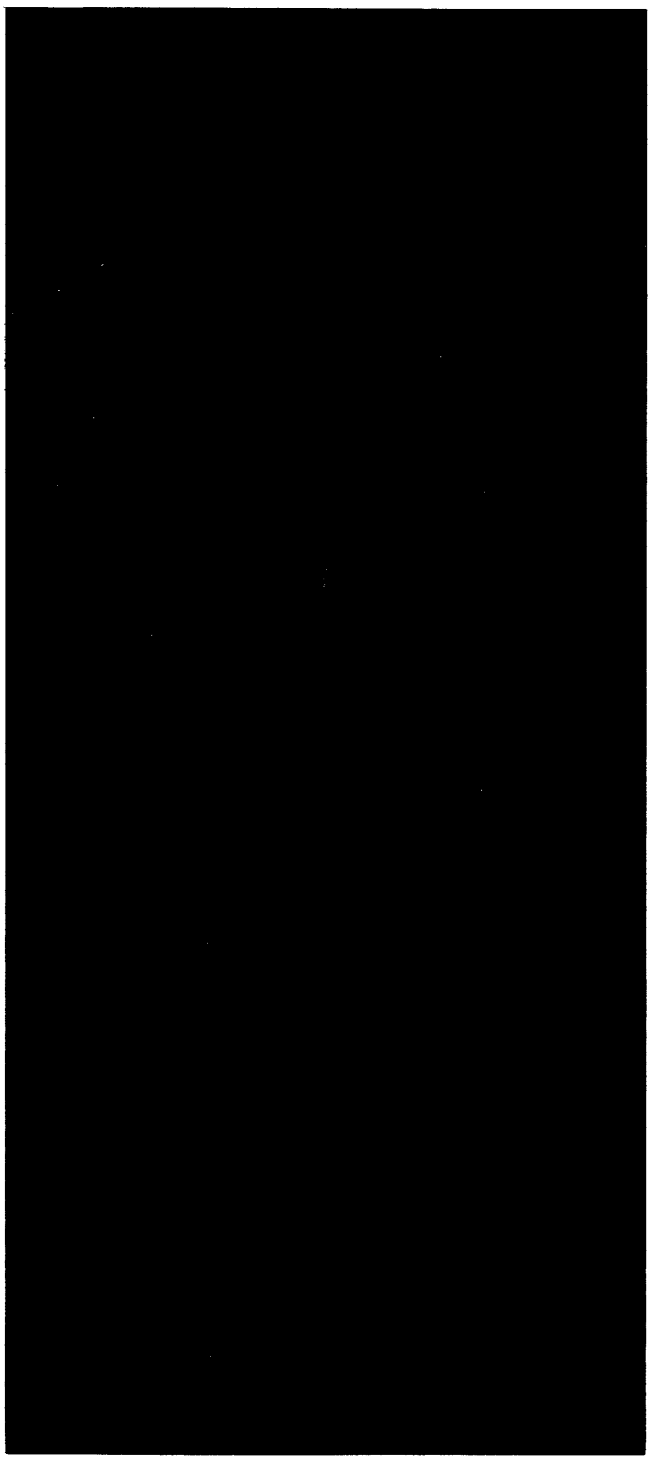
Honeywell

OPERATOR'S GUIDE

SERIES 60 (LEVEL 6) GCOS/BES2

SOFTWARE



Honeywell

OPERATOR'S GUIDE

SERIES 60 (LEVEL 6)

GCOS/BES2

SUBJECT:

Operator's Guide for Series 60 (Level 6) GCOS/Basic Executive System 2

SOFTWARE SUPPORTED:

This manual supports Release 0200 of the Series 60 (Level 6) GCOS/Basic Executive System (BES2) software. When a later release of the system occurs, see the Subject Directory of the latest Series 60 (Level 6) GCOS/BES Software Overview and System Conventions manual (Order No. AU50) to ascertain whether this revision of this manual supports that release.

DATE:

July 1976

ORDER NUMBER:

AU46, Rev. 0

PREFACE

This manual specifies the operating procedures for the GCOS/Basic Executive System 2 (GCOS/BES2), including the procedures for loading and starting up online and offline application programs and system programs that execute on the Series 60 (Level 6) 6/30 Models. Unless otherwise stated herein, the term BES refers to the GCOS/BES2 software; the term Level 6 indicates the specific models of Series 60 (Level 6) on which the software executes.

The manual is divided into parts, each of which contains sections consecutively numbered within the part. Figure, table and page numbers in this document are three-part numbers identifying the part, the section, and the specific figure, table or page within the section (for example, Figure III-5-4 is the fourth figure in Section 5 of Part III).

Part I summarizes the loading procedures contained in this manual and the suggested method of using the manual. Part II describes control panel functionality/procedures and identifies the peripherals used in performing operating procedures. Part III details the loading procedures for system programs; it includes separate sections on operating procedures for each system program and complete lists of messages issued by each system program. Part IV contains the operating procedures for user-written online application programs and all messages issued by the Configuration Load Manager and the online application. Part V presents the operating procedures for user-written offline application programs.

Appendices A, B and C provide functional descriptions of the Disk, Paper Tape and Card Loaders, respectively, and Appendix D discusses the use of the Offline Trap Handler with the loaders. Appendix E details the procedure for using the MDUMP bootstrap record to perform a memory dump to a diskette. Appendix F describes the use of the basic control panel in bootstrapping and loading programs. Appendix G contains an explanation of the codes used in coded error messages in this manual.

File No.: 1S43

AU46

The convention used throughout this manual to identify the keys on the control panel is to specify the functional name of the key, capitalizing and underscoring any letters in the name which actually appear on the key on the control panel (for example, Execute refers to the key labeled E which performs the function of execution). Refer to the section on the control panel in Part II for functional names of control panel keys.

In this manual, the following conventions and symbols are used:

Δ	Blank space
UPPERCASE CHARACTERS	Reserved words or symbols; enter or use them exactly as shown
lowercase characters	Symbolic name or value; you must supply the exact value
brackets [ ]	Indicate optional information
braces { }	Indicate that you must select one of the enclosed entries

GCOS/BES2 SUBJECT DIRECTORY

This subject directory is designed to assist the user in finding information about specific topics related to GCOS/BES2. Topics are listed alphabetically; each topic is accompanied by the order number of each manual in which the topic is described. At the end of the Subject Directory, all GCOS/BES2 manuals are listed according to the alphabetic/numeric sequence of their order numbers.

The following publications constitute the GCOS/BES2 manual set. The subject Directory in the latest Series 60 (Level 6) GCOS/BES2 Software Overview and System Conventions manual lists the current revision number and addenda (if any) for each manual in the set.

Order No.	Manual Title
AS32	Series 60 (Level 6) GCOS/BES FORTRAN Reference Manual
AU41	Series 60 (Level 6) GCOS/BES2 COBOL Reference Manual
AU43	Series 60 (Level 6) GCOS/BES2 Assembly Language Reference Manual
AU44	Series 60 (Level 6) GCOS/BES2 BASIC Reference Manual
AU45	Series 60 (Level 6) GCOS/BES2 Executive and Input/Output
AU46	Series 60 (Level 6) GCOS/BES2 Operator's Guide
AU47	Series 60 (Level 6) GCOS/BES2 Utility Programs
AU48	Series 60 (Level 6) GCOS/BES2 Program Development Tools
AU49	Series 60 (Level 6) GCOS/BES2 Planning and Building an Online Application
AU50	Series 60 (Level 6) GCOS/BES2 Software Overview and System Conventions

In addition to the GCOS/BES2 manual set, the following manual is required by GCOS/BES users as a general hardware reference:

Order No.	Manual Title
AS22	Honeywell Level 6 Minicomputer Handbook

The following manual provides detailed information regarding programming for the Multiline Communications Processor:

Order No.	Manual Title
AT97	Series 60 (Level 6) MLCP Programmer's Reference Manual

CONTENTS

Part I
Introduction

ILLUSTRATIONS

TABLES

PART I

INTRODUCTION

SECTION 1
INTRODUCTION

All of the procedures for operating the BES programs appear in this manual. Before performing any bootstrapping and loading procedure, you should become familiar with the control panel and other peripherals required to load and execute programs. Read Part II for the ways in which you can enter and retrieve data from the system, and pertinent information on using the control panel, the system or operator's console, and other peripherals.

Parts III, IV and V of the manual specify the operating procedures for setting up, bootstrapping and loading, and executing system programs and online and offline application programs. Table I-1-1 summarizes the different loading procedures described in this manual. The table indicates the two types of loaders used, the Disk Loader and the Paper Tape Loader, and the programs with which they are used. Availability of the system or operator's console in specific loading environments is also indicated.

The different command input devices that can be used to submit commands to the Configuration Load Manager (CLM) to be used in configuring and loading an online application are indicated in Table I-1-1. The devices to which the command input file for a system program can be assigned are listed in the file requirements table in the section of Part III that describes that system program.

Refer to Table I-1-1 for loading techniques that are available for each type of program, and the part and section of this manual in which each loading procedure is specified. You can use the table as a quick reference aid to locate any specific loading procedure. Each loading procedure is described in its entirety, including:

- The complete bootstrapping and loading procedure for that table entry
- All setup information and file requirements for the program or component being loaded
- All messages the program or component may issue to the operator

While certain procedures, such as bootstrapping, may be similar from program to program, each procedure is presented fully for each load environment, so that you can locate in one place all information pertinent to that load environment.

Table I-1-1. Loading Procedures for Online and Offline Application Programs and System Programs

Load Device	System/ Operator's Console Available?	Online User Program (With CLM)			Offline System Programs	Offline User Programs
		Command Input Device:				
		KSR	CR	DSK		
Disk	YES	IV-2	IV-2	IV-2	III-1	V-2
	NO	NO	IV-2	IV-2	NO	V-2
Paper tape	YES	IV-3	IV-3	IV-3	NO	V-3

NOTE: Table entries consist of a two-part number identifying the location in this manual in which the procedure is specified, as follows:

Roman numeral - Number of part in this manual

Arabic number - Number of section in the part

The entry NO signifies that the program cannot be loaded under the specified circumstances.

All messages issued by a program or component are listed in the section that describes the operating procedures for that program or component. For each program or component, two tables are specified, if applicable:

- One table of coded error messages, arranged numerically
- One table of uncoded error and informational messages, arranged alphabetically

To locate the description of a particular message, consult the section of this manual that describes the program or component you are currently running. To determine which program or component issued a coded error message, refer to Appendix G for a list of the component numbers that appear as the first two digits of the 6-digit error code.

PART II

OPERATOR INTERFACE WITH SOFTWARE AND HARDWARE

SECTION 1
CONTROL PANEL

The full control panel provides a means of powering up the system, controlling the central processor, accessing registers and memory, and initiating different central processor states. A basic control panel that provides limited functionality is also available. It can be used to power up and initialize the system and to perform automatic bootstrapping and loading of a program into memory, but cannot be used to access registers or memory or control the central processor.

Each control panel configuration is described separately below.

DESCRIPTION OF FULL CONTROL PANEL

The full control panel permits the operator to power up and initialize the system, initiate bootstrapping, start and stop program execution, single step a program, enter and display registers/memory, and indicate central processor status. The layout of the full control panel appears in Figure II-1-1.

Functional Elements of Full Control Panel

The following functional areas can be identified on the full control panel:

1. Power/panel security switches (used also in the basic control panel configuration)
2. Control keys
3. Indicator lamps
4. Register display
5. Hexadecimal-pad keys

POWER/PANEL SECURITY SWITCHES

Two switches are located on the left side of the panel, one for system power, the other for panel security. Refer to Table II-1-1 for a description of each switch.

