ONLY
3. BOTH WORKSTATION AND PRINTER
4. END DISPLAY OPERATION
                                  ENTER SELECTION NUMBER: __
```

Your available choices in screen 5–4 are:

`1. WORKSTATION ONLY`

The display menu screen

Choose this item if you want your menu to be displayed on your workstation.

`2. PRINTER ONLY`

The display action table

Choose this item if you want your menu to be printed on the printer.

`3. BOTH WORKSTATION AND PRINTER`

The display help screens

Choose this item if you want your menu output both to your workstation and printer. The display goes to the printer first.

4. END DISPLAY OPERATION

Choose this item if you want to terminate the MENUGEN display function. You then return to the MENUGEN home screen.

If you choose to display your menu, screen 5-5 asks you which part of it you want to see:

SCREEN 5-5

```
                                                          MG31
    WHAT PORTION OF THE MENU MODULE DO YOU WANT TO DISPLAY?
1. MENU SCREEN                    4. ENTIRE MENU MODULE
2. ACTION TABLE                   5. DISPLAY ANOTHER MENU MODULE
3. HELP SCREEN(S)                 6. END DISPLAY OPERATION
                                     ENTER SELECTION NUMBER: __
```

Your choices for screen 5-5 include:

1. MENU SCREEN

Displaying the menu screen

Choose this item if you want to display the menu screen for your menu. MENUGEN displays the screen and waits until you press XMIT to return to screen 5-5.

2. ACTION TABLE

Displaying the action table

Choose this item if you want to display the action table for your menu. That display appears like the one in screen 5-6:

SCREEN 5-6

```
                                                          MG0904T
ITEM #  ACTION TYPE  ACTION
 1           S       RV PAYROL1:(PAYLOAD,PACK01)
             S       RV PAYROL2:(PAYLOAD,PACK01),,IF=MASTERFILE


TYPES: (S)YS COMMAND, (D)ATA, (M)ENU COMMAND
```

We've borrowed the action table from screen 4-16 to show you how it would be displayed here. The screen itself is the same as the one you'd see if you were creating the menu. The legend at the bottom of the screen shows the three familiar action types used by the menu processor. If the action table takes up more than one screen, MENUGEN displays them all, replacing one screen with the next when you press XMIT. After the last screen, you return to screen 5-5.

3. HELP SCREEN(S)

Displaying the help screens

Choose this item if you want to display the help screens that are linked to your menu. After you transmit screen 5-5, you see screen 5-7:

SCREEN 5-7

```
                                                        MG32

        DISPLAY HELP SCREEN OPTIONS

1. DISPLAY SPECIFIC HELP SCREEN
2. DISPLAY ALL HELP SCREENS
3. END HELP SCREEN DISPLAY

                                  ENTER SELECTION NUMBER: __
```

Which help screens to display?

Screen 5-7 presents you with your options for displaying help screens. If you want to display the help screens for a specific item, you choose item 1. You then see screen 5-8:

SCREEN 5-8

```
                                                        MG32@1S

WHICH ITEM'S HELP SCREEN SHOULD BE DISPLAYED?
ENTER ITEM NUMBER __
```

Simply enter the number of the item whose help screen you want to see, and MENUGEN then displays the screen. (To get the summary help screen, enter item 00. If there's more than one help screen for an item, MENUGEN displays them all, one at a time.)

Of the other two options in screen 5-7, choose item 2 to display the help screens for all items in your menu, or choose item 3 if you want to end the help screen display operation and return to screen 5-5.

If you choose item 1 or 2, you'll return to screen 5-5 after the display is finished.

```
4. ENTIRE MENU MODULE
```

Displaying an entire menu

Choose this item to see your entire menu in this order: menu screen, action table, and all the help screens for every item. After each screen is displayed, go on to the next screen by pressing XMIT. After the last screen, you'll return to screen 5-5.

```
5. DISPLAY ANOTHER MENU MODULE
```

Getting another menu to display

If you've finished viewing your present menu and want to see another, choose this item. MENUGEN then returns to screen 5-2 to ask you for more information on the next menu you want to display.