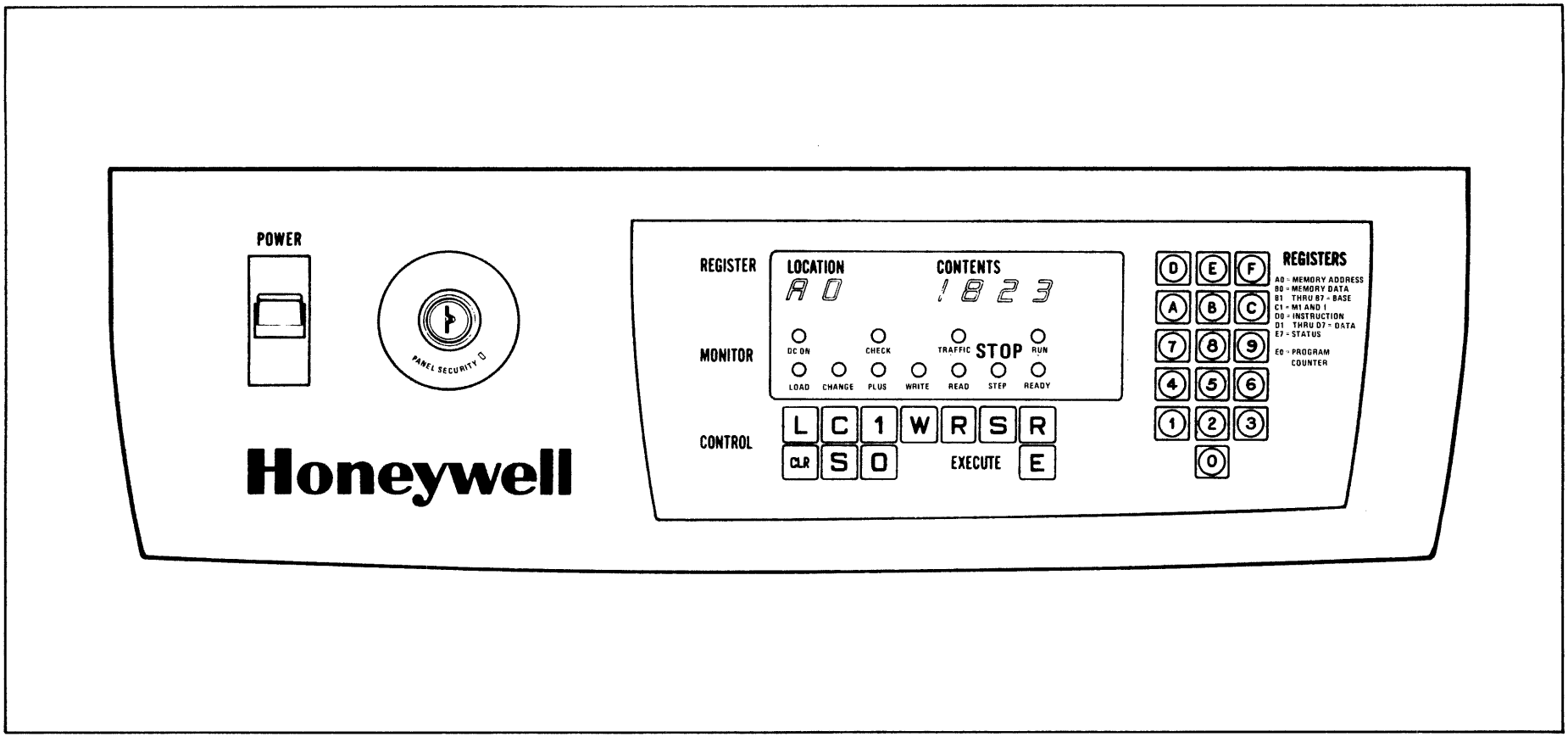


Figure II-1-1. Layout of Full Control Panel

Table II-1-1. Control Panel Power/Panel Security Switches

Switches	Description
Power	A two-position switch that engages or disengages system power. Place switch up for power on, down for power off. When power is switched on, the DC ON indicator will illuminate after dc power is attained. Also during power-up, the master clear procedure is automatically performed (refer to "Master Clear" later in this section).
Panel Security	A two-position switch that is activated by inserting the panel key and turning left for panel lock, right for panel unlock. When panel is locked, all panel switches and keys are inoperable, except for power, and the register display is disabled (not illuminated). When the panel is unlocked, the register display illuminates, indicating that the panel is operational.

CONTROL KEYS

Eleven push-button keys are located in the lower center segment of the panel in the area labeled CONTROL. These keys are employed to place the system in various states and/or modes of operation. The processor can be placed in the following primary operating states:

- Stop state - Processor is not executing instructions
- Run state - Processor is executing instructions

Within the stop and run states, the processor can be placed in various operating modes, which are described in this section.

Figure II-1-2 is an annotated closeup view of the control key area with each key identified by its function. In this manual, control keys are referred to by their function, with the actual label of the key underlined; e.g., Load, Plus 1, Read.

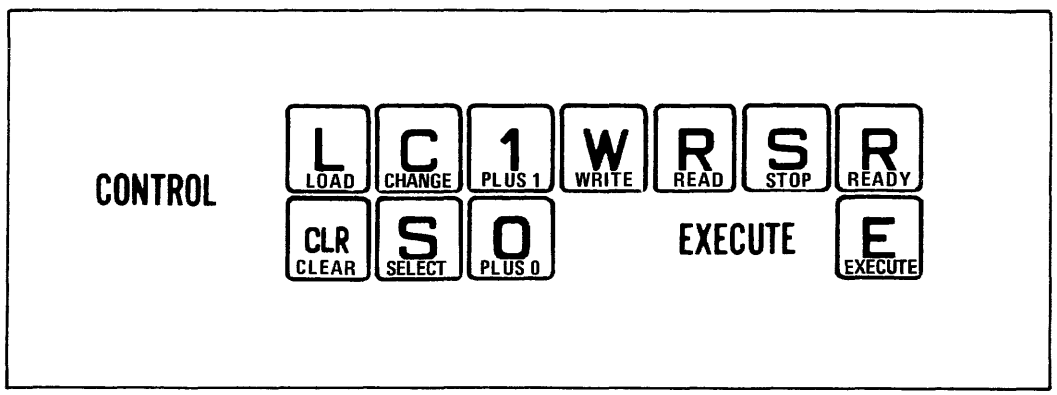


Figure II-1-2. Control Keys Identified by Function

Each control key and the functions it performs is described in Table II-1-2. The selection codes for all registers referred to in this table are shown in Table II-1-4 later in this section.

Table II-1-2. Control Panel Control Keys

Key	Function	Description
CLR	CLeaR	Pressing this key while in the stop state causes master clearing of the processor to be initiated. The following clearing and initializing functions are performed when CLeaR is pressed: • P-register, M1 register, and instruction register are cleared and set to 0. • All pending interrupts are cleared. • Real-time clock (RTC) and watchdog timer (WDT) are stopped. • Privileged state is set to 1 and interrupt priority level is set to 0. • Quality Logic Test (QLT) is started in each controller.
L	Load	Pressing this key places the processor in load mode and ready mode and allows running of the central processor QLT and bootstrapping of the bootstrap record into memory. The Load key is used in conjunction with Execute; when the Execute key is pressed, the central processor/memory QLT is run and then bootstrapping is performed.
S	Select	Pressing this key places the processor in select mode. In this mode, a register selection code can be keyed in, using the hexadecimal-pad keys, to designate a specific register that is to be acted upon or whose contents are to be displayed. Select mode may be initiated while the processor is in any state.
C	Change	Pressing this key places the processor in change mode. In this mode, modifications to the contents of the selected register can be keyed in using the hexadecimal-pad keys; the central processor must not be in run state.
1	Plus 1	Pressing this key places the processor in plus 1 mode. In this mode, each time the Execute key is pressed, the memory address register (A0) is incremented by 1 before the next memory location is read or written. The processor must be in read or write mode before Plus 1 is pressed.
0	Plus 0	Pressing this key resets the plus 1 mode. In this mode the memory address register is not modified during a memory read or write operation.
W	Write	Pressing this key first places the processor in a stop state, if not already in that state; resets the plus 1 mode, if in plus 1 mode; resets the load mode, if in load mode; and places the processor in write mode. In this mode, when the Execute key is pressed the processor writes the contents of the selected register into the location addressed by the memory address register (A0).

Table II-1-2 (cont). Control Panel Control Keys

Key	Function	Description
R	Read	Pressing this key first places the processor in a stop state, if not already in that state; resets the plus 1 mode, if in plus 1 mode; resets the load mode, if in load mode; and places the processor in read mode. In this mode, when the Execute key is pressed the processor reads into the selected register the contents of the location addressed by the memory address register (A0).
S	Stop	Pressing this key stops instruction execution and places the processor in a stop state. When the processor is in the stop state, various operating procedures can be performed through the control panel (refer to "Operating Procedures for Full Control Panel" below). Also, when in the stop state, the processor is automatically in step mode. In step mode, one instruction is executed each time the Execute key is pressed.
R	Ready	Pressing this key places the processor in ready mode. In this mode, the processor can execute a series of instructions constituting a program. If the Execute key is pressed, the processor enters the run state and commences execution of the program.
E	Execute	Pressing this key causes various execution functions to be performed; the functions performed depend on which operating mode the processor is in at the time the Execute key is pressed. If the processor is in ready mode, pressing Execute places the processor in a run state and it executes instructions starting with the instruction in the instruction register. If the instruction register contains a zero, execution will begin at the address specified in the P-register. Execution continues until a halt instruction is encountered or the Stop, Read or Write key is pressed. If the processor is in step mode, pressing Execute causes a single instruction to be executed. The processor remains in step mode after the single instruction is executed. If the processor is in read or write mode, whenever Execute is pressed, the selected memory location is displayed or changed. If the processor is in load mode, pressing Execute causes the QLT to be run. After the QLT has terminated, pressing Execute causes a bootstrapping operation to occur.

INDICATOR LAMPS

Eleven illuminating lamps are located in the middle center portion of the panel in the area labeled MONITOR. These lamps indicate status and operating mode of the system. Refer to Table II-1-3 for a description of each indicator lamp.

Table II-1-3. Control Panel Indicator Lamps

Indicators	Description
DC ON	When lit, this lamp indicates that operational dc power is on in the system.
CHECK	When lit, this diagnostic lamp indicates that at least one bus element has not successfully completed its logic test (QLT) or a bus element is not properly plugged into the bus.
TRAFFIC	When lit, this lamp indicates that the processor is executing instructions (excluding the halt instruction).
RUN	When lit, this lamp indicates that the processor is in the run state; i.e., the processor is executing programs. If the TRAFFIC lamp goes off while the RUN lamp is lit, this condition indicates that the processor is continually executing a halt instruction.
LOAD	Lamp illuminates when processor is placed in load mode. Lamp goes off when load operation is successfully completed.
CHANGE	Lamp illuminates when processor is placed in change mode. When the lamp is lit, the contents of a selected register may be changed by keying in data from the hexadecimal-pad keys, provided that the processor is not in run state.
PLUS	This lamp lights when the Plus 1 key is pressed. When lit, sequential memory locations may be read or written. The lamp extinguishes when the Plus 0, Read or Write key is pressed.
WRITE	Lamp illuminates when processor is placed in write mode. When the lamp is lit, data entered through the control panel can be written into memory.
READ	Lamp illuminates when processor is placed in read mode. When the lamp is lit, data may be read from memory via the control panel.
STOP/STEP	This lamp lights when the processor is placed in the stop state; i.e., it is no longer executing instructions. When the lamp is lit, the processor is in a step mode. In this mode, one instruction will be executed each time the Execute key is pressed.
READY	This lamp lights when the Ready key is pressed, causing the processor to be placed in a ready mode. If the Execute key is pressed next, the processor will enter the run state.

REGISTER DISPLAY

Twenty-one registers may be accessed from the control panel. A 6-digit hexadecimal display located in the upper center portion of the control panel, in the area labeled REGISTER, accommodates this capability. The register display is divided into two sections, one labeled LOCATION, the other CONTENTS:

- LOCATION - A 2-digit hexadecimal display that indicates the coded location of the specific register selected. The selection codes assigned to access the visible registers are listed alphanumerically in Table II-1-4. A complete list of the selection codes is also stenciled on the control panel in the upper righthand portion of the panel labeled REGISTERS.
- CONTENTS - A 4-digit hexadecimal display that indicates the contents of the selected register. The specific visible register type/number associated with the assigned selection code is listed in the contents column of Table II-1-4.

Table II-1-4. Register Selection Codes

LOCATION SELECTION CODES		CONTENTS VISIBLE REGISTERS
A	0	MEMORY ADDRESS REGISTER (0–15)
B	0	MEMORY DATA REGISTER (0–15)
B	1	B1 (0–15)
B	2	B2
B	3	B3
B	4	B4
B	5	B5
B	6	B6
B	7	B7
C	1	M1 REGISTER (0–8) | I-REGISTER (8–15)
D	0	INSTRUCTION REGISTER (0–15)
D	1	R1 (0–15)
D	2	R2
D	3	R3
D	4	R4
D	5	R5
D	6	R6
D	7	R7
E	0	PROGRAM COUNTER (P-REGISTER) (0–15)
E	7	S-REGISTER (0–15)
H1 | H2		H3 (0–4) | H4 (4–8) | H5 (8–12) | H6 (12–15)

HEXADECIMAL-PAD KEYS

Sixteen push-button keys are located on the right side of the control panel. These keys supply hexadecimal digit input to the processor in order to access, display, and/or change the visible registers. The entry is immediately displayed on the REGISTER display as each hexadecimal digit is keyed in.

In the select mode, inputs from the hexadecimal-pad keys select the register to be displayed/operated on. These inputs are simultaneously displayed in the LOCATION field on the display. Hexadecimal-pad keys 8 through F modify the leftmost location character (H1). Hexadecimal-pad keys 0 through 7 modify the rightmost location character (H2).

In the change mode, inputs from the hexadecimal-pad keys change the contents of the selected register. These inputs are simultaneously displayed in the CONTENTS field of the register display. Each key stroke shifts and loads the corresponding hexadecimal character into the least significant hexadecimal position of the selected register and the display. Specifically, each hexadecimal-pad key stroke enters the new character into the H6 position of the selected register and the CONTENTS field of the register display. At the same time, the currently displayed characters (in all four positions) shift one position to the left.

The activity resulting from one key stroke is illustrated as follows:

$$\text{out} \leftarrow H3 \leftarrow H4 \leftarrow H5 \leftarrow H6 \leftarrow \text{enter}$$

You may take advantage of this shifting function and key in only a limited number of characters to arrive at the contents desired. For example, if the current CONTENTS display shows 4032 and you wish to change the contents to 3275, then key in only the characters 7, 5.