```
6. END DISPLAY OPERATION
```

The END MENUGEN DISPLAY operation

If you've finished displaying menus, choose this item. It returns you to the MENUGEN home screen.

6. Using Menus with User Programs

6.1. GENERAL

Menus and consolidated data management

If your system has consolidated data management, you can create menus through which you can pass input data to your program. Though there are other ways to input data, menus offer several advantages, some of which we've already noted:

- Menus are easy to use.
- You can create and modify menus quickly and interactively.
- You can enforce security by limiting allowable responses to those you provide for in a menu's action table.
- You can create menus separately from the programs that use them, allowing for modular programming. Changes to menus or programs are also easy to make.

For example, if you wrote a COBOL program to accept input from punch cards, it's easy to adapt it to take the same input from a workstation by means of a menu. In fact, it would require only these changes to your software:

- Creating your menu with MENUGEN
- Changing your job control to link the COBOL program with your menu

Depending on your application, these changes are relatively easy to make, and you'll gain the advantages of menu use we listed previously. In this section, we guide you in making these changes, not only to programs written in COBOL, but also in FORTRAN, RPG II, and BAL.

6.2. CONCEPTS

Consolidated data management (CDM)

First, a short explanation of consolidated data management. Consolidated data management (CDM) is an OS/3 facility that handles the flow of data between your user programs and the various input/output devices supported by your system. All you need to do is make a formal request in your program to CDM, and it does the rest, moving the desired data to or from the device you want, whether that device is magnetic tape, card reader, disk — or even a workstation using a menu. That's because one feature of CDM is your ability to request the same operation, in the same way, for a wide variety of devices. It frees you of the burden of satisfying the device-dependent requirements for an operation.

User programs

How do programs use menus with CDM? The thing to keep in mind about menus is that a workstation running under control of a menu is treated by your program as a card file. To CDM itself, menus need special handling, just as any input/output device does. But to a program calling on CDM, a menu could just as well be a card file. To grasp better how menus are both like and unlike other files, look at Figure 6-1:

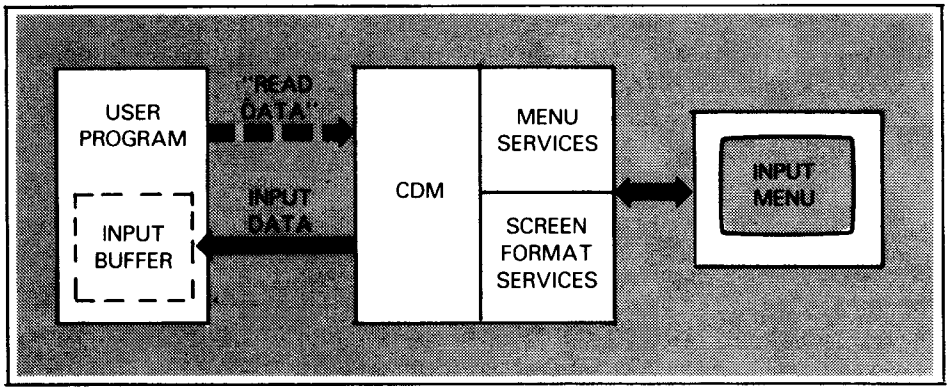


Figure 6-1. A Data Path between the User Program and a Menu

As Figure 6-1 shows, the path that data takes from the workstation to a user program travels across several OS/3 components. First, the user program calls a menu by issuing a request to CDM for data input — in effect, a request to *read* data from an input file, placing it in an input buffer set aside for that purpose.

Up to this point, the program could be asking for data from a card file. But CDM is aware, as the program is not, that this particular file is not on cards, but on an entirely different device with its own special characteristics. So it passes on the request to menu services.

Working hand in hand with screen format services, menu services take over the operator's workstation and display the menu screen. When the operator makes a reply, menu services search the action table corresponding to the reply. What happens next depends on the action:

How different action types pass input data

- If the action is input data, menu services pass it through CDM, unchanged, to the program input buffer.

- If the action is a SCREEN menu function command, menu services temporarily overlay the menu screen with the screen named in the command. Any input data that the operator enters on that screen is then passed through CDM to the program's input buffer in the same manner as if the action had been input data. Barring further actions for the menu item, the menu screen then reappears.

Again it's immaterial to the program where the data came from, just so the data arrives. That's because CDM and menu services combine to make the process of data input as transparent to the program as possible. You need to concern yourself with only two parts of Figure 6–1: what the workstation operator sees and what data is passed to your program. Since you already know how to create a menu, we concentrate on how your program interacts with CDM and, through CDM, with your menu.

6.3. PROGRAM CONSIDERATIONS

One of menu services' most flexible features is that you can use menus as input to programs written in any of four OS/3 languages: COBOL, RPG II, FORTRAN, and BAL. In this subsection, we'll explain the menu concepts and interfaces common to all four languages.

First the concepts: whatever language a program using menus is written in, CDM returns two types of information to the program – input data records and the EOF (end-of-file) indicator.

Input data

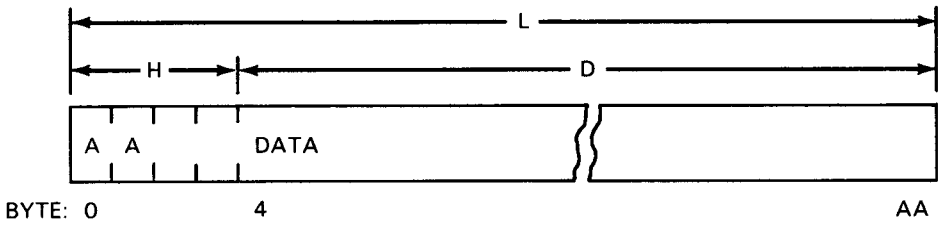
We've already defined input data as that string of data passed from a menu to a user program. It can take either of two forms: fixed-length record or variable-length record.

- Fixed-length records

  Your program may define the menu input file as having records of some fixed length. If a menu action passes data to the program, the data goes into a main storage buffer starting at the first byte in the buffer. If the data string is shorter than the buffer, CDM fills the remaining buffer space with blanks. If the data is longer than the buffer, CDM passes only as much of the data as fits in the buffer, truncating the rest.

- Variable-length records

  Your program may define the menu input file as having records whose length may vary, while never exceeding some maximum length. If a menu action passes data to the program, the data goes into the main storage buffer in the following format:

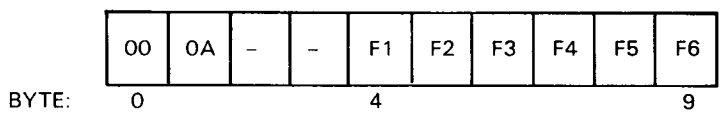


Format

In this diagram, H represents a 4-byte record header attached to the beginning of the record by CDM. Of these four bytes, only the first two need be considered by your program; they contain the record length (L) expressed in binary. Note that L gives the length of the entire record, header included, so assuming that the first byte of the record is 0 and that the record length is AA:

- the data itself begins at byte 4; and
- the length of the data is AA-4 bytes.

For example, look at the following record:



Example

Because the record length in the header is 10 (hexadecimal A), the data begins at byte 4 and extends six (10–4) bytes. You must provide for this record in your program by setting aside a data field large enough to hold the longest record the field will ever receive. In the case of input data actions, that figure is 76 bytes: 4 bytes for the header and 72 bytes for the action data. (Recall from Section 3 that you can specify an input data action up to 72 bytes long.) In the case of data input through a screen called by the SCREEN command, the maximum length is determined by the screen format itself. As it happens, all OS/3 languages allow you to interrogate the header and data fields of a menu record separately.

The end-of-file (EOF) indicator

The other type of input information, the EOF indicator, is a feature of many input files, menus included. It is set off while more records remain to be read in a file but set on after the last record in the file is read. The end of a card file is easy to define; it's simply the last card. The end of a menu file is defined a little bit differently, though. In fact, there are three ways a menu can set this indicator on:

1. By the operator pressing the F15 and FUNCTION keys simultaneously. This always turns off menu processing and returns the EOF indicator to the program, regardless of the data that the operator may have entered on the workstation screen.

2. By a BACK menu function command executed when the current menu is on the first level (called directly from the user program). This doesn't happen if the menu is at or beyond the second level.

3. By a RETURN menu function command executed when the command causes the menu processor to end menu processing (there being no menus in the menu chain for which the MARK command was executed)

All languages provide facilities for special handling of EOF indicators. Usually, setting the indicator on is a signal that the operator has finished entering data and wishes to end the program, or at least a particular function of it.

6.4. MENU CONSIDERATIONS

As we stated earlier, an input data type of menu action can contain any alphanumeric data string you wish. However, once you've defined that string with MENUGEN, it remains fixed until you call MENUGEN again to define a different string. If you expect variable data input from the operator, the SCREEN menu function command may be the way to go. However, if you want to restrict operator input to a relatively small set of values, your menu can include each of those values as an input data type of action, returned to the program by the operator selecting its corresponding menu item.

6.5. LINKING MENUS TO USER PROGRAMS – // USE MENU

The // USE MENU statement

To link a menu (or menus) to a program using it, you are responsible for providing the proper job control statements. These statements are basically the same as they would be for any program that's executed at a workstation. You must provide a device assignment set or DVC/LFD sequence for every file that the program uses, including the workstation file, and, if the menu is stored in a library other than the YFMT library, the menu's library file. In addition to the standard job control language (JCL), you must indicate within the DVC/LFD sequence for the workstation that menu services are to be provided. You do this by using the // USE MENU statement.

The format for the USE statement is:

```
// [symbol] USE MENU [,{ $Y$FMT/menu-file-LFD }]
                       { menu-file-LFD/$Y$FMT }
                       { $Y$FMT               }
   [,initial-menu] [,{ nnn }]
                     { 1   }
   [,menu-1=alias-1[,...,menu-12=alias-12]
```

As you can see, most of the parameters in this statement are optional, so that if your menu is stored on YFMT, the following alone is sufficient to indicate the use of menu services:

```
// USE MENU,,initial-menu
```

With this statement included in the DVC/LFD sequence for the workstation file, your program can retrieve the menu named initial-menu from YFMT simply by issuing the proper request. It is this statement that directs program input requests to the workstation file.

The job control stream to execute a program using menus must always include a DVC/LFD sequence, which contains the USE statement, for the workstation. A DVC/LFD sequence also must be provided for the menu library file if it is not YFMT.

Consider the following example:

```
// JOB PAYROL
     .
     .
     .
// DVC 200
// USE MENU,,MENU01   } DVC/LFD sequence for workstation file
// LFD INPUT          } by using menus
     .
     .
     .
// EXEC PAYROLL
```

This example contains a DVC/LFD sequence for a workstation file whose device code number is 200 and LFD name (the name used by the program to reference this file) is INPUT. The menu to be used is on YFMT; thus, no DVC/LFD sequence is given for this file. The menu name MENU01 signifies to program PAYROLL that, when it attempts to read file INPUT, the menu labeled MENU01 (contained in YFMT) is to be displayed on the operator's workstation screen. For more information concerning the USE MENU statement, see Appendix A. For general job control information, refer to the current version of the job control user guide.

6.6. EXAMPLES

In the following discussion, we'll describe a common data processing problem and show how it can be solved by using menu services jointly with programs written in COBOL, FORTRAN, RPG II, and BAL. We suggest that you read through the following general narrative of the problem before turning to the programming language you are interested in.

Problem Description

A program processes a data file through four operations: add a record, delete a record, update a record, or print the entire file. The operator chooses the operation he wants to perform, performs it, and then chooses another operation, repeating the process until he's finished all his work. We want to simplify the process by having the operator enter a different numerical value for each operation he wants. In other words, if the program asks WHAT OPERATION DO YOU WANT? he could make any of the following entries:

1. Add a record

2. Delete a record

3. Update a record

4. Print the entire file

Since the operator has to have some way of ending the program, we add a fifth operation:

5. End the program

This list of operations naturally lends itself to menu form (screen 6–1):

SCREEN 6-1

```
FILE PROCESSING MENU
1. ADD A RECORD
2. DELETE A RECORD
3. UPDATE A RECORD
4. PRINT ENTIRE FILE
5. END THE PROGRAM
```

We'll call this menu OPMENU. We'll assume that we've already created it and put it in the system YFMT file. We'll also assume that if we displayed its action table with MENUGEN, it would look like this (screen 6–2):

SCREEN 6-2

```
                DISPLAY ACTION TABLE
ITEM #  ACTION TYPE  ACTION
 1          D        1
 2          D        2
 3          D        3
 4          D        4
 5          M        BACK
```

Note that the actions for items 1–4 are input numeric data (ACTION TYPE=D for data). If you choose item 3, for example, the menu returns the number 3 to the program that called it. The action for item 5 is not data, but a menu function command, BACK, which does two things: It ends menu processing, and it returns the end-of-file indicator to the program.

Our program, which we'll call PROCFIL, looks like this:

PROCFIL

```
1.  Begin program PROCFIL.
2.  Open all files.
3.  Read a record from the input (menu) file.
4.  If PROCFIL has reached the end of the input file, go to line 11.
5.  Else, test input record as follows:
6.     If input-record = '1', then add a record;
7.        else, if input-record = '2', then delete a record;
8.        else, if input-record = '3', then update a record;
9.        else, print the file (input-record = '4').
10.    Then, when operation is finished, go to line 3.
11. Close all files; there are no more operations to do.
12. End program.
```

This program sets a pattern that all our program examples will follow. It is input driven; that is, it keeps running as long as there's input to read and act upon. After reading the last record in the file and performing the last operation, PROCFIL ends. The input file could be on almost any device available in OS/3. Through job control and CDM, a menu can mimic other file types so effectively that the program won't know the difference.

One more thing: Good programming practice says that PROCFIL should test an input value to ensure that it equals 1, 2, 3, or 4. We haven't included any such test because we don't need to: that chore is taken care of by menu OPMENU, which we've designed so that it will never return anything but those four values (or an end-of-file indicator).

Let's see what this means. If the operator asks for an item that is on the menu, say item 2, it relays his choice to the program, which acts on it. On the other hand, if the operator asks for an item not in the menu, it simply rejects his choice and asks for another entry. We can think of OPMENU as a gatekeeper, passing certain selected data to a program while barring the way to anything else.

With this information in mind, we'll show you the individual programs and the job control each program needs to use menu services.

COBOL Program

The following COBOL program, COBMEN, is one solution to our menu processing problem. We're showing only those parts of COBMEN that relate to menu services. To follow through the program, use the circled keys.

COBOL PROGRAM

```
IDENTIFICATION DIVISION.
PROGRAM-ID. COBMEN.
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
.
.
DATA DIVISION.
FILE SECTION.
FD  INFIL
    BLOCK CONTAINS 80 CHARACTERS (1)
    RECORD CONTAINS 80 CHARACTERS
    LABEL RECORD IS OMITTED.
01  INREC PIC X(80). (2)
.
.
WORKING-STORAGE SECTION.
77  WORK-AREA PIC X(80).
01  FUNCTION-REC REDEFINES WORK-AREA.
    02  FUNCTION-CODE PIC 9.
    02  FILLER PIC X(79).
.
.
PROCEDURE DIVISION.
BEGIN-PROG.
    OPEN INPUT INFIL.  (3)
.
.
MAINLOOP.
    READ INFIL INTO WORK-AREA  AT END GO TO END-THE-PROG.  (4)
    IF FUNCTION-CODE=1 THEN PERFORM ADD-A-RECORD  (5)
        ELSE IF FUNCTION-CODE=2 THEN PERFORM DELETE-A-RECORD
        ELSE IF FUNCTION-CODE=3 THEN PERFORM UPDATE-A-RECORD
        ELSE PERFORM PRINT-THE-FILE.
    GO TO MAINLOOP.  (6)
.
.
END-THE-PROG.
    CLOSE INFIL.  (7)
    STOP RUN.
```

JOB CONTROL

```
// JOB COBMENU
.
.
// DVC 200  (8)
// USE MENU,,OPMENU  (9)
// LFD INFIL  (10)
.
.
// EXEC COBMEN00
/&
```

(1) File INFIL is the input file. Its record and block sizes are both 80 bytes.

(2) The input buffer, like the record and block sizes, is 80 bytes long.

(3) When execution starts, the program opens file INFIL.

(4) The READ command causes the menu processor to display OPMENU on the workstation screen. Processing pauses here until the operator makes his selection and presses XMIT.

- If the selection is item 1–4, CDM relays the input data back to COBMEN and causes it to resume processing (see (5)).
- If the operator chooses item 5 (END MENU), CDM does not return a value of 5 to the program. Instead, it returns an end-of-file indicator, causing the program to branch to END-THE-PROG (see (7)).
- Any other selection causes the menu to reappear on the workstation and wait until the operator makes a valid selection.

(5) At this point, input buffer FUNCTION-CODE contains a value of 1, 2, 3, or 4:

- If FUNCTION-CODE is 1, COBMEN performs a routine (not shown here) that adds a record to the file, just as the workstation operator requested.
- If FUNCTION-CODE is 2, COBMEN performs a routine that deletes a record from the file.
- If FUNCTION-CODE is 3, COBMEN performs a routine that updates a record already in the file.
- If FUNCTION-CODE is none of the above, it must then be 4, so COBMEN prints the entire file.

(6) At the end of whichever operation it's just performed, COBMEN branches back to MAINLOOP to read the next input record. To the workstation operator, the menu screen reappears with an empty prompt, waiting for him to make another choice.

(7) This routine ends the program, the operator having chosen item 5, END MENU. It closes INFIL and terminates the program, causing OPMENU to disappear and returning control of the operator's workstation to the system.

(8) This identifies the device, a workstation, that is to be the program's input device.

(9) This statement calls menu services to manage the workstation screen by using menu OPMENU. Since the menu is on the YFMT system file, no other parameters are needed.

(10) This statement identifies the workstation/menu combination of (8) and (9) as input file INFIL to the program.

FORTRAN Program

The following FORTRAN program, FORMEN, is another solution to our menu processing problem. We're showing only those parts of FORMEN that relate to menu services. To follow through the program, use the circled keys.

FORTRAN PROGRAM

```
      PROGRAM FORMEN
      INTEGER MENOPT  (1)
C  ***************************************
C   READ INPUT FILE TO GET NEXT OPERATION
C  ***************************************
100   READ (1,200,END=1000) MENOPT     (2)
200   FORMAT (I1)
      GO TO (300, 400, 500, 600),MENOPT  (3)
C  ***************************************
C             MENU OPTION 1
C  ***************************************
300   CONTINUE  (4)
      .
      .
      GO TO 100    (5)
C  ***************************************
C             MENU OPTION 2
C  ***************************************
400   CONTINUE  (6)
      .
      .
      GO TO 100
C  ***************************************
C             MENU OPTION 3
C  ***************************************
500   CONTINUE   (7)
      .
      .
      GO TO 100
C  ***************************************
C             MENU OPTION 4
C  ***************************************
600   CONTINUE     (8)
      .
      .
      GO TO 100
C  ***************************************
C          PROCESSING ENDS HERE
C  ***************************************
1000  CONTINUE    (9)
      STOP
      END
```

JOB CONTROL