Panel Display Interpretation

The CONTENTS field on the display panel is shown in hexadecimal notation. The 4-character hexadecimal display value represents the binary value of the 16-bit visible register. Each hexadecimal character is equivalent to a binary value of four bits. Thus if the display shows a value of 4CA2, this hexadecimal value represents the stored binary value of

$$\underset{4}{0100}\ \underset{C}{1100}\ \underset{A}{1010}\ \underset{2}{0010}$$

For most registers, the display value is usable in hexadecimal form and does not need to be converted to binary. However, the exceptions are the M1, I- and S-registers. These registers specify various status, security and control indicators on a bit basis. Therefore, to properly interpret these registers, you must convert their hexadecimal displays to binary. Refer to Table II-1-5 for a list of hexadecimal/binary/decimal conversions for the first 16 digit values.

Table II-1-5. Hexadecimal/Binary/Decimal Conversion

Hexadecimal	Binary	Decimal
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
A	1010	10
B	1011	11
C	1100	12
D	1101	13
E	1110	14
F	1111	15

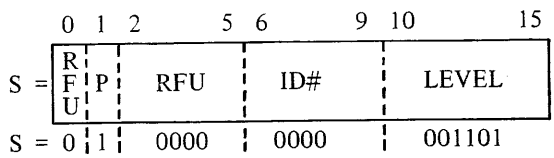
As an example, assume you wish to analyze the contents of the S-register. After you have selected and keyed in the proper selection code (E7), the display panel shows the following hexadecimal character display:

LOCATION	CONTENTS
E7	400D

For the information to be meaningful, first you must convert the hexadecimal value to binary:

hexadecimal 400D = binary 0100 0000 0000 1101

Next, you must overlay this binary value onto the bit fields structured for the S-register:



You can now analyze the system status and security indicators:

- P = 1 indicates that the system is in a privileged state
- ID# (processor identity) = 0000 indicates CP channel 0
- LEVEL (interrupt priority level) = 001101 when converted to decimal (binary 001101 = decimal 13) indicates the actual priority level:

  Interrupt priority level = 13

- RFU = Reserved for future use

OPERATING PROCEDURES FOR FULL CONTROL PANEL

This subsection describes some of the basic operating processes that may be performed with the full control panel. These sample processors also demonstrate the functionality of the various panel elements.

The functions that can be performed from the control panel are governed by the operating state of the processor. For example, in relation to program execution, when the processor is in the run state, only the limited actions of displaying registers and executing programs are possible from the control panel. When the processor is in the stop state, the actions possible are much more extensive; they consist of displaying/changing memory, displaying/changing registers, executing single instructions, restarting programs, and master clearing the processor.

Before you perform any operation from the panel (except for power on/off), you must first unlock the control panel using the panel security switch.

Display Memory

Any memory location may be accessed and displayed on the control panel. However, memory can be displayed only when the processor is in a stop state. The memory address register (A0) is the only visible register that can be used to access memory locations from the control panel. Any visible data register may be used for reading/displaying memory data, but by convention the memory data register (B0) is usually used for this purpose and will be so used in this procedure.

The following procedure describes a method for displaying the contents of one memory location and, as an option, displaying the contents of subsequent memory locations.

1. Press Read.

   This action initially places the processor in a stop state, if the processor is not as yet in this state. Then the processor is placed in read mode and instructed that the contents of the memory location addressed by the memory address register (A0) are to be displayed. The READ indicator will illuminate when the Read control key is pressed.

2. Press Select.

   This action places the processor in select mode. This step is necessary only if the CHANGE indicator is lit.

3. Press hexadecimal-pad keys A, 0 to enter the 2-digit selection code for the memory address register. Digits A, 0 will appear in the LOCATION field of the REGISTER display.

4. Press Change.

   This action places the processor in change mode, which must be in effect before you key in the address of the memory location to be displayed. The CHANGE indicator will illuminate when the Change control key is pressed.

5. Key in, via the hexadecimal-pad keys, the 4-digit hexadecimal value representing the address of the memory location to be read. This address will appear in the CONTENTS field of the REGISTER display.

6. Press Select.

   This action returns the processor to select mode, which must be in effect before you select the memory data register. The CHANGE indicator will go off when the Select control key is pressed.

7. Press hexadecimal-pad key B to enter the first digit of the 2-digit selection code for the memory data register. The second digit need not be entered, since the hexadecimal character 0, entered in step 3, is still present in position H2. Digits B 0 will show in the LOCATION field of the REGISTER display.

8. Press Execute.

   When this action is initiated, the data contents of the selected memory location will be loaded into the selected register (B0) and displayed in the CONTENTS field of the REGISTER display.

9. If successive memory locations are to be displayed using the current address of the memory address register as a base, press Plus 1. The PLUS indicator will illuminate when the Plus 1 control key is pressed.

10. Press Execute.

    When this action is taken, the memory address register will be incremented by 1 and the memory data of the succeeding memory location will appear in the CONTENTS field of the REGISTER display.

11. Repeat step 10 for each sequential memory location to be displayed.

Change Memory

Any memory location may be accessed and changed from the control panel. However, memory can be changed only when the processor is in write mode. As mentioned previously, the memory address register (A0) is the only visible register that can be used to access memory locations from the control panel. Any visible data register may be used for writing/changing memory data, but by convention the memory data register (B0) is usually used for this purpose and will be so used in this procedure.

The following procedure describes a method for changing the contents of one memory location and, as an option, changing the contents of subsequent memory locations.

1. Press Write.

   This action initially places the processor in a stop state, if the processor is not already in this state. Then the processor is placed in write mode and instructed that the contents of the memory location addressed by the memory address register (A0) are to be changed. The WRITE indicator will illuminate when the Write control key is pressed.

2. Press Select.

   This action places the processor in select mode, which must be in effect before you select the memory address register. This step is necessary only if the CHANGE indicator is lit.

3. Press hexadecimal-pad keys A, 0 to enter the 2-digit selection code for the memory address register. Digits A 0 will appear in the LOCATION field of the REGISTER display.

4. Press Change. This action places the processor in change mode, which must be in effect before you key in the address of the memory location to be changed. The CHANGE indicator will illuminate when the Change control key is pressed.

5. Key in, via the hexadecimal-pad keys, the 4-digit hexadecimal value representing the address of the memory location to be changed. This address will appear in the CONTENTS field of the REGISTER display.

6. Press Select.

   This action returns the processor to select mode, which must be in effect before you select the memory data register. The CHANGE indicator will go off when the Select control key is pressed.

7. Press hexadecimal-pad key B to enter the first digit of the 2-digit selection code for the memory data register. The second digit need not be entered, since the hexadecimal character 0, entered in step 3, is still present in position H2. Digits B 0 will show in the LOCATION field of the REGISTER display.

8. Press Change.

   This action places the processor in change mode, which must be in effect before you key in data for the memory location that is to be changed. The CHANGE indicator will illuminate when the Change control key is pressed.

9. Key in, via the hexadecimal-pad keys, the 4-digit hexadecimal value representing the new data that is to be entered into the memory location to be changed. The data entered will appear in the CONTENTS field of the REGISTER display.

10. Press Execute.

    When this action is initiated, the new data contents will be loaded into the selected memory location.

11. If successive memory locations are to be changed using the current address of the memory address register as a base, press Plus 1. The PLUS indicator will illuminate when the Plus 1 control key is pressed.

12. Repeat steps 9 and 10 for each sequential memory location to be changed. Note that while in plus 1 mode (PLUS indicator lit), each time the Execute key is pressed the memory address register is incremented by one before the new data contents are loaded into the new incremented memory location.

Display Registers

The contents of any one of the 21 visible registers may be displayed on the control panel. A register may be displayed when the processor is in any state.

The following procedure describes a method for displaying the contents of one register. The same procedure applies regardless of the processor state.

1. Press Select.

   This action places the processor in select mode, which must be in effect before you select the register to be displayed.

2. Key in, via the hexadecimal-pad keys, the 2-digit selection code (as specified in Table II-1-4) for the desired register to be displayed. The selection code will appear in the LOCATION field of the register display. When the desired selection code is keyed in, the data contents of the selected register will be displayed in the CONTENTS field of the REGISTER display. If the processor is in the run state and a register is selected, the updated contents of the register are displayed continuously if the panel is unlocked.

Change Registers

The contents of 18 of the 21 visible registers may be changed from the control panel (the M1, I-, and S-registers cannot be modified from the control panel). A register may be changed only when the processor is in a stop state. The following procedure describes a method for changing the contents of one register.

1. Press Stop if the processor is in the run state. The STOP/STEP indicator will illuminate when the Stop control key is pressed.

2. Press Select.

   This action places the processor in select mode, which must be in effect before selecting the register to be changed.

3. Key in, via the hexadecimal-pad keys, the 2-digit selection code (given in Table II-1-4) for the desired register to be changed. The selection code will appear in the LOCATION field of the REGISTER display. When the desired selection code is keyed in, the current data contents of the selected register will be displayed in the CONTENTS field of the REGISTER display.

4. Press Change.

   This action places the processor in change mode, which must be in effect before you key in the data to the register that is to be changed. The CHANGE indicator will illuminate when the Change control key is pressed.

5. Key in, via the hexadecimal-pad keys, the hexadecimal value representing the new data that is to be entered into the selected register. The data entered will appear in the CONTENTS field of the REGISTER display and be loaded into the selected register.

Stop Program Execution

While a program is running, you can stop program execution at any time by pressing Stop. The STOP/STEP indicator will illuminate and the RUN, READY, and TRAFFIC indicators will go out. When Stop is pressed, the processor completes executing the current instruction and enters the stop state. When the processor is in the stop state, the following pertinent conditions exist:

1. The processor is automatically placed in step mode; i.e., it is ready to execute one instruction at a time.
2. The instruction register (D0) contains the instruction to be executed next.
3. The P-register (E0) contains an address incremented by 1 from the address of the instruction to be executed next.

Note that when a program is running and a halt instruction is encountered (RUN indicator remains lit, but TRAFFIC indicator goes out), the processor does not enter the stop state; it enters an idle condition. In this condition, the halt instruction is continuously reexecuted and the processor is subject to external interrupts, etc. To place the processor in a stop state, press the Stop key.

Execute Single Instruction(s)

While running a program, you may wish to stop processing and step through the execution of one or more instructions. This procedure is accomplished from the control panel, as follows:

1. Press Stop.

   Refer to the previous procedure, "Stop Program Execution," for relevant information concerning processor/panel status after a stop state is attained.

2. Determine whether the processor has stopped at a point (address) from which you wish to begin executing single instructions. Display and view the contents of the P-register (E0) using the procedure previously described for displaying registers. The P-register (E0) will contain an address incremented by 1 from the address of the instruction to be executed next. If this address is one more than the point at which you wish to start executing single instructions, proceed to step 8. However, if a new starting point is desired, continue to the next step.

3. Press CLeaR.

   This action sets the instruction register (D0) and the P-register (E0) to zero; this step must be performed before you specify a new starting address.

   NOTE: In an online environment, do not press CLeaR. Instead, select the instruction register (D0) and change its contents to 0000_{16} so that peripheral devices are not affected.

4. Press Select.

   This action places the processor in select mode, which must be in effect before you select the P-register (E0).

5. Press hexadecimal-pad keys E, 0 to enter the 2-digit selection code for the P-register. Digits E 0 will appear in the LOCATION field of the REGISTER display.

6. Press Change.

   This action places the processor in change mode, which must be in effect before you key in the address of the instruction to be executed next. The CHANGE indicator will illuminate when the Change control key is pressed.

7. Key in, via the hexadecimal-pad keys, the 4-digit hexadecimal value representing the address of the next instruction to be executed. This address will be entered in the P-register and will appear in the CONTENTS field of the REGISTER display.

8. Press Execute.

   When this action is initiated, first an attempt is made to execute the instruction contained in the instruction register (D0). If the instruction register contains an executable instruction, then that instruction is executed. After it is executed, the next succeeding instruction is placed in the instruction register, the P-register (E0) is incremented by 1, and the processor is returned to step mode.

   If the instruction register (D0) contains the value zero, the one instruction addressed by the P-register (E0) is fetched and executed. The contents of the P-register are incremented by 1 and the instruction addressed by the P-register is placed into the instruction register. Finally, the contents of the P-register are incremented by 1 again (i.e., the address of the instruction residing in the instruction register, plus 1) and the processor is returned to the step mode.

9. Repeat step 8 for each successive instruction to be executed singly. At any time after executing a single instruction, you may return to the run state (see the next procedure on restarting programs).

Restart Program

A program may be restarted from the control panel at any time and at any point after it has been stopped during execution. However, when you are restarting a program, it is your responsibility to ensure that:

- No I/O was pending at the time the machine was halted.
- The registers are restored to the values they contained when the program stopped executing.

The restart procedure is as follows:

1. Determine whether the current start address is the point at which you wish to restart the program. Display the contents of the P-register (E0), using the procedure previously described for displaying registers. The P-register will contain an address incremented by 1 from the address of the instruction to be executed next. If this address is one more than the point at which you wish to restart the program, then proceed to step 7. However, if a new starting point is desired, continue to the next step.

2. Press CLeaR.

   This action sets the instruction register (D0) and the P-register (E0) to zero; this step must be performed before you specify a new starting address.

3. Press Select.

   This action places the processor in select mode, which must be in effect before you select the P-register (E0).

4. Press hexadecimal-pad keys E, 0 to enter the 2-digit selection code for the P-register. Digits E 0 will appear in the LOCATION field of the REGISTER display.

5. Press Change.

   This action places the processor in change mode, which must be in effect before you key in the address of the instruction from which you wish to restart execution of the program. The CHANGE indicator will illuminate when the Change control key is pressed.