```
// JOB FORMENU
      .
      .
// DVC 200     (10)
// USE MENU,,OPMENU     (11)
// LFD FORT1        (12)
      .
      .
// EXEC FORMEN00
/&
```

(1) We define MENOPT as an integer variable.

(2) This statement reads a record from device 1 into MENOPT. As defined by the FORMAT statement at line 200, MENOPT is a 1-character variable. The program treats device 1 as an input file; consequently, it branches to statement 1000 (END=1000) when it reaches the end of that file.

What is device 1? As you'll see, it's a workstation running under control of menu OPMENU. Thus, every time the READ statement is executed, two things happen:

- Menu OPMENU appears on the operator's workstation.
- When the operator enters his choice of operation, the menu processor passes its corresponding data value to variable MENOPT. This determines which operation program FORMEN is to perform.

(3) This computed GO TO statement takes advantage of the fact that menu OPMENU returns one of four values: the numbers 1, 2, 3, or 4. Depending on which of these values MENOPT has, the program branches to the corresponding statement. For example, if MENOPT has a value of 3, the program branches to the third address in the list, 500.

(4) If the operator chooses to add a record (item 1 in OPMENU), MENOPT gets a value of 1, and program control branches to this statement. The program then proceeds to add a record to the data file.

(5) The add-a-record operation completed, this statement passes program control back to statement 100 ((2)) so that OPMENU can ask the operator what operation to perform next.

(6) If the operator wants to delete a record (item 2 in OPMENU), MENOPT gets a value of 2, and program control branches here. Starting with statement 400 and continuing to the following GO TO 100 statement, the program deletes a record from the data file.

(7) If the operator wants to update a record (item 3 in OPMENU), MENOPT gets a value of 3, and program control branches here. Starting with statement 500 and continuing to the following GO TO 100 statement, the program updates a record in the data file.

(8) If the operator wants to delete a record (item 4 in OPMENU), MENOPT gets a value of 4, and program control branches here. Starting with statement 600 and continuing to the following GO TO 100 statement, the program deletes a record from the data file.

(9) If the operator chooses item 5 of OPMENU, no data at all is passed to MENOPT. Instead, CDM sets an indicator that tells the program there's no more data from device 1. In that case, the END=1000 specification in the READ statement (see (2)) comes into play, causing the program to branch to statement 1000. This series of statements terminates the program, causing OPMENU to disappear and giving up control of the operator's workstation.

(10) This identifies the device to be used as the program's input device, a workstation.

(11) This statement calls menu services to manage the workstation screen with menu OPMENU. Since the menu is on the YFMT system file, no other parameters are needed.

(12) This statement identifies the workstation/menu combination of 10 and 11 as the input file to the program. We use the FORTRAN default LFD-name of FORT1, since it is device 1 that we are using (see (2)).

RPG II PROGRAM

The following RPG II program, RPGMEN, is another solution to our menu processing problem. We're showing only those parts of RPGMEN that relate to menu services. To follow through the program, use the circled keys.

RPG II PROGRAM **JOB CONTROL**

```
                1         2         3         4         5         6         7
       .........Ø.........Ø.........Ø.........Ø.........Ø.........Ø.........Ø
       Ø1ØH D       1
 (1)   Ø2ØFINFIL    IPEAF       8Ø           DISK
 (2)   Ø3ØFPRNTR    O    F 12Ø 12Ø           PRINTER
 (3)   Ø4ØIINFIL    AA  Ø1    1 D1
       Ø5ØI         AA  Ø2    1 D2
       Ø6ØI         AA  Ø3    1 D3
       Ø7ØI         AA  Ø4    1 D4
 (4)   Ø8ØOPRNTR    H   Ø7    1
       Ø9ØO                             2Ø 'MENU SCREEN TEST    '
 (5)   1ØØO         D 2       Ø1
       11ØO                             2Ø 'OPTION 1 SELECTED   '
       12ØO         D 2       Ø2
       13ØO                             2Ø 'OPTION 2 SELECTED   '
       14ØO         D 2       Ø3
       15ØO                             2Ø 'OPTION 3 SELECTED   '
       16ØO         D 2       Ø4
       17ØO                             2Ø 'OPTION 4 SELECTED   '
 (6)   18ØO         T 3       LR
       19ØO                             2Ø 'END OF PROGRAM      '
```

```
// JOB RPGMENU
     .
     .
// DVC 2ØØ        (7)
// USE MENU,,OPMENU     (8)
// LFD INFIL         (9)
// DVC 2Ø  // LFD PRNTR
     .
// EXEC RPGMENØØ
/&
```

(1) Line 020 specifies an input file named INFIL, which is on disk and has a record size of 80 bytes. At run time, this becomes the menu file on a workstation. Because job control and CDM can hide the differences between the disk file specified in the program and the workstation the program will actually use, there is no need to change either the file's device type or record size.

(2) Line 030 specifies a printer output file named PRNTR.

(3) Lines 040 to 070 are record identification entries all keyed to the first character of a record read from INFIL. Every time the program reads a record, these entries turn on different indicators, depending on what the record contains. Remember that menu OPMENU returns a record that has the value 1, 2, 3, or 4 in its first byte. Therefore, when the program reads a record from INFIL, it tests the first byte of the record for the presence of one of these four values. For example, if the first byte of the record contains a 2, line 040 turns indicator 02 on, all other indicators remaining turned off.

In addition to all the preceding indicators, the program turns the LR (last record) indicator on when, instead of returning a record, INFIL reaches the end of the file.

(4) Lines 080 and 090 print the header line "MENU SCREEN TEST" at the top of the first page output by the program.

(5) Lines 100 to 170 determine which of four detail lines is printed as the detail output operation of the program. For example, if indicator 02 had earlier been turned on, the line "OPTION 2 SELECTED" is printed before the program begins another detail cycle.

(6) When an end-of-file condition on INFIL turns the LR indicator on (see (3)), the program performs its total output operation by printing the line "END OF PROGRAM". Then the program ends.

(7) This identifies the device, a workstation, that is to be the program's input device.

(8) This statement calls menu services to manage the workstation screen by using menu OPMENU. Since the menu is on the YFMT system file, no other parameters are needed.

(9) This statement identifies the workstation/menu combination of (7) and (8) as input file INFIL.

BAL Program

The following BAL program, BALMEN, is another solution to our menu processing problem. We're showing only those parts of BALMEN that relate to menu services. To follow through the program, use the circled keys.

BAL PROGRAM

```
BALMEN    START Ø
          .
          .
          OPEN  INFIL,(INRIB)  (1)
MAINLOOP  DMINP INFIL,INBUFF   (2)
          BP    ENDFILE        (3)
          CLI   INBUFF,C'1'    (4)
          BE    ADDREC
          CLI   INBUFF,C'2'    (5)
          BE    DELREC
          CLI   INBUFF,C'3'    (6)
          BE    UPDREC
          B     PRINTFIL       (7)
          .
          .
ADDREC    EQU   *              (8)
          .
          .
[ADDREC instructions omitted]
          .
          .
          B     MAINLOOP
          .
          .
[DELREC, UPDREC, and PRINTFIL follow the same pattern.]
          .
          .
ENDFILE   CLOSE INFIL          (9)
          EOJ
          .
          .
INFIL     CDIB                 (10)
INRIB     RIB   RCSZ=80,BFSZ=80,IOA1=INBUFF
INBUFF    DS    20F
          END
```

JOB CONTROL

```
// JOB BALMENU
.
.
// DVC 200          (11)
// USE MENU,,OPMENU (12)
// LFD INFIL        (13)
.
.
// EXEC BALMEN00
/&
```

(1) This statement opens input file INFIL, which menu services will use to send data to the program.

(2) This statement is the first in a sequence of statements that form the main body of the program. Each time it's executed, menu services display OPMENU on the operator's workstation and wait for a reply. Then, when the operator makes a choice, menu services send the appropriate data back to the program, placing it in buffer INBUFF.

(3) If the operator chooses item 5 of OPMENU (END MENU), menu services set the condition code (CC) to 4, indicating end-of-file. The BP statement tests the CC: if it equals 4, it passes control to label ENDFILE (see (9)). Otherwise, control passes to the next statement.

(4) Starting here, the program tests the value placed in INBUFF by menu services to determine what the operator wants to do to the data file. If the value is 1, the program goes to the routine ADDREC that adds a record to the data file; otherwise, it continues on to the next instruction.

(5) If the INBUFF value is 2, BALMEN goes to the routine DELREC that deletes a record already in the file.

(6) If the INBUFF value is 3, BALMEN goes to routine UPDREC, which updates a record already in the file.

(7) If program control comes this far, then INBUFF, by default, must contain a value of 4; BALMEN therefore goes to routine PRINTFIL, which prints the entire file.

(8) This is an outline of ADDREC, called when the operator chooses item 1 (see (4)). As you can see, the routine does its work and then returns program control to the statement labeled MAINLOOP ((2)) so the operator can choose another BALMEN operation. Routines DELREC, UPDREC, and PRINTFIL work likewise.

(9) This instruction closes INFIL, causing OPMENU to disappear from the workstation screen.

(10) This sequence of statements defines file INFIL and the resource information block (RIB) that is associated with it when BALMEN opens the file at (1) . The file record is fixed at 80 bytes in length and uses INBUFF as its buffer.

(11) This identifies the device to be used as the program's input device, a workstation.

(12) This statement calls menu services to manage the workstation screen with menu OPMENU. Since the menu is on the YFMT system file, no other parameters are needed.

(13) This statement identifies the workstation/menu combination of (11) and (12) as input file INFIL to the program.

Appendix A. Job Control Considerations

The // USE MENU job control statement

When using menus with user programs, you must indicate within the DVC/LFD sequence for the operator's workstation that menus are to be used. That is the purpose of the USE MENU job control statement. The format for the statement is:

Format

```
// [symbol] USE MENU[,{$Y$FMT/menu-file-LFD }]
                     {menu-file-LFD/$Y$FMT }
                     {$Y$FMT               }
  [,initial-menu].,{nnn}
                   .{1  }
  [,menu-1=alias-1[,...,menu-12=alias-12]
```

where:

MENU
Indicates that the program is to use a menu.

```
{$Y$FMT/menu-file-LFD }
{menu-file-LFD/$Y$FMT }
{$Y$FMT               }
```

Names up to two files to be searched for menus, one file of which must be YFMT. Any name you use must match an LFD name specified in a previously defined device assignment set for a menu library file (always a MIRAM file). A menu-file-LFD is one to eight alphanumeric characters long. If you don't specify anything for this parameter, it is assumed that all menus reside in system format file YFMT. When coding this parameter, remember the following:

- If you omit YFMT from *YFMT/menu-file-LFD,* then code */menu-file-LFD.* This causes the menu processor to search for menus and their associated screens first in YFMT then, if not found there, in the menu-file-LFD.

- If you omit YFMT from *menu-file-LFD/YFMT,* then code *menu-file-LFD/.* This causes the menu processor to search for menus and their associated screens first in the menu-file-LFD library and then, if not found there, in YFMT.

NOTES:

1. *If you specify a menu-file-LFD, you must also include YFMT (by specification or default). This is because YFMT contains error screen formats that must be accessible to the menu processor if it encounters errors in your menu.*

2. *If you specify a menu-file-LFD, all screen formats used by the menu and generated by the screen format generator must be stored in either YFMT or in the menu-file-LFD library. If you intend to use only YFMT, all needed screen formats must be stored in YFMT.*

`initial-menu`

Specifies the name of the menu that is to be displayed when the program requests input from the workstation; must be one to eight alphanumeric characters long.

This parameter is required if your program is written in COBOL, FORTRAN, or RPG II; these languages don't give you any way of specifying the name of a menu within a program. If your program is in BAL, you can use either this parameter or the DMSEL (MENU) macroinstruction to specify a menu name, so this parameter is optional. (For more information on the DMSEL macroinstruction, see the current version of the data management macroinstructions user guide.)

`nnn`

Specifies the number of menus to be resident in main storage at one time, in the range 1 to 255. The default value is 1. The more menus you make resident, the faster menu processing takes place, although they take up more main storage.

`menu-n=alias-n`

Allows you to equate a menu name specified in an application program (alias) to a menu with a different menu name (given when the menu was created). A maximum of 12 alias-name sets may be specified. The menu and alias names each must be from one to eight alphanumeric characters in length.

Examples

The following is an example of the USE statement with some accompanying parameters:

```
// USE MENU,MENUFIL/$Y$FMT,PAYMENU,2
```

This statement indicates:

- The menu processor first searches for the menu in a library file other than YFMT. The LFD-name for that file is MENUFIL.
- The menu named PAYMENU is the one called by menu services when the program wants input.
- Up to two menus can reside in main storage at the same time.

The following example of the USE statement, written for use with a BAL program, has a different set of parameters from the first example:

```
// USE MENU,,,,PROGMENU=ALIASMEN
```

This statement indicates:

- The menu processor searches only YFMT for the menu.
- A DMSEL statement in the BAL program specifies a menu named PROGMENU. Instead of PROGMENU, however, the menu processor displays another menu named ALIASMEN.

The control stream for a job that uses menu services could include these job control statements:

```
// JOB YOURJOB
          .
          .
          .
// DVC 50
// VOL ABC
// LBL MENUFILE
// LFD MENUFIL
          .
          .
          .
// DVC 200
// USE MENU,MENUFIL/$Y$FMT,PAYMENU
// LFD WORKSTN
          .
          .
          .
// EXEC PRGRM1
/&
```

Device assignment set for the menu library file

Menu library file LFD name

Device assignment set for the workstation

When you run YOURJOB, PRGRM1 is executed. PRGRM1 contains an instruction to open WORKSTN, which opens the menu library file MENUFIL. Another instruction later in the program requests input data from the WORKSTN file; in response, menu services are called upon to display menu PAYMENU, the operator enters data in PAYMENU, and that data is passed back to PRGRM1.

Appendix B. Creating Menu and Help Screens by Using Screen Format Generator

As we've stated throughout this manual, you can create menu screens and help screens for your menus by using the screen format generator (SFG). The only role your menu plays is to call these screens whenever it itself is called. This allows you two kinds of flexibility:

1. Your screens can be more flexible, not having to conform to the fixed formats contained in MENUGEN templates.

2. You can create these screens independently of your menu. You can, for example, create a screen once with SFG but use it in as many menus as you wish. Or you can divide the work involved among several programmers, saving time and promoting efficiency.

We'll discuss menu screen and help screen guidelines separately.

B.1. MENU SCREEN GUIDELINES

A menu screen must include the menu prompt. When you create the screen by using SFG, you must provide for the prompt by specifying exactly one field with the following characteristics:

1. Its length must conform to the following rules:

   a. If the number of items in your menu doesn't exceed 9, the field must be two characters long to accommodate both the question mark and any allowable item number in the menu.

   b. If the number of items in your menu exceeds 9, some of your item numbers will be two digits by themselves. Therefore, to handle these and the question mark, your input field must be three characters long.

2. It must be an alphanumeric field (X) to handle item numbers and the ? character.

3. It must be an input-only (I) field.

You cannot have more than 99 items in a menu. Apart from that restriction, you can place your prompt anywhere in the menu, arrange your items any way you wish, or even include nonitem text to help the operator along.

B.2. HELP SCREEN GUIDELINES

Help screens are intended to be display-only screens. You can map out the text of your help screen any way you wish. Like menu screens, they have a single input-only field. Its purpose here, though, is not to input data. Instead, the presence of an input field causes the screen format coordinator, which manages these screens, to anticipate some sort of input. So, it displays a help screen and keeps it on the workstation screen until the operator, having finished reading the screen, *inputs* data simply by pressing XMIT. The input field is a dummy field that isn't used for passing data at all.

As stated, you must specify one input (I) field and no output or bidirectional fields in the help screen. The field must accept alphanumeric data (X) and must be exactly one byte long. Usually, it appears in your screen right after the words PRESS TRANSMIT TO CONTINUE.

Appendix C. MENUGEN Error Screens