6. Key in, via the hexadecimal-pad keys, the 4-digit hexadecimal value representing the restart address. This address will be entered into the P-register and will appear in the CONTENTS field of the REGISTER display.

7. Press Ready.

   This action places the processor in ready mode; i.e., it instructs the processor that program execution is to begin when the Execute key is pressed. The READY indicator will illuminate and the STOP indicator will go out when the Ready control key is pressed.

8. Press Execute.

   When this action is initiated, the processor is placed in a run state and will attempt to start executing the program beginning with the instruction contained in the instruction register (D0).

   If the instruction register contains an executable instruction, program execution will begin with the instruction specified by the instruction register. If the instruction register contains the value zero, program execution will begin with the instruction addressed by the P-register (E0).

   The RUN and TRAFFIC indicators will illuminate after the Execute key is pressed.

   Program execution will continue until a software halt is encountered or the Stop, Read or Write key is pressed.

Master Clear

The master clear procedure is used to set or restore the processor to a standard initialized state as a prelude to certain functions such as a bootstrap procedure. The following procedure indicates the master clearing operation:

1. Press Stop if the processor is not already in the stop state. Master clear is only operative when the processor is not in the run state. The STOP/STEP indicator will illuminate when the Stop control key is pressed.

2. Press CLeaR.

   The specific functions that are activated by pressing CLeaR are listed in Table II-1-2, under the description of the CLeaR control key.

DESCRIPTION OF BASIC CONTROL PANEL

The basic control panel provides a means of powering up and initializing the system. An initialize switch on the panel can be used to master clear the system, cause all QLT's to run, and then bootstrap into memory a bootstrap record that automatically causes bootstrapping and loading of the program specified in the bootstrap record. The processor then begins program execution. Refer to Appendix F for the procedure for loading a program using the basic control panel. A layout of the entire basic control panel appears in Figure II-1-3.

Functional Elements of Basic Control Panel

The following functional areas may be identified on the basic control panel:

1. Power/panel security switches (used also with the full control panel)
2. An initialize switch
3. Indicator lamps

POWER/PANEL SECURITY SWITCHES

Two switches are located on the left side of the panel, one for system power, the other for panel security. Refer to Table II-1-6 for a description of each switch.

Table II-1-6. Basic Control Panel Power/Panel Security Switches

Switches	Description
Power	A two-position switch that engages or disengages system power. Place switch up for power on, down for power off. When power is switched on, the DC ON indicator will illuminate after dc power is attained. Also during power-up, the master clear procedure is automatically performed (refer to "Master Clear" earlier in this section).
Panel Security	A two-position switch that is activated by inserting the panel key and turning left for panel lock, right for panel unlock. When panel is locked, the initialize switch is inoperable.

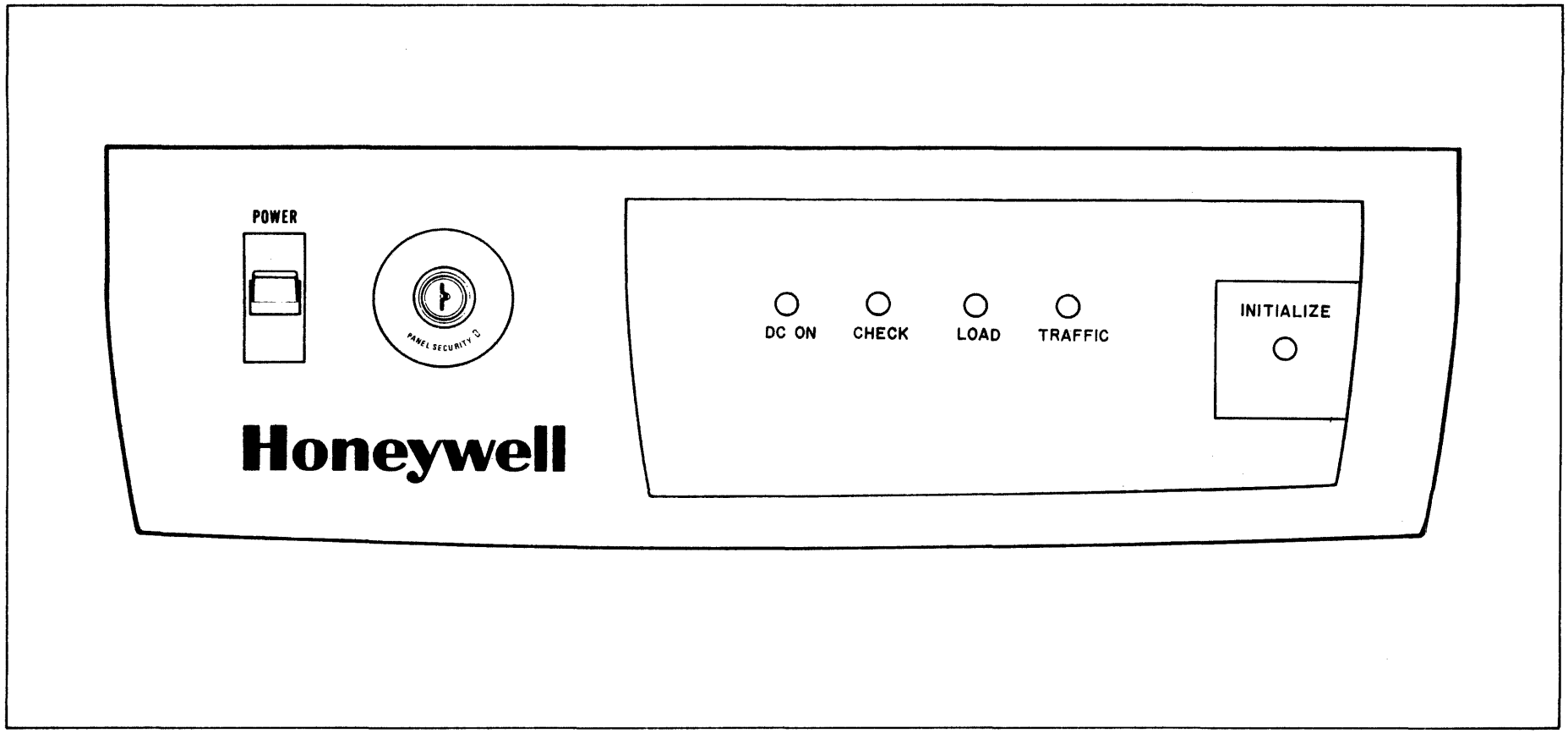


Figure II-1-3. Layout of Basic Control Panel

INITIALIZE SWITCH ON BASIC CONTROL PANEL

The initialize switch is a push switch on the right side of the control panel. If the panel is unlocked, pressing the initialize switch initializes the system, runs the central processor QLT, places the central processor in load mode, and causes bootstrapping of the bootstrap record into memory. The bootstrap device channel number is 0400_{16}. The bootstrap record causes the program to be read into memory, relocating it according to the relocation factor preset in the bootstrap record.

NOTE: It is the user's responsibility to prevent the program being loaded from overlaying the hardware dedicated memory area.

The program then begins execution. (See Appendix F for the procedure for loading a program using the basic control panel.)

INDICATOR LAMPS ON BASIC CONTROL PANEL

Four illuminating lamps are located in the center portion of the panel. These lamps indicate status and operating mode of the system. Refer to Table II-1-7 for a description of each indicator lamp.

Table II-1-7. Basic Control Panel Indicator Lamps

Indicator	Description
DC ON	When lit, this lamp indicates that operational dc power is on in the system.
CHECK	When lit, this diagnostic lamp indicates that at least one bus element has not successfully completed its logic test (QLT) or a bus element is not properly plugged into the bus.
LOAD	Lamp illuminates when processor is placed in load mode. Lamp goes off when load operation is successfully completed.
TRAFFIC	When lit, this lamp indicates that the processor is executing instructions.

SECTION 2
OPERATOR COMMUNICATION WITH THE SOFTWARE

Two-way communication between the operator and the system is achieved by means of the control panel (described in Section 1 above) and keyboard input/output devices (the KSR-33 teleprinter, the Keyboard Typewriter Console, and the CRT Keyboard Console Unit). Input can also be submitted to the system through other peripherals.

SYSTEM PROGRAMS

When you are loading and running a system program, you must submit commands to the software through a device that was assigned as the command input device during the loading process. Possible devices include:

- System console (a KSR-like device whose channel number is stored in the loader communication area in memory)
- Another KSR-like device, card reader or disk specified as the command input device

If the assigned command input device is the system console or another KSR-like device, you can carry on an interactive dialog with a system program, in this form:

- You submit commands to the program through the device keyboard.
- The program displays responses to commands, informational messages and error messages on the device.

If the assigned command input device is the card reader or a disk, you submit commands on cards or disk, and the program displays these commands on the system console. When an error occurs, the program displays an error message on the system console and returns control to the Command Processor.

The alternative command input devices that can be assigned to a particular system program are listed under the section that describes operating procedures for that program (see Part III).

APPLICATION PROGRAMS

During loading of an online or offline application program, you can submit commands to the software through a KSR-like device used as an operator's console (the channel number of the device must have been stored in the loader communication area during loading), or if a card reader, disk, or another KSR-like device has been designated as the command input device, through that device.

If you are configuring and loading an online application, you can submit commands to the loaded Configuration Load Manager (CLM) through the same command input device you used in loading the CLM, or you can assign a different command input device to CLM during the loading process.

Once the online application loaded by the CLM is executing, you can conduct an interactive dialog initiated by any task running under the Executive, using a KSR-like device assigned as an operator's console (see the Executive and Input/Output manual). Possible communications devices include the KSR terminal, CRT terminal, and visual information projection (VIP) system (see the Planning and Building an Online Application manual). When the communications subsystem is included in an online configuration, you can assign a KSR-like communications terminal as an operator's console. Communications terminals are not supported in an offline environment.

SECTION 3
PERIPHERAL SETUP AS RELATED TO SOFTWARE OPERATIONS

This section presents restrictions and special notes on the use of peripherals with the software. Comments on the use of KSR-like devices (KSR-33, Keyboard Typewriter Console, or CRT Keyboard Console Unit) apply to both offline and online environments, including the configuration in which the peripheral is being used as a communications device.

CHANNEL NUMBERS

Devices attached to the Multiple Device Controller (MDC) are software addressable via channel numbers. Channel numbers are physically set for a particular system prior to or at installation time. Software is independent of channel numbers and may accommodate any setting.

ERROR MESSAGES ON KSR-LIKE DEVICES

If error messages are assigned to a KSR-like device and either the device is inoperable or no error message is printed on the device, then the ensuing results are unspecified. Under these conditions, no other way exists to inform the user of a device error and a halt may not be executed.

TELEPRINTER (KSR-33)

System Operation of KSR-33

The control switch must be set to the LINE position in order to use the terminal with the system.

Carriage Return

Each time the carriage return key is pressed, a second or echo carriage return is generated by the system firmware. The double shift motion of the carriage should cause no concern or alarm.

Special Editing Characters

The system treats characters @ and CAN as special characters. These characters are allocated the following functions:

@ (shift P): Delete one or more characters, one character at a time, in reverse order; for example,

```
TSET@@@EST = TEST
```

To produce the @ character on the KSR, press and hold the SHIFT key and then press the P key.

CAN (CTRL X): Delete entire line just keyed in. To produce the CAN character on the KSR press and hold the CTRL key and then press the X key. The CAN character is the nonprintable control character of X.

Since the offline Debugger works in character mode, the characters @ and CAN cannot perform their special functions in this mode.

The \ character causes the characters @, CAN, CR and \ to be treated as data input and not as editing characters.

KEYBOARD TYPEWRITER CONSOLE

Movement of Print Head

When characters are typed slowly (at a rate of 0.8 seconds or more per character), after each key is pressed the print head will move four characters to the right, to provide visibility of the last character typed. At normal typing speed, the print head does not move to the right after each key is pressed.

DISKETTE

Read/Write Errors on a Diskette

During program development, program execution terminates when uncorrectable read or write errors are encountered on the diskette. An error is specified only after three retries have been unsuccessfully attempted and control is returned to the Command Processor or the individual component.

Write Protection

Since diskettes do not have write protection, precautionary measures should be taken to prevent inadvertent destruction of useful information.

Defective Sectors

When a new or used diskette is being initialized (using the volume initialization command in Utility Set 1) and an abnormal number of sectors (15 to 35) are detected and reported as defective, this may be due to the improper seating of the diskette device head. If this condition occurs, you should first reseat the diskette (remove the diskette from the device and reenter it into the same or a different diskette device) and perform volume initialization again. If the problem persists, try replacing the diskette with an alternate diskette, since the original diskette may be defective. If the problem continues after you have replaced the diskette, notify your Honeywell service representative.

CARTRIDGE DISK

Read/Write Errors on a Cartridge Disk

During program development, program execution terminates when uncorrectable read or write errors are encountered on a cartridge disk. An error is specified only after four retries have been unsuccessfully attempted, and control is returned to the Command Processor or the individual component.

Write Protection

A write protect switch on the cartridge disk can be set by the operator.

Defective Sectors

When a new or used cartridge disk is being initialized (using the volume initialization command in Utility Set 1) and an abnormal number of sectors (15 to 128) are detected and reported as defective, this may be due to the improper seating of the disk device head. If this condition occurs, you should first reseat the cartridge disk (remove the disk from the device and reenter it into the same or a different disk device) and perform volume initialization again. If the problem persists, try replacing the cartridge disk with an alternate disk, since the original disk may be defective. If the problem continues after you have replaced the cartridge disk, notify your Honeywell service representative.