```
                                                    MG9901E
            OPEN ERROR ON USER OUTPUT FILE
 LBL: filename
 VSN: volserno
 THE ABOVE FILE MUST BE A MIRAM FILE RESIDING ON THE SPECIFIED
 VSN.  TO TRY AGAIN, RESPOND 'Y' BELOW.  PROMPT MG0101S (MENU
 MODULE OUTPUT INFORMATION) WILL BE DISPLAYED.  YOU CAN THEN
 TRY TO OPEN THE SAME OR DIFFERENT FILE. AN 'N' RESPONSE WILL
 END THE CREATE OPERATION.
 RETRY OPENING OUTPUT FILE? (Y/N) _
```

This screen appears if, during the create operation, MENUGEN fails to open the output file you specified by using screen 3–3.

```
                                                    MG9902E
 AN ERROR WAS DETECTED WHEN WRITING THE MENU MODULE TO
 YOUR FILE. IF YOU WANT TO TRY WRITING IT AGAIN, RESPOND
 'Y' BELOW. AN 'N' RESPONSE WILL END THE CREATE OPERATION
 WITHOUT WRITING THE MENU MODULE.
     RETRY WRITING MM TO YOUR FILE? (Y/N)_
```

This message may appear if an I/O error prevents MENUGEN from writing your newly created menu to the destination file you specified by screen 3–3. You may try again to write the menu to the same file (reply Y). If this fails, you should reply N; this returns you to the MENUGEN home screen.

```
                                                    MG9903E
 AN ERROR WAS DETECTED WHEN READING THE MENU MODULE
 FROM THE MENU GENERATOR SCRATCH FILE. THE MENU MODULE
 CANNOT BE SAVED. IT MUST BE CREATED AGAIN. THE MENU
 GENERATOR HOME SCREEN (MG01) WILL BE DISPLAYED.
                          PRESS TRANSMIT TO CONTINUE_
```

```
                                                          MG9904E
ERROR DETECTED WHEN WRITING THE ACTION RECORD TO THE MENU
GENERATOR SCRATCH FILE. IF DESIRED, AN ATTEMPT WILL BE MADE
TO WRITE THE INCOMPLETE MENU MODULE TO YOUR FILE. THE CREATE
OPERATION WILL END AND THE HOME SCREEN MG01 WILL BE DISPLAYED.
DO YOU WANT TO WRITE THE MENU MODULE TO YOUR FILE (Y), OR (N)
```

This error may occur during the create operation. To retry the write operation, reply Y. To skip the retry, reply N. Whether or not you choose to retry the operation, the MENUGEN home screen reappears.

```
                                                          MG9905E
              DUPLICATE MODULE FOUND
MENU MODULE: menuname
LBL: filename
VSN: vsn
THE ABOVE MENU MODULE ALREADY EXISTS IN THE SPECIFIED FILE.
RESPOND 'N' BELOW TO CHANGE YOUR OUTPUT SPECIFICATIONS (WRITE
WILL BE TRIED AGAIN). REPLY 'C' BELOW AND THE MENU MODULE WON'T
BE WRITTEN.
DO YOU WANT TO OVERWRITE THE EXISTING MENU MODULE? (Y/N/C)_
```

```
                                                          MG9906E
                OPEN ERROR ON USER INPUT FILE
AN ERROR WAS DETECTED WHEN TRYING TO OPEN THE FILE
WHERE YOUR MENU SCREEN IS LOCATED. IF YOU WOULD
LIKE TO TRY AGAIN, ENTER 'Y' BELOW. YOU WILL BE
ASKED TO SUPPLY A VSN AND LBL AGAIN. IF YOU REPLY
'N' YOUR SELECTION OF 'DISPLAY MENU SCREEN' WILL BE
IGNORED, AND YOU WILL BE ALLOWED TO CHOOSE ANOTHER
ITEM.
DO YOU WANT TO RETRY DISPLAYING YOUR MENU SCREEN? (Y/N)_
```

This message appears if, while you are using the modify or display operation, MENUGEN can't open your secondary file. This can happen when MENUGEN is trying to retrieve a menu screen or a help screen.

```
                                                          MG9907E
                    ERROR DETECTED ON MENU SCREEN
MENU SCREEN NAME: menuname
FILE LBL: filename
FILE VSN: volserno
AN ERROR WAS DETECTED TRYING TO DISPLAY YOUR MENU SCREEN
LISTED ABOVE. EITHER AN IO ERROR OCCURRED, OR IT WAS NOT
FOUND. IF YOU WANT TO RETRY, RESPOND 'Y' BELOW. IF YOU
RESPOND 'N', THE MENU SCREEN WILL NOT BE DISPLAYED.
DO YOU WANT TO RETRY DISPLAYING YOUR MENU SCREEN? (Y/N)_
```

This message appears if, while you are using the modify or display operation, MENUGEN can't find a screen in your secondary file or can't display one it found. This can happen when MENUGEN is trying to retrieve a menu screen or a help screen.

```
                                                          MG9908E
              INVALID ITEM SELECTION
THIS ITEM SELECTION IS NOT ALLOWED. ONLY 4 MENU GENERATOR
CREATED HELP SCREENS ARE ALLOWED PER ITEM NUMBER. IF MORE HELP
SCREENS ARE NEEDED FOR THIS ITEM, THEN SCREENS CREATED BY THE
SCREEN FORMAT GENERATOR MAY BE USED.
                                  PRESS TRANSMIT TO CONTINUE _
```

If you are adding help screens to a menu item that you are creating with MENUGEN (screen 3–26, item 1), you are limited to four help screens per item. This restriction doesn't apply to help screens created by SFG.

```
                                                          MG9909E
        INVALID ITEM SELECTION
''NO HELP FOR THIS ITEM'' IS AN INVALID SELECTION. YOU HAVE
CREATED AT LEAST ONE HELP SCREEN FOR THIS ITEM ALREADY.
YOU MUST CHOOSE EITHER ''HELP FOR NEXT ITEM'' OR ''HELPS ARE
COMPLETE, END CREATE OPERATION.''
                                  PRESS TRANSMIT TO CONTINUE _
```

```
                                                          MG9910E
      INVALID ACTION DETECTED IN MENU MODULE menuname
THE ABOVE MENU GENERATOR SUPPLIED MENU CONTAINS AN INVALID
ACTION IN ITS ACTION TABLE. THE CURRENT MENU GENERATOR
OPERATION WILL END. RESTORE THE MENU MODULE FROM A VALID COPY.
                                  PRESS TRANSMIT TO CONTINUE -
```

```
                                                          MG9920E
ERROR DETECTED WHEN READING MENU MODULE menuname
YOU CAN TRY AGAIN OR TERMINATE.
MENU MODULE NAME: menuname
FILE LBL: filename
VSN: vsn
DO YOU WANT TO TERMINATE?_ (Y)ES,(N)O  ( DEFAULT IS N )
```

You may get this message during the modify operation. You can retry the read operation by replying N or simply pressing XMIT. If the retry fails, end the modify operation by replying N; this returns you to the home screen.

```
                                                          MG9921E
                    WRITE ERROR
A WRITE ERROR WAS DETECTED WHILE MOVING THE MENU MODULE
TO SCRATCH. THE MODIFY OPERATION WILL TERMINATE.
                        PRESS TRANSMIT TO CONTINUE _
```

This message indicates that you must either choose another MENUGEN option or terminate MENUGEN. Sometimes terminating and then calling MENUGEN clears up the problem.

```
                                                          MG9922E
MODULE XXXXXXXX IS NOT A MENU MODULE.
ENTER A DIFFERENT NAME OR TERMINATE
MENU MODULE NAME: ________
FILE LBL: ____________________________________________
VSN: ______
TERMINATE?_  (Y)ES,(N)O
```

You get this message if, during the modify operation, you ask MENUGEN to get a module that turns out to be incompatible with it. You can either ask MENUGEN to get a different module, by changing the menu, file, or volume name and pressing XMIT, or you can terminate the modify operation by replying Y to the TERMINATE? prompt.

```
                                                        MG9923E
        CHANGE MENU SCREEN TEXT
THE TEXT OF MENU SCREENS CREATED BY THE SCREEN FORMAT
GENERATOR CANNOT BE CHANGED USING THE MENU GENERATOR
         PRESS TRANSMIT TO CONTINUE _
```

```
                                                        MG9924E
                    modify-operation
                  ERROR operation RECORD


            PRESS TRANSMIT TO CONTINUE _
```

An I/O error occurred during the specified modify operation. The MENUGEN home screen reappears. Try the operation again. If that doesn't work, try terminating and restarting MENUGEN.

```
                                                        MG9925E
                    CHANGE EXISTING ACTIONS
THERE ARE NO ACTIONS ASSOCIATED WITH ITEM 01.
      PRESS TRANSMIT TO CONTINUE. _
```

```
                                                        MG9926E
                    DELETE ACTIONS
  THERE ARE NO ACTIONS CONFIGURED FOR THIS ITEM.
YOU MUST SELECT ITEM 2 ON MENU MG22 TO ADD ACTIONS

                PRESS TRANSMIT TO CONTINUE _
```

```
                                                        MG9927E
                    FILE OPEN ERROR
LBL: filename
VSN: vsn
THE ABOVE FILE MUST BE A MIRAM FILE RESIDING ON THE
SPECIFIED VSN.
        RETRY OPENING FILE? _ (Y)ES,(N)O
```

On the modify operation, MENUGEN failed to open the specified file. You can retry the operation by replying Y; this causes MENUGEN to display screen 4–2, which you use to specify the file you want opened, whether the same file as before or a different file. If you want to skip the operation, reply N; this returns you to the home screen.

```
                                                        MG9928E
     MODIFY ACTIONS
ITEM NUMBER 0 IS NOT VALID.
ITEM NUMBERS START AT 1.
PRESS TRANSMIT TO CONTINUE _
```

```
                                                        MG9929E
               ADD ACTIONS
              THIS ITEM IS ALREADY CONFIGURED.
TO ADD ACTIONS FOR THIS ITEM SELECT OPTION 1 ON MENU MG22

                    PRESS TRANSMIT TO CONTINUE _
```

```
                                                        MG9930E
      OPEN ERROR DETECTED ON USER INPUT FILE
FILE LBL: filename
VSN: volserno
THE ABOVE FILE MUST BE A MIRAM FILE RESIDING ON THE SPECIFIED
VSN. TO TRY AGAIN, RESPOND 'Y' BELOW. PROMPT MG0103S (WHERE IS
MENU MODULE) WILL BE DISPLAYED. YOU CAN THEN TRY TO OPEN THE
SAME OR DIFFERENT FILE. AN 'N' RESPONSE WILL END THE DISPLAY
OPERATION.
     RETRY OPENING THE OUTPUT FILE? (Y/N) _
```

This message may appear during the MENUGEN display or modify operation. It indicates that the file you say contains your menu module can't be opened for some reason. MENUGEN repeats screen 4–2 (if you're modifying the menu) or screen 5–2 (if you're displaying it) to let you specify the input file once again.

```
                                                        MG9930S


THE FOLLOWING HELP SCREEN screen-name IS FOR ITEM 01


          PRESS TRANSMIT TO CONTINUE _
```

When you ask to display help screens, this screen precedes each help screen that has been created with the screen format generator.

```
                                                          MG9931E

            MENU MODULE NOT FOUND
MENU MODULE NAME: menuname
FILE LBL: filename
VSN: ------

THE ABOVE MENU MODULE WAS NOT FOUND. A 'Y' RESPONSE BELOW WILL
ALLOW YOU TO CHANGE THE NAME, LBL OR VSN AND TRY AGAIN. A 'N'
RESPONSE WILL END THE DISPLAY OPERATION.
         RETRY? (Y/N) _
```

This message may appear during the MENUGEN display operation. It indicates that MENUGEN can't find the menu you want on the file you specified. If you reply with Y, MENUGEN repeats screen 5–2 to let you specify the menu or input file once again.

```
                                                          MG9931S


THE FOLLOWING MENU GENERATOR CREATED HELP SCREEN IS FOR ITEM 01

                        PRESS TRANSMIT TO CONTINUE _
```

When you ask to display help screens, this screen precedes each help screen that has been created with MENUGEN.

```
                                                          MG9932E

ERROR DETECTED WHEN READING MENU MODULE menuname

AN ERROR WAS DETECTED WHILE READING YOUR MENU MODULE. THE
MODULE MAY HAVE TO BE RE-CREATED. THE DISPLAY OPERATION WILL
BE TERMINATED.
                        PRESS TRANSMIT TO CONTINUE _
```

This message may appear during the MENUGEN display operation. It indicates that MENUGEN encountered a problem while reading the menu module given by menuname. The problem may be serious enough that you have to re-create your file. In any case, the display operation ends and MENUGEN displays the home screen.

```
                                                          MG9933E
                    SECONDARY FILE FAILED TO OPEN
AN ERROR OCCURRED WHILE TRYING TO OPEN THE FOLLOWING FILE:
FILE LBL: filename
VSN: volserno
BY RESPONDING 'Y' YOU CAN RETRY USING THE SAME OR DIFFERENT LBL
AND VSN. AN 'N' RESPONSE WILL END THE DISPLAY OPERATION.
      RETRY OPENING SECONDARY FILE? (Y/N) _
```

This message may appear during the MENUGEN display or modify operation. It indicates that some problem prevented MENUGEN from opening the specified secondary file. If you respond with Y, MENUGEN redisplays an appropriate screen for you to specify a secondary file once again. If you respond with N, MENUGEN returns to the home screen.