CARD READER

Card Code Set

The only card code set supported by the system software is the American Standard Code for Information Exchange (ASCII).

Card Reader Error

When a card reader error is encountered, as indicated by the appropriate error message, the last card read, i.e., the last card in hopper, is the card in error.

PAPER TAPE READER/PUNCH (ASR-33)

Incomplete Paper Tape Operation

If a paper tape operation is interrupted due to a physical cause (for example, if the tape tears), the operation cannot be completed. In an online environment, a device time-out occurs, but the ASR remains in tape read mode.

Paper Tape Loading

When multiple modules are to be loaded from the same paper tape, it is recommended that these modules be located on one paper tape strip. Sufficient space should be left between modules on the same tape.

PART III

OPERATING PROCEDURES FOR SYSTEM PROGRAMS

SECTION 1
BOOTSTRAPPING AND LOADING SYSTEM PROGRAMS

All system programs are loaded in the same way, using the Disk Loader, Command Processor, and system console. The procedure for bootstrapping and loading a system program is described in this section. Each of the remaining sections in this part discusses the program-specific information required to load a particular program, including the file requirements and the commands to be submitted to the Command Processor before the system program is loaded.

If you are loading two or more system programs in sequence, you need perform the full bootstrapping and loading procedure only for the first program in the sequence; see "Sequential Running of System Programs" in this section for the procedures to follow under those circumstances.

The types of error and informational messages issued by system programs on the system console or another KSR-like device used as a command input device are discussed in this section. The specific messages issued by each system program are listed later in this part in the section that describes the program.

Loader halts specified in this section are described in Appendix A, "Disk Loader."

BOOTSTRAPPING AND LOADING PROCEDURE FOR SYSTEM PROGRAMS

Bootstrapping and loading a system program into memory involves:

- Invocation of the bootstrapping procedure which causes the boot loader to be loaded into memory
- Use of the boot loader to load the Disk Loader, which loads the Offline Trap Handler and the Command Processor into memory
- Submission of commands to the Command Processor that contain information to be used in loading and running the system program
- Loading of the system program by the Disk Loader, which then turns control over to the program

Bootstrapping and loading a system program, summarized in Figure III-1-1, consists of these phases (as described in the following subsections):

- Mounting the system program disk volume (if necessary)
- Performing one of two possible bootstrapping procedures
- Performing loading of the system program

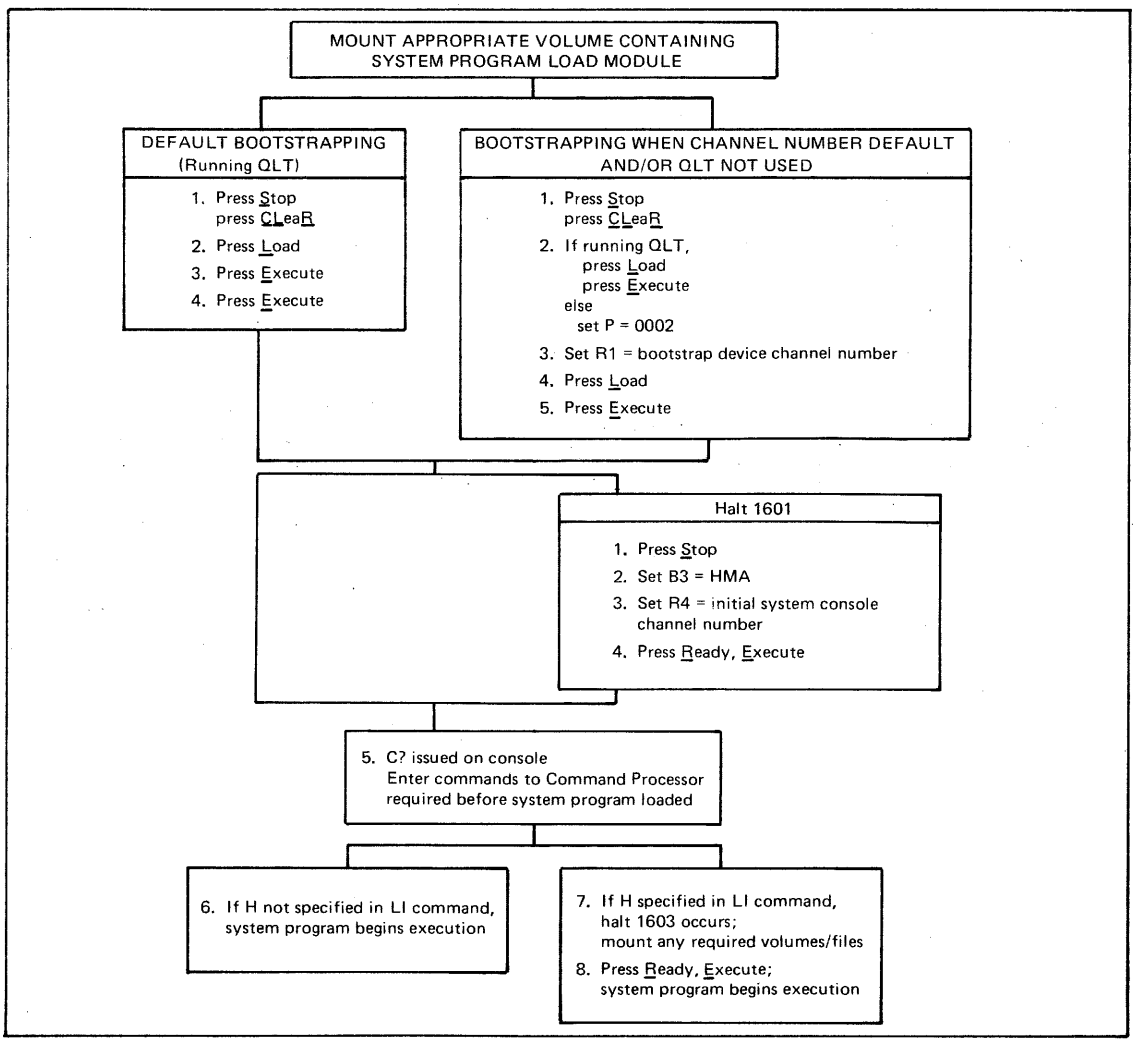


Figure III-1-1. Procedure for Bootstrapping and Loading System Program

In the bootstrapping and loading steps that follow, the values for the high memory address and the system console channel number that were set in the bootstrap record and boot loader when they were created by the Bootstrap Generator are referred to as "default" values because they can be modified through the control panel. The indicator setting for loader halt 1601 as preset in the boot loader during bootstrap generation cannot be modified during bootstrapping and loading.

Control panel operations specified in this part are described in detail under "Control Panel," Section 1 of Part II of this manual. All registers used in the following procedures are identified by selection code in Table II-1-4, "Register Selection Codes."

Mounting System Program Disk Volume

Before you bootstrap and load a system program, determine whether the disk volume containing the appropriate system program load module is mounted. This volume must include the bootstrap record, boot loader and the file that contains

the Disk Loader, the Offline Trap Handler, and the system program. If the volume is not mounted, mount it either on a device whose channel number is 0400_{16} (the default bootstrap device channel number) or on another appropriate device to be used in bootstrapping.

Bootstrapping Phase for System Programs

Control panel operation causes the bootstrap record to be read into memory from the peripheral device on which it is stored; the bootstrap record then loads the boot loader. The LOAD indicator lamp goes on when the Load key is pressed, and will remain on until the bootstrapping routine jumps to the first location of the bootstrap record.

Perform one of the following two bootstrapping procedures as described in the next two subsections:

- Default bootstrapping: Use this procedure if you wish to perform the Quality Logic Test (QLT) and use the default bootstrap device channel number (0400_{16}). (The QLT clears memory; therefore, do not perform the QLT if you need to preserve the contents of memory.)
- Bootstrapping when channel number default and/or QLT not used: Perform this procedure if you wish to change the bootstrap device channel number and/or omit the QLT.

DEFAULT BOOTSTRAPPING

The default bootstrapping procedure is as follows:

1. Perform master clear operation, automatically setting the P-register to 0000_{16}.
2. Press Load.
3. Press Execute. The QLT will be run, setting register R1 to the default bootstrap device channel number and the P-register to 0002_{16}.
4. Press Execute, causing the bootstrap record to be loaded into memory. The bootstrap record loads the boot loader, which places the default high memory address in register B3 and the default system console channel number in register R4. The boot loader halts if the indicator for loader halt 1601 was set in the boot loader when it was created by Bootstrap Generator. At loader halt 1601, proceed to step 1 of the loading phase. If no halt occurs, the boot loader automatically loads the Disk Loader as in step 5 of the loading phase.

BOOTSTRAPPING WHEN CHANNEL NUMBER DEFAULT AND/OR QLT NOT USED

The bootstrapping procedure described below permits you to use a different bootstrap device channel number and/or omit the QLT.

1. Perform master clear operation.
2. If you are running the QLT, press Load and then Execute (the QLT sets the P-register to 0002_{16}); otherwise, set the P-register to 0002_{16}.
3. Set the bootstrap record device channel number in register R1 as follows:
   a. If you are using a channel number other than the default channel number, place the desired channel number in register R1.

b. If you are using the default channel number but are not running the QLT, place the default channel number 0400_{16} in register R1. (If you are using the default channel number and are running the QLT, the QLT places the default channel number in register R1.)

4. Press Load.

5. Press Execute, causing the bootstrap record to be loaded into memory. The bootstrap record loads the boot loader, which places the default high memory address in register B3 and the default system console channel number in register R4.

   NOTE: Although the bootstrap record on a fixed cartridge disk cannot be bootstrapped directly into memory, this device can be used as the system load device by invoking the following procedure:

   At step 3a above, enter in register R1 the cartridge disk unit channel number, specifying a 1 in the low-order byte (e.g., 0401_{16}). Pressing Load and Execute causes the bootstrap record and boot loader to be read from the removable cartridge disk. (All other programs, including the Disk Loader, Offline Trap Handler, and the program specified in the removable cartridge disk boot loader, will be searched for and loaded from the fixed cartridge disk.)

The boot loader halts if the indicator for loader halt 1601 was preset in the boot loader. At loader halt 1601, proceed to step 1 of the loading phase. If no halt occurs, the boot loader automatically loads the Disk Loader, as in step 5 of the loading phase.

Loading Phase for System Programs

After you have completed one of the two alternative bootstrapping procedures described above, perform the following loading procedure:

1. At loader halt number 1601, press Stop, placing the central processor in the stop state, unless you wish to use the defaults for the high memory address and system console channel number.

2. Enter in register B3 the high memory address, unless you are using the default value.

3. Enter in register R4 the channel number of the KSR-like device you are using as the initial system console, unless you are using the default value.

4. If you have performed steps 1 through 3, press Ready, then Execute.

5. The Disk Loader is loaded into high memory. The Disk Loader loads the Offline Trap Handler, and then loads the Command Processor, which issues the system console message C?, indicating that it is ready to receive input.

   Enter commands to the Command Processor through the system console (or the specified command input device) as required to load the system program you wish to run. For the commands that can be submitted to the Command Processor before a particular system program is loaded, see the section covering program-specific information for that program later in this part. Specify the halt option (H) for the load initialize (LI) command if all required volumes or files have not been made available.

6. If you do not specify the halt option (H) in the load initialize command to the Command Processor, the loader loads the system program and turns control over to it to start program execution.

7. If you specify the halt option (H) in the load initialize command to the Command Processor, the Disk Loader loads the specified system program and then halts (halt number 1603 in register R1). Make available to the system program any other required volumes or files that are not already mounted.
8. Press Ready, then Execute. The system program will begin execution.

SEQUENTIAL RUNNING OF SYSTEM PROGRAMS

When you wish to run two or more system programs in sequence (for example, when you want to run the Editor and the Assembler one after the other), you need perform bootstrapping only once, as part of the procedure for loading the first component in the sequence.

When each system program reaches completion, it returns to the Command Processor level; the Command Processor then issues the system console message C?. When the system component returns to the Command Processor level, the attach table in memory remains intact. The only commands you need submit to the Command Processor to load the next system program are attach commands for those attachments which must be changed or added, and the load command specifying the component to be loaded. You can submit all of the commands in batch mode, if you have specified a disk or card reader as the command input device for the Command Processor (through an EX command).

NOTE: All loading of system programs must be accomplished from the same type of device.

If attach table is overlayed, when the system program returns to the Command Processor level and the system console message C? is issued, you must submit all attach commands to the Command Processor required for the next component to be loaded. (If you are using default values for logical file numbers 02 and 04, you need not specify those values.)

At completion of a system program, the program cannot return to the Command Processor level if the disk volume containing the system programs has been demounted from the device from which the system program was loaded. A loader halt (halt number 1612 in register R1) will occur at completion of the system program to permit you to remount the system program disk volume. Mount the volume, then press Ready and Execute.

MESSAGES ISSUED BY SYSTEM PROGRAMS

The system programs issue error and informational messages through the system console or another KSR-like device, if it is assigned as the command input device. All messages issued by each system program are listed in this part of the manual. To locate the messages issued by a particular system program, turn to the section in this part that describes program-specific information for that system program. Each section contains tables of messages issued by that system program.