```
                                                          MG9934E
  ERROR DETECTED WHILE ATTEMPTING TO DISPLAY SCREEN ________
WHILE TRYING TO DISPLAY THE ABOVE SCREEN, EITHER AN IO ERROR
OCCURRED, OR THE SCREEN WAS NOT FOUND. THE DISPLAY OF THE ABOVE
SCREEN WILL BE IGNORED.
                                  PRESS TRANSMIT TO CONTINUE _
```

This message may appear during the MENUGEN display or modify operation. It indicates that some problem prevented MENUGEN from displaying the specified screen found in your secondary file. MENUGEN returns to the home screen.

```
                                                          MG9935E
                ERROR DETECTED WHEN OPENING PRINTER
AN ERROR WAS DETECTED WHEN OPENING THE PRINTER. THE DISPLAY
OPERATION WILL BE TERMINATED.
                                  PRESS TRANSMIT TO CONTINUE _
```

```
                                                          MG9936E

                    INVALID MENU MODULE
THE ABOVE MENU MODULE IS INVALID. IT WILL HAVE TO BE
RECREATED. THE DISPLAY OPERATION WILL BE TERMINATED.
              PRESS TRANSMIT TO CONTINUE _
```

This message may appear during the MENUGEN display or modify operation. It indicates that the menu you specified to display or modify has been corrupted to such a degree that it is unusable. You have to re-create it by using the create operation. MENUGEN returns to the home screen when you press XMIT.

```
                                                          MG9937E

          HELP FOR ITEM NUMBER nn WAS NOT FOUND
THE SELECTED ITEM NUMBER DOES NOT HAVE A HELP ASSOCIATED WITH
IT. EITHER THE ITEM NUMBER IS INVALID, OR THE CREATE OPERATION
WAS ENDED PRIOR TO CREATING A HELP FOR THIS ITEM. THIS DISPLAY
WILL BE IGNORED.
                        PRESS TRANSMIT TO CONTINUE _
```

This message should not be confused with the NO HELP PROVIDED FOR THIS ITEM message that you can specify while creating help screens with MENUGEN. The only items for which you see this message are items that don't exist or items for which no help screen, not even the NO HELP PROVIDED screen, was specified at all.

```
                                                          MG9960E

              CHANGE HELP SCREEN

THERE IS NO HELP SCREEN ASSOCIATED WITH ITEM 01
                    PRESS TRANSMIT TO CONTINUE _
```

You see this screen if you try to modify a help screen for an item for which there are no help screens. MENUGEN redisplays screen 4–18.

```
                                                          MG9961E
        END THE MODIFY OPERATION
A READ ERROR OCCURRED WHEN MOVING THE MENU MODULE
FROM THE WORK FILE. THE MENU GENERATOR WILL END.

                  PRESS TRANSMIT TO CONTINUE _
```

A read error involving the scratch file has forced MENUGEN to terminate. Try rerunning MENUGEN.

```
                                                          MG9962E
             END MODIFY OPERATION
A WRITE ERROR OCCURRED MOVING THE MENU MODULE
TO YOUR FILE. YOU CAN TRY AGAIN OR TERMINATE.
TO TRY AGAIN ENTER BELOW
MENU MODULE NAME:
FILE LBL: filename
VSN: vsn
TO TERMINATE ENTER 'Y' HERE _
```

A write error prevented MENUGEN from moving the scratch file to your designated output file. If you want to try again, reply N or just press XMIT. This action displays screen 4–5, which you use to specify the same menu and file or a different menu or file. If you want to skip the operation, reply Y; this returns you to the home screen.

```
                                                          MG9963E
        CHANGE HELP SCREEN
THE TEXT OF HELP SCREENS CREATED BY THE
SCREEN FORMAT GENERATOR CANNOT BE CHANGED
USING THE MENU GENERATOR.
           PRESS TRANSMIT TO CONTINUE _
```

You have asked MENUGEN to change the text of a help screen created by SFG, which is something only SFG can do. Press XMIT to return to screen 4–18 and use SFG to make the screen changes you want.

```
                                                          MG9964E
                ADD ACTIONS
THERE IS NO ROOM IN INTERNAL TABLE FOR
ITEM NUMBER CORRESPONDING TO THE ACTIONS YOU
HAVE JUST ADDED. THE MENU GENERATOR WILL END.
     PLEASE CONTACT YOUR SYSTEMS ANALYST.
  PRESS TRANSMIT TO END THE MENU GENERATOR _
```

This message indicates a serious problem in MENUGEN. Contact your Sperry Univac representative.

```
                                                          MG9965E
           ADD/DELETE HELP SCREENS
ITEM nn HAS A BAD HELP SCREEN RECORD. DELETE ALL
HELP SCREENS FOR THIS ITEM AND RECREATE THEM.

                  PRESS TRANSMIT TO CONTINUE _
```

```
                                                          MG9966E
          MENU MODULE menuname DOES NOT EXIST
        REENTER YOUR INPUT INFORMATION OR TERMINATE
MENU MODULE NAME: menuname
FILE LBL: (DEFAULT IS $Y$FMT)
       filename
VSN: vsn (DEFAULT IS RES)
TO TERMINATE ENTER 'Y' HERE _ (DEFAULT IS N)
```

On the modify operation, MENUGEN could not find the specified module on the specified file. You have to terminate the modify operation, so reply Y and return to the home screen.

```
                                                          MG9967E
   DELETE HELP SCREENS
HELP SCREENS DO NOT EXIST FOR ITEM nn
    PRESS TRANSMIT TO CONTINUE _
```

```
                                                        MG9968E
    DISPLAY HELP SCREENS
NO HELP SCREENS EXIST FOR ITEM nn

   PRESS TRANSMIT TO CONTINUE _
```

```
                                                        MG9969E
        DISPLAY HELP SCREENS
THE TEXT OF HELP SCREENS CREATED BY THE SCREEN
FORMAT GENERATOR CANNOT BE DISPLAYED USING THE
MENU GENERATOR.

             PRESS TRANSMIT TO CONTINUE _
```

While in the modify operation, you have asked MENUGEN to display the text of a help screen created by SFG, something it cannot do. Upon pressing XMIT, you return to the modify operation.

```
                                                        MG9970E
    REPLACE MENU SCREEN
     NO SCREEN NAME WAS ENTERED.
YOU WERE SUPPOSED TO ENTER THE NAME OF
A SCREEN CREATED BY THE SCREEN FORMAT
GENERATOR.

     PRESS TRANSMIT TO CONTINUE _
```

```
                                                        MG9990E

MENU MODULE DOES NOT CONTAIN ANY ACTIONS

     PRESS TRANSMIT TO CONTINUE _
```

MENUGEN displays this message during the modify operation if it is asked to display a menu's actions and it finds the menu has none. MENUGEN returns to the previous modify operation menu.

```
                                                          MG9991E

MENU MODULE DOES NOT CONTAIN ANY HELP SCREENS

      PRESS TRANSMIT TO CONTINUE _
```

MENUGEN displays this message during the modify operation if it is asked to display a menu's help screens and it finds the menu has none. MENUGEN returns to the previous modify operation menu.

Index

SPERRY UNIVAC

USER COMMENT SHEET

Your comments concerning this document will be welcomed by Sperry Univac for use in improving subsequent editions.

Please note: This form is not intended to be used as an order blank.

_____

(Document Title)

_____ _____ _____

(Document No.) *(Revision No.)* *(Update No.)*

Comments:

Cut along line.

From:

_____

(Name of User)

_____

(Business Address)

Fold on dotted lines, and mail. (No postage stamp is necessary if mailed in the U.S.A.)
Thank you for your cooperation

FOLD

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 21 BLUE BELL, PA.

POSTAGE WILL BE PAID BY ADDRESSEE

SPERRY UNIVAC

ATTN.: SYSTEMS PUBLICATIONS

P.O. BOX 500
BLUE BELL, PENNSYLVANIA 19424

CUT

FOLD