The following types of messages can be issued:

1. Error messages printed on the system console or another KSR-like command input device in one of two formats:
   a. Coded error messages, beginning with a code number (six hexadecimal digits in parentheses), and optionally including other information on the same line
   b. Uncoded error messages, consisting of text only
2. Informational messages, always issued as text

All messages for a specific system program are grouped into two types of tables for convenient reference:

- Coded error messages, arranged numerically
- Uncoded messages (both error and informational), arranged alphabetically

Each table lists the format of each message, the condition that caused the message to be generated, and any operator action required.

The code used in coded error messages has the format

(xxyyzz)

where xx represents the number of the component reporting the error, yy is the category of the error, and zz is the number of the error within the specified category. The values that can appear in the code are described in Appendix G, "Explanation of Error Message Codes."

Abbreviations that appear in these error message tables include:

lfn - Logical file number (two decimal digits)
cccc - Channel number (four hexadecimal digits)
sswd - Software status word (four hexadecimal digits)

The software status word (ZIRSTl of an IORB) is described in the Executive and Input/Output manual.

When you are submitting commands through the system console or another KSR-like command input device in interactive mode, any error or informational message generated is issued in response to the command which has just been submitted.

SECTION 2
COMMAND PROCESSOR

The Command Processor is a system program used during the loading of other system programs and optionally offline application programs from disk. Commands can be entered to the Command Processor through the system console or a specified command input device. The Command Processor uses the command input to establish, or modify, certain control information in memory that is referred to during the loading and execution of the next program.

OPERATING PROCEDURES FOR COMMAND PROCESSOR

Initial loading of the Command Processor requires that the following steps be taken:

1. Mount the disk volume that contains the Command Processor.
2. Perform the bootstrapping and loading procedure described in Section 1 of this part. During this procedure, the Disk Loader loads the Command Processor into memory.
3. Enter commands to the Command Processor. (See Figure III-2-1 later in this section.)

After the Command Processor has been loaded into memory, it can be invoked by performing step 3.

Command Processor File Requirements

Table III-2-1 describes the files used during loading and execution of the Command Processor.

Table III-2-1. Files Used During Loading and Execution of Command Processor

File	Device Type to Which the File Can Be Assigned	Comments
Program	Disk; can be a diskette or cartridge disk	Required file. This is the file from which the Command Processor is loaded. The name of the file and the member name of the Command Processor are provided to the Disk Loader as it is loaded.
Command	KSR-like device; can be the system console or another KSR or CRT Disk; can be a diskette or cartridge disk Card Reader	Required file. The Command Processor's command input device is the system console unless a different device is specified in an EX command. If a disk or card reader is the command input device, messages from the Command Processor are issued through the system console.

Input to Command Processor

Figure III-2-1 illustrates the input commands that can be entered to the Command Processor immediately before another program is loaded. The load command must always be entered. The other commands must be entered only if appropriate information does not exist in the loader communication area and attach table after the Command Processor is loaded into memory. For detailed information regarding the elements in input commands to the Command Processor, see the Command Processor section of the Program Development Tools manual.

Command Type	Command Format
Load Initialize	LIΔ {rel[,H] / H[,rel]}
Attach	ATΔ lfn {,sdncccc [, {file-name / volume-id}] / ,=lfn$_1$}
Load	member-nameΔ[argument$_1$ [,argument$_2$] . . .]

ELEMENT	
rel	Relocation factor
H	Causes a halt to occur immediately after the program is loaded
lfn	Logical file number
sdn	Symbolic device name
cccc	Number of channel on which symbolic device located
file-name	Name of the disk file to be attached to specified logical file number
volume-id	Name of the disk volume to be attached to specified logical file number
member-name	Name of a program (a member of a partitioned file) to be loaded into memory
argument	An argument placed in memory for program use after it is loaded.

Figure III-2-1. Commands That Can Be Entered to Command Processor

MESSAGES ISSUED BY COMMAND PROCESSOR

Table III-2-2 describes the error and informational messages issued by the Command Processor. These messages are listed alphabetically. For a discussion of these messages, see "Messages Issued by System Programs" in Section 1 of this part.

Table III-2-2. Error and Informational Messages Issued by Command Processor

Message	Condition	Operator Action
CDR RD ERR	Read error on command input device.	Prepare device so it can be used and then enter GO command.
CHANNEL CONFIG TABLE FULL	Channel configuration table is full; i.e., 10 CC commands have been specified.	Specify DC command to remove specified channel configuration table entries or IC command to remove all entries and then reenter CC command.
CMD DEV NOT OPERATIONAL	Command input device is not operational.	Prepare device so it can be used and then enter GO command.
CMD FILE NOT FOUND	Designated command file cannot be found.	Reenter EX command, specifying correct file name.
CMD INPUT STREAM EXHAUSTED	The end of the command file has been reached.	No operator action required.
CMD MEMBER NOT FOUND	Specified member of the command file on the command input device cannot be found.	Reenter EX command, specifying correct member name.
DEV NOT OPERATIONAL	Device on specified channel is not operational.	Prepare device so it can be used or ignore error message and enter GO command.
DSK RD ERR	Read error on command input device.	Prepare device so it can be used and then enter GO command.
INPUT REQ TIME-OUT	Time-out has occurred on KSR-like input device.	Enter desired command.
INVALID CHANNEL NO	Requested channel number is illegal or not available in configuration.	Reenter corrected command.
INVALID COMMAND	Format of command line is illegal.	Reenter corrected command.
INVALID DEVICE FOR CMD	Invalid command input device specified.	Reenter corrected command.
INVALID HEX DIGIT	Relocation factor contains an invalid hexadecimal character.	Reenter corrected command.
INVALID KEYWORD	Character other than H was specified in load initialize (LI) command.	Reenter corrected command.
INVALID LFN	Logical file number attached is not from 00 through 99.	Reenter corrected command.
INVALID LOAD DEVICE	Specified device for load file is not same as bootstrap device.	Reenter corrected command.
INVALID REL FACTOR	Relocation factor contains more than four hexadecimal characters.	Reenter corrected command.
INVALID SDN	Symbolic device name specified is not one of the following: DSK, ASR, KSR, LPT, SPT, CDR.	Reenter corrected command.

Table III-2-2 (cont). Error and Informational Messages Issued by Command Processor

Message	Condition	Operator Action
LOAD DEV CANNOT BE DETACHED	Logical file number 00 cannot be detached.	No operator action required.
= LOGICAL FILE IS NOT ATTACHED	An attach command was specified in the format lfn,=lfn_1. The file identified by lfn_1 was not previously attached.	Enter missing attach command and then reenter attach command that caused the error.
NO CMD ENTERED	Only spaces were typed before carriage return.	Reenter command correctly.
sdn NOT ON CHANNEL	Symbolic device indicated by sdn is not on the specified channel.	Reenter corrected command.
PARAMETER ERR	Specified parameter (operand) contains an error or command exceeds eight characters.	Reenter corrected command.
PARAMETER MISSING	Required parameter (operand) missing.	Reenter corrected command.
PROGRAM NOT FOUND	Specified member to be loaded cannot be found.	Reenter corrected load command.
TOO MANY PARAMETERS	Extra parameter(s) (operand) has been specified.	Reenter corrected command.
TYPE GO TO CONTINUE	A command other than GO was entered at a WAIT or DEV NOT OPERATIONAL condition.	To continue processing, enter GO.

SECTION 3

EDITOR

OPERATING PROCEDURES FOR EDITOR

Prior to executing the Editor, the following steps must be taken:

1. Mount the disk volume that contains the Command Processor and the Editor.
2. Perform the bootstrapping and loading procedure described in Section 1 of this part, if necessary. During this procedure, the Disk Loader loads the Command Processor into memory.
3. (Optional) Mount the disk volume that contains the old master. When creating a new source module (without changing an existing source module), omit this step.
4. (Optional) Mount the disk volume that will contain the Editor output; if only a listing is desired, omit this step.
5. Make available the devices for the command and list files, if applicable.
6. Enter commands to the Command Processor (see Figure III-3-1 later in this section); this attaches the peripheral devices and causes the Editor to be loaded into memory.
7. Enter Editor initialization commands. (See "Initializing the Editor" later in this section.)

NOTE: If a required disk volume cannot be mounted because there is no empty disk drive, either (1) change disk volumes when the typeout MEMBERS? is issued or (2) designate in the load initialize command that the Editor is to halt after it is loaded into memory; when the halt occurs, demount the disk volume that contains the Editor and mount the required volume on that drive.

Editor commands are described in the Program Development Tools manual.

Editor File Requirements

Table III-3-1 describes the files used during loading and execution of the Editor. Files are assigned to devices in attach commands to the Command Processor, as described below.

Table III-3-1. Files Used During Loading and Execution of Editor

File	Device Types to Which the File Can Be Assigned	Comments
Program	Disk	Required file. The program file is the file from which the Editor is loaded. The name of this file is provided to the Disk Loader during bootstrapping. This file can be reassigned by the AT 00 command to the Command Processor. The member name EDIT must be specified in the load command to the Command Processor.
Input (Source Modules)	Disk	Required file only when altering an existing source module. Assigned by the AT 01 command to the Command Processor; modules to be edited must reside in the file specified in this command. The member name of the source module to be edited must be specified in the 01 initialization command. To return to the Command Processor so that you can designate a different file as the input file, enter the quit initialization command. (See "Initializing the Editor" later in this section.)
List (Edited Files)	Printer System Console KSR or CRT	Optional file. Assigned by AT 02 command to the Command Processor. A listing is produced when the print command is specified.
Output (Source Modules)	Disk	Required file for creating a new source module; optional if only a listing is desired. Assigned by AT 03 command to the Command Processor. The output file can reside on the same disk volume as the input file. The member name of the resulting source module must be specified in the 03 initialization command (see "Initializing the Editor" later in this section); the output member name can be the same as the input member name. There must be space on the output file for two work members; one of those members will become the final version of the edited module. Therefore, be sure that the output file contains twice as much space as the final edited module will require.
Command	System Console KSR or CRT Disk Card Reader	Required file. The system console is assumed to be the command input device unless a different device is specified in an AT 04 command to the Command Processor. If a card reader or disk device is the command input device, all messages from the Editor are issued through the system console.

Input to Command Processor Before Editor is Loaded

Before the Editor is loaded into memory, one or more commands must be entered to the Command Processor. Figure III-3-1 illustrates the input commands that can be entered to the Command Processor immediately before the Editor is loaded. The load command must always be entered. The other commands must be entered only if appropriate information does not exist in the loader communication area and attach table after the Command Processor is loaded into memory. In Figure III-3-1, the classification of commands as "required" applies only to those situations where the appropriate information does not already exist in memory. For detailed information regarding the elements in input commands to the Command Processor, see the Command Processor section of the Program Development Tools manual.

Command	Command Format	Required or Optional
Load Initialize	LI normal options	Optional
Attach (Program File)	AT 00,DSKcccc,file-name	Optional
Attach (Input File)	AT 01,DSKcccc,file-name	Optional
Attach (List File)	AT 02,sdnccc	Optional
Attach (Output File)	AT 03,DSKcccc,file-name	Optional
Attach (Command File)	AT 04,sdnccc[,file-name]	Optional
Load	EDIT[Δmember-name]	Required

NOTE: If the AT 04 command specifies a disk, you must specify the name of the member that contains the Editor commands. The member name can be one to eight alphanumeric characters.

Figure III-3-1. Input to Command Processor Before Editor is Loaded

After the load command has been entered, pressing RETURN causes the program named in the load command (EDIT) to be loaded into memory.

Initializing the Editor

After the Editor is loaded into memory, there is a typeout MEMBERS?. The Editor must then be initialized by entering initialization commands through the command input device. (If the command input device is a card reader or disk, no operator response is necessary. The initialization commands must be included at the beginning of the card deck or at the beginning of the member of the disk file that contains Editor commands.) The initialization commands are listed below; they can be entered in any order except for the GO command, which must be entered last. (The initialization commands are described in detail in the Program Development Tools manual.) If an error occurs, the typeout MEMBERS? is reissued and you must reenter the initialization command that caused the error.

01 member-name (optional); if specified, the AT 01 command must have been specified

03 member-name (optional); if specified, the AT 03 command must have been specified

INCLUDE (optional)

LL nnn (optional)

SUPPRESS (optional)

GO (required)

To terminate the current execution of the Editor, enter QT or QUIT; control is passed to the loader, which will reload the Command Processor. You can then attach different files, if desired. After the Editor is reloaded, you must reenter the initialization commands.

MESSAGES ISSUED BY EDITOR

The Editor issues coded error messages and uncoded error and informational messages; these messages are described in Tables III-3-2 and III-3-3, respectively. Coded messages are listed numerically; uncoded messages are listed alphabetically. For a discussion of these messages, see "Messages Issued by System Programs" in Section 1 of this part, and Appendix G.

Table III-3-2. Coded Error Messages Issued by Editor

Message		Condition	Operator Action
Code	Other Information		
19nnnn	cccc sswd R TO RETRY, A TO ABORT	Printer or card reader on the specified channel number is no longer operational. If the condition is corrected and R is entered, processing resumes where it stopped; otherwise, no more output is produced for the current execution of the Editor and control returns to the Command Processor.	Determine and correct the problem; i.e., be sure that the printer is not out of paper or the card reader is not out of cards. To retry, enter R. If the problem cannot be corrected, abort the current execution of the Editor by entering A.
190001	FATAL ERROR lfn	Fatal error. The typeout MEMBERS? is reissued.	Either (1) enter QT, reload the Editor, and retry, or (2) reboot the system, reload the Editor, and retry. If the error message is reissued, contact your Honeywell service representative.

Table III-3-2 (cont). Coded Error Messages Issued by Editor

Message		Condition	Operator Action
Code	Other Information		
190002	MBR NOT FND 01	Old master member name (designated in 01 command line) cannot be found. The typeout MEMBERS? is reissued.	Either (1) reenter command and designate correct member name, or (2) return control to the Command Processor by entering QT, and then attach a different file, or (3) mount a different disk volume.
190003	FATAL ERR lfn	Fatal error. The typeout MEMBERS? is reissued.	See the operator action for error code 190001.
190004	NO DATA SPACE lfn	No data space available in disk output file, so a new member cannot be created. The typeout MEMBERS? is reissued.	Enter QT, reallocate and reinitialize the disk output file by using utility commands, and then reload the Editor. (Utility commands are described in the Utility Programs manual.)
190005	NO INDX SPACE lfn	No index space available in disk file, so a new member cannot be created. The typeout MEMBERS? is reissued.	Enter QT, reallocate and reinitialize the disk file by using utility commands, and then reload the Editor. (Utility commands are described in the Utility Programs manual.)
190006	FATAL ERR lfn	Fatal error. The typeout MEMBERS? is reissued.	See the operator action for error code 190001.
19000A	FILE NOT FND lfn	Designated file cannot be found in the disk on the channel number specified. The typeout MEMBERS? is reissued.	If the wrong disk is mounted, mount the correct disk and then continue entering commands. If the correct disk is mounted but an attach command was specified incorrectly, enter QT. After the typeout C?, reenter the attach command and reload the Editor.
19000C	FILE NOT PART lfn	File is not partitioned.	Either (1) enter QT and specify a partitioned file in attach command that caused the error, or (2) reallocate the file so that it is partitioned.

Table III-3-2 (cont). Coded Error Messages Issued by Editor

Message		Condition	Operator Action
Code	Other Information		
190101	cccc sswd	Channel number designated in cccc is being used to process an outstanding request. If the system console or another KSR-like device is the command input device, there is a typeout E? to designate that an Editor command can be entered.	Either (1) enter B to restart the edit or (2) reload the Editor. If the error message is reissued, contact your Honeywell representative.
190102	cccc sswd	Invalid physical address passed by Editor to driver. cccc probably is an invalid channel number. If the system console or another KSR-like device is the command input device, there is a typeout E? to designate that an Editor command can be entered.	Either (1) enter B to restart the edit or (2) reload the Editor. If the error message is reissued, contact your Honeywell representative.
190104	cccc sswd	Invalid parameter(s) passed by Editor to driver (internal error). If the system console or another KSR-like device is the command input device, there is a typeout E? to designate that an Editor command can be entered.	Either (1) enter B to restart the edit or (2) reload the Editor. If the error message is reissued, contact your Honeywell representative.
190105	cccc sswd [R TO RETRY, A TO ABORT]	Device on channel number cccc is unavailable. If the device on cccc is a printer or card reader, the retry message is issued. If the device on cccc is a disk and the system console or another KSR-like device is the command input device, there is a typeout E? to indicate that an Editor command can be entered.	If the retry message is issued, take the appropriate action. If there is a typeout E?, check the disk on channel number cccc and then enter B to restart the edit.
190106	cccc sswd [R TO RETRY, A TO ABORT]	Time-out has occurred in the device on channel number cccc. If the device on cccc is a printer or card reader, the retry message is issued.	If the retry message is issued, take the appropriate action. If there is a typeout E?, check the disk on chanel number cccc and then enter B to restart the edit.

Table III-3-2 (cont). Coded Error Messages Issued by Editor

Message		Condition	Operator Action
Code	Other Information		
190106 (cont)		If the device on cccc is a disk and the system console or another KSR-like device is the command input device, there is a typeout E? to indicate that an Editor command can be entered.	
190107	cccc sswd [R TO RETRY, A TO ABORT]	Hardware error in device on channel number cccc. Status bits specified by sswd. If the device on cccc is a printer or card reader, the retry message is issued. If the device on cccc is a disk and the system console or another KSR-like device is the command input device, there is a typeout E? to indicate that an Editor command can be entered.	If the retry message is issued, take the appropriate action. If there is a typeout E?, check the disk on channel number cccc and then enter B to restart the edit.
190F01	TRUNC AFTER n CHARS	A line was read from the old master that contains more than n characters; i.e., more than the number of characters designated in the LL initialization command. Subsequent characters are truncated.	No operator action required.
190F02	ADJ INCLUDES	Consecutive *READ commands were not separated by at least one non-*READ command or comment line. The designated member(s) are inserted into the new master, but the *READ command line(s) are not deleted.	No operator action required.
190F03	NO COM MBR	AT 04 command assigned command file to disk, but valid member name not specified in load command. Control returns to Command Processor.	Reenter the load command, including as an argument the member name of the disk file, or detach the AT 04 command and reassign it to a KSR-like device.
190F04	ATTACH ERR lfn	No 01 or 03 initialization command was specified or an 01 or 03 initialization command was specified but the corresponding AT 01 or AT 03 attachment was not specified. The typeout MEMBERS? is reissued.	Enter an 01 or 03 initialization command or return control to the Command Processor by entering QT and then enter the AT 01 or AT 03 command.

Table III-3-3. Uncoded Error and Informational Messages Issued by Editor

Message	Condition	Operator Action
CMD ERR command	Last command entered contains an error. Command error is printed below message.	Determine the error and reenter the corrected command.
DO YOU WISH TO DELETE memname ?	The output file already contains a member that has the member name specified in the 03 initialization command.	Enter Y or N: Y - Old member is deleted and output work member is renamed so that it has the member name specified in 03 initialization command. N - Old member is not deleted and there is a typeout of the message OUTPUT MEMBER IS EDWORKO.
MEMBERS?	Request for an initialization command; i.e., 01, 03, LL, INCLUDE, SUPPRESS, GO, or QUIT.	Enter the desired initialization command(s).
OUTPUT MEMBER IS EDWORKO	Work member has the Honeywell-designated member name EDWORKO.	No operator action required. This member may be renamed by specifying the rename (RN) utility command before the next execution of the Editor.
SEARCH FAILED	A condition specified in the command could not be met within the character positions/lines searched.	No operator action required.
0 LEN - DEL	The last command entered created a line that contains no data; i.e., a line of zero length. The line is deleted.	No operator action required.

SECTION 4

MACRO PREPROCESSOR

OPERATING PROCEDURES FOR MACRO PREPROCESSOR

Before executing the Macro Preprocessor, take the following steps:

1. Mount the disk volume that contains the Macro Preprocessor and the Command Processor.
2. Perform the bootstrapping and loading procedure described in Section I of this part, if necessary. During this procedure, the Disk Loader loads the Command Processor into memory.
3. If the disk is not already mounted, mount the disk volume that contains the source module to be expanded.
4. (Optional) Mount the disk volume that contains the macro library file or files.
5. If the disk is not already mounted, mount the disk volume that will contain the Macro Preprocessor output.
6. Enter commands to the Command Processor (see Figure III-4-1 later in this section); this attaches the required devices and causes the Macro Preprocessor to be loaded into memory and executed.

NOTE: If a required disk volume cannot be mounted because there is no empty drive, designate in the load initialize command to the Command Processor that the Macro Preprocessor is to halt after it is loaded into memory. When halt number 1603 occurs, demount the disk volume that contains the Macro Preprocessor and mount the required volume on that drive.

Macro Preprocessor File Requirements

Table III-4-1 describes required and optional files and volumes used during loading and execution of the Macro Preprocessor. Files are assigned to devices in attach commands to the Command Processor, as described below.

Table III-4-1. Files Used During Loading and Execution of Macro Preprocessor

File	Device Types to Which the File Can be Assigned	Comments
Program	Disk	Required file. The program file is the file from which the Macro Preprocessor is loaded. The name of this file is provided to the Disk Loader during loading. This file can be reassigned by the AT 00 command to the Command Processor. The member name MACRO must be specified in the load command to the Command Processor.
Input	Disk	Required file. Assigned by AT 05 and optional AT 06 commands to the Command Processor. The AT 05 command is required; it assigns the file containing the source module to be expanded. The member name of the unexpanded input source module must be specified as the first argument in the load command. The AT 06 command is optional; it assigns the volume containing input macro library files. All macro library files must reside on the same disk volume. The volume name (the volume-id field in the volume label) must be specified in the AT 06 command. If no macro library files are to be used, detach any volume or file attached to logical file number 06.
Output	Disk	Required file. Assigned by AT 01 command to the Command Processor. The output file can reside on the same disk volume as the input file specified in the AT 05 command. The output file member containing the expanded assembly language source statements will have the same member base name as the unexpanded input source module specified as the first argument in the load command.

Input to Command Processor Before Macro Preprocessor is Loaded

Before the Macro Preprocessor is loaded into memory, one or more commands must be entered to the Command Processor. Figure III-4-1 illustrates the input commands that can be entered to the Command Processor immediately before the Macro Preprocessor is loaded. The load command must always be entered. The other commands must be entered only if appropriate information does not exist in the loader communication area and the attach table after the Command Processor is loaded into memory. In Figure III-4-1, the classification of commands as "required" applies only to those situations where the appropriate information does not already exist in memory.

Command Type	Command Format	Required or Optional
Load Initialize	LI normal options	Optional
Attach (Program File)	AT 00,DSKcccc,file-name	Optional
Attach (Output File)	AT 01,DSKcccc,file-name	Required
Attach (Input File)	AT 05,DSKcccc,file-name	Required
Attach (Optional Input File)	AT 06,DSKcccc,volume-id	Optional
Load	MACROΔarg_1 [,arg_2]...[a]	Required

[a] arg_1 designates the member name of the input source module to be expanded. This member name comprises one to six alphanumeric characters and must not include a suffix. Optional argumemts are listed in Table III-4-2.

Figure III-4-1. Input to Command Processor Before Macro Preprocessor is Loaded

For detailed information regarding the elements in input commands to the Command Processor, see the Command Processor section of the Program Development Tools manual.

After the load command has been entered, pressing RETURN causes the program named in the load command (MACRO) to be loaded into memory.

Table III-4-2. Macro Preprocessor Arguments

Argument	Meaning
OA	Overwrite attach table. Instructs the Macro Preprocessor to use as additional work storage the memory space reserved for the attach table and the Disk Loader; when the Macro Preprocessor reaches completion, Disk Loader halt 1604 occurs. Pressing Execute causes the Command Processor to be reloaded. Default is exit to Disk Loader, preserving file attachments.
IC	Include controls. Instructs the Macro Preprocessor to include as comment statements in the expanded source output all macro control statements, inline macro routines, macro calls, and all statements that contain error flag(s). Default is exclusion of such comments from the expanded source output.
MC	Macro calls. Instructs the Macro Preprocessor to include all macro call statements as comment statements in the expanded source output. Default is exclusion of such comments from the expanded source output.

MESSAGES ISSUED BY MACRO PREPROCESSOR

The Macro Preprocessor issues coded error messages and uncoded informational messages. Coded error messages are listed numerically and described in Table III-4-3. Uncoded informational messages are listed alphabetically in Table III-4-4. For a discussion of these messages, see "Messages Issued by System Programs" in Section 1 of this part and see Appendix G.

Table III-4-3. Coded Error Messages Issued By Macro Preprocessor

Message		Condition	Operator Action
Code	Other Information		
230001	lfn	Fatal error. Control returns to Command Processor.	1. Reload the Macro Preprocessor[a] and retry. 2. If the error message is reissued, call your Honeywell service representative.
230002	lfn	Either (1) the member name specified in a load command cannot be found, (2) a member-name specified in a LIBM statement cannot be found in the specified library file, or (3) the output member specified for the output file already exists. Control returns to Command Processor.	Use the list utility command to determine whether (1) the source member is in the attached input file, or (2) the member-names specified in the LIBM statements are in the specified library files in the input volume attached to lfn 06. If the output member name already exists, use the delete utility command to delete the member contained in the file specified in the AT 01 command. (Utility commands are described in the Utility Programs manual.)
230003	lfn	Fatal error. Control returns to Command Processor.	See operator action for error code 230001.
230004	lfn	Output member has filled available data space in partitioned disk file. Control returns to Command Processor.	Reallocate and reinitialize the disk volume by using utility commands, and then retry[a] the macro procedure. (These commands are described in the Utility Programs manual.)
230005	lfn	Output member cannot be assigned to disk because there is no index space available.	Reallocate and reinitialize the disk volume by using the utility commands and then retry[a] the macro procedure. (These commands are described in the Utility Programs manual.)
230006	lfn	Fatal error. Control returns to Command Processor.	See operator action for error code 230001.

Table III-4-3 (Cont). Coded Error Messages Issued by Macro Preprocessor

Message		Condition	Operator Action
Code	Other Information		
23000A	lfn	A file that has been attached cannot be found in the index of the designated disk volume. Control returns to Command Processor.	Obtain a listing of the files on the designated disk volume by using the list utility command. (See the Utility Programs manual.)
23000C	lfn	A file that has been attached is not partitioned. Control returns to Command Processor.	Be sure that the input and output files are partitioned. To determine whether a file is partitioned, use the list utility command. (See the Utility Programs manual.)
230101	cccc sswd	Channel number designated in cccc is being used to process an outstanding request.	Retry[a] the macro procedure.
230102	cccc sswd	Invalid channel number specified by cccc. Control returns to Command Processor.	Determine the correct channel number and retry[a] the macro procedure.
230104	cccc sswd	Invalid parameter passed by Macro Preprocessor to driver (internal error). Control returns to Command Processor.	See operator action for error code 230001.
230105	cccc sswd	Device on channel number cccc is not ready. Control returns to Command Processor.	Be sure that the door on the disk device is closed. Retry[a] the macro procedure.
230106	cccc sswd	Device time-out detected by disk driver.	Retry[a] the macro procedure. If the error message is reissued, there probably is a hardware error.
230107	cccc sswd	Hardware error in device on channel number cccc. Status bits specified by sswd.	Retry[a] the macro procedure; the hardware error may be intermittent.

Table III-4-3 (cont). Coded Error Messages Issued By Macro Preprocessor

Message		Condition	Operator Action
Code	Other Information		
230F01		Member name not designated in load command; the first (or only) argument of the load command must designate the member name of the source module to be expanded.	Retry[a] the macro procedure.
230F02		Invalid argument specified in load command.	To determine what arguments can be specified, see Table III-4-2. Retry[a] the macro procedure.
230F03		A file or volume required by the Macro Preprocessor is not attached.	To obtain a printout of the files and volumes currently attached, use the print attachments command, which is described in the Program Development Tools manual. Be sure that an input file and an output file are attached. Be sure that a macro library volume is attached if library macro routines are being referenced. Retry[a] the macro procedure.
230F05		No END statement in input member.	Specify an END statement at the end of the input member. Retry[a] the macro procedure.
230F07		All available main storage has been exhausted.	Perform one of the following: 1. Retry[a] the macro procedure, after restructuring the input program to reduce: a. the number and/or length of inline macro routines, b. the number and/or length of macro parameter and/or variable values, or c. the number of library macro routines whose names are being made available for reference. 2. If additional memory is available, allocate more memory to the Macro Preprocessor by relocating it to a lower memory address (i.e., specify a smaller relocation factor in the load initialize command) and retry[a] the macro procedure.

Table III-4-3 (cont). Coded Error Messages Issued by Macro Preprocessor

Message		Condition	Operator Action
Code	Other Information		
			3. Specify the OA argument in the load initialize command and retry[a] the macro procedure.
230F08		An ENDM statement does not appear at the end of a library input member, or at the end of an inline macro routine.	Specify an ENDM statement at the end of either the library input member or macro routine, as appropriate. Retry[a] the macro procedure.
230F09		The name of the volume mounted on the device attached to logical file number 06 is not the volume name designated in the AT 06 command. (The two volume names are compared whether or not reference is made to the library-resident macro routine(s).)	Make the necessary correction and retry[a] the macro procedure.

[a]If the argument OA was specified in the load command, reenter all of the commands to the Command Processor. Otherwise, reenter the command that caused the error and the load command.

Table III-4-4. Uncoded Informational Messages Issued by Macro Preprocessor

Message	Condition	Operator Action
ERROR(S)	At least one nonfatal error condition was detected during the macro procedure. This message is issued immediately before the Macro Preprocessor releases control of the central processor.	No operator action required.
MACRO rrrr	The Macro Preprocessor (for Release rrrr) is executing. This message is issued immediately after control of the central processor is released to the Macro Preprocessor.	No operator action required.
NO ERROR(S)	No nonfatal errors were detected during the macro procedure. This message is issued immediately before the Macro Preprocessor releases control of the central processor.	No operator action required.

SECTION 5
ASSEMBLER

OPERATING PROCEDURES FOR ASSEMBLER

Prior to executing the Assembler, the following steps must be taken:

1. Mount the disk volume that contains the Assembler and the Command Processor.
2. Perform the bootstrapping and loading procedure described in Section 1 of this part, if necessary. During this procedure, the Disk Loader loads the Command Processor into memory.
3. If the disk is not already mounted, mount the disk volume that contains the source statements to be processed.
4. (Optional) Mount the disk volume that will contain the Assembler output.
5. (Optional) Make available the device for the list file.
6. Enter commands to the Command Processor (see Figure III-5-1 later in this section); this attaches the peripheral devices and causes the Assembler to be loaded into memory and executed.

NOTE: If a required disk volume cannot be mounted because there is no empty drive, designate in the load initialize command that the Assembler is to halt after it is loaded into memory. When the halt occurs, demount the disk volume that contains the Assembler and mount the required volume on that drive.

Assembler File Requirements

Table III-5-1 describes the files used during loading and execution of the Assembler. Files are assigned to devices in attach commands to the Command Processor, as described below.

Table III-5-1. Files Used During Loading and Execution of Assembler

File	Device Types to Which the File Can be Assigned	Comments
Program	Disk	Required file. The program file is the file from which the Assembler is loaded. The name of this file is provided to the Disk Loader during bootstrapping. This file can be reassigned by the AT 00 command to the Command Processor. The member name ASM must be specified in the load command to the Command Processor.
Input (Source Module)	Disk	Required file. Assigned by AT 01 command to the Command Processor. This file contains assembly language source statements.
List (Assembler Listing)	Disk Printer	Optional file. Assigned by AT 02 command to the Command Processor. If the list file is assigned to disk, it must not be in the same file as the output file; i.e., the file name specified in the AT 02 command must not be the same as that specified in the AT 03 command. If the list file is assigned to disk, it will have the same member base name as the input source module specified as the first argument in the load command.
Output (Object Modules)	Disk	Optional file. Assigned by AT 03 command to the Command Processor. If the list file is assigned to disk, the output file must not be in the same file as the list file; i.e., the file name specified in the AT 03 command must not be the same as that specified in the AT 02 command. The output file member containing the object text will have the same member base name as the input source module specified as the first argument in the load command.

Input to Command Processor Before Assembler is Loaded

Before the Assembler is loaded into memory, one or more commands must be entered to the Command Processor. Figure III-5-1 illustrates the input commands that can be entered to the Command Processor immediately before the Assembler is loaded. The load command must always be entered. The other commands must be entered only if appropriate information does not exist in the loader communication area and attach table after the Command Processor is loaded into memory. In Figure III-5-1, the classification of commands as "required" applies only to those situations where the appropriate information does not already exist in memory. For detailed information regarding the elements in input commands to the Command Processor, see the Command Processor section of the Program Development Tools manual.

Command Type	Command Format	Required or Optional
Load Initialize	LI normal options	Optional
Attach (Program File)	AT 00,DSKcccc,file-name	Optional
Attach (Input File)	AT 01,DSKcccc,file-name	Required
Attach (List File)	AT 02,sdncccc[,file-name]	Optional
Attach (Output File)	AT 03,DSKcccc,file-name	Optional
Load	ASMΔarg_1[,arg_2]....[a]	Required

[a] arg_1 designates the member name of the source module to be assembled. This member name comprises one to six alphanumeric characters and must not include a suffix. Optional arguments are listed in Table III-5-2.

Figure III-5-1. Input to Command Processor Before Assembler is Loaded

After the load command has been entered, pressing RETURN causes the program named in the load command (ASM) to be loaded into memory.

Table III-5-2. Assembler Arguments

Argument	Meaning
LE	List error lines only.
NL	Suppress listing (overrides LE argument and LIST Assembler control statement).
NO	Suppress object text.
NOTE: Assembler arguments can be entered in any order.	

MESSAGES ISSUED BY ASSEMBLER

The Assembler issues coded error messages and an uncoded informational message. Coded error messages are listed numerically and described in Table III-5-3. The uncoded informational message is listed in Table III-5-4. For a discussion of these messages, see "Messages Issued by System Programs" in Section 1 of this part, and Appendix G.

Table III-5-3. Coded Error Messages Issued by Assembler

Message		Condition	Operator Action
Code	Other Information		
100001	lfn	Fatal error. Control returns to Command Processor.	1. Reload the Assembler and retry. 2. If the error message is reissued, call your Honeywell service representative.

Table III-5-3 (cont). Coded Error Messages Issued by Assembler

Message		Condition	Operator Action
Code	Other Information		
100002	lfn	Either (1) the member name specified in a load command cannot be found, or (2) the output member specified for the list or output file already exists. Control returns to Command Processor.	Use the list utility command to determine whether the source member is in the attached input file. (See the Utility Programs manual.) If the list or output member name already exists, use the delete utility command to delete the member contained in the file specified in the AT 02 or AT 03 command. (Utility commands are described in the Utility Programs manual.)
100003	lfn	Fatal error. Control returns to Command Processor.	See operator action for error code 100001.
100004	lfn	List member or object output member has filled available data space in partitioned disk file. Control returns to Command Processor.	Reallocate and reinitialize the disk volume by using utility commands, and then retry the assembly procedure. (Utility commands are described in the Utility Programs manual.)
100005	lfn	List member or object output member cannot be assigned to disk because there is no index space available.	Reallocate and reinitialize the disk volume by using utility commands, and then retry the assembly procedure. (Utility commands are described in the Utility Programs manual.)
100006	lfn	Fatal error. Control returns to Command Processor.	See operator action for error code 100001.

Table III-5-3 (cont). Coded Error Messages Issued by Assembler

Message		Condition	Operator Action
Code	Other Information		
10000A	lfn	A file that has been attached cannot be found in the index of the designated disk volume. Control returns to Command Processor.	Obtain a listing of the files on the designated disk volume by using the list utility command. (See the Utility Programs manual.)
10000C	lfn	A file that has been attached is not partitioned. Control returns to Command Processor.	Be sure that the input and output files, and the list file (if assigned to disk) are partitioned. To determine whether a file is partitioned, use the list utility command. (See the Utility Programs manual.)
100101	cccc sswd	Channel number designated in cccc is being used to process an outstanding request.	Retry the assembly procedure.
100102	cccc sswd	Invalid channel number specified by cccc. Control returns to Command Processor.	Determine the correct channel number and then retry the assembly procedure.
100104	cccc sswd	Invalid parameter passed by Assembler to driver (internal error). Control returns to Command Processor.	See operator action for error code 100001.
100105	cccc sswd [READY PRINTER, TYPE R TO RETRY, A TO ABORT]	Device on channel number cccc is not ready. If the device on cccc is a disk, control returns to Command Processor. If the device on cccc is a printer, the retry message is issued.	If a disk caused the error, be sure that the door on the disk device is closed. Retry the assembly procedure. If the printer caused the error, prepare the printer so that it can be used. To resume Assembler operation, enter R.

Table III-5-3 (cont). Coded Error Messages Issued by Assembler

Message		Condition	Operator Action
Code	Other Information		
100106	cccc sswd [READY PRINTER, TYPE R TO RETRY, A TO ABORT]	Device time-out detected by disk driver or printer driver. If the device on cccc is a printer, the retry message is issued.	If the printer was not ready, prepare it so it can be used. To resume Assembler execution, enter R. If a disk caused the error, retry the Assembly procedure. If the error message is reissued, there probably is a hardware error.
100107	cccc sswd	Hardware error in device on channel number cccc. Status bits specified by sswd.	Retry the assembly procedure; the hardware error may be intermittent.
100F01		Member name not designated in load command; the first (or only) argument of the load command must designate the member name of the source module to be assembled.	Retry the assembly procedure.
100F02		Invalid argument specified in load command.	To determine what arguments can be specified, see Table III-5-2. Retry the assembly procedure.
100F03		A file required by the Assembler is not attached.	To obtain a printout of the files currently attached, use the print attachments command, which is described in the Program Development Tools manual. Be sure that an input file is attached. A list file and an output file are optional, depending on the Assembler options selected. Retry the assembly procedure.

Table III-5-3 (cont). Coded Error Messages Issued by Assembler

Message		Condition	Operator Action
Code	Other Information		
100F04		Symbol table overflow.	Either (1) rewrite the source module so that some labels are temporary rather than permanent, (2) rewrite the source module so that it is several smaller modules, or (3) if additional memory is available, allocate more memory to the Assembler by relocating it to a lower memory address. The Assembler can be relocated by specifying a relocation factor in the load initialize command. Retry the assembly procedure.
100F05		No END Assembler-control statement in input member	Specify an END statement at the end of the input member. Retry the assembly procedure.
100F06		Listing and object output members assigned to the same disk file.	Retry the assembly procedure. Be sure to alter either the AT 02 or AT 03 command so that the list and object output members are in different disk files.

Table III-5-4. Uncoded Informational Message Issued by Assembler

Message	Condition	Operator Action
ASM-rrrr	The Assembler (for Release rrrr) is executing.	No operator action required.

SECTION 6
FORTRAN COMPILER

OPERATING PROCEDURES FOR FORTRAN COMPILER

Prior to executing the FORTRAN Compiler, the following steps must be taken:

1. Mount the disk volume that contains the Command Processor and the FORTRAN Compiler.
2. Perform the bootstrapping and loading procedure described in Section 1 of this part, if necessary. During this procedure, the Disk Loader loads the Command Processor into memory.
3. If the disk is not already mounted, mount the disk volume that contains the input to the compiler. If there is no empty drive on which to mount the volume, demount the volume that contains the Command Processor.
4. (Optional) Mount the disk volume that will contain compiler output.
5. Make available the device for the list file.
6. Enter commands to the Command Processor (see Figure III-6-1); this attaches the peripheral devices and causes the compiler to be loaded into memory and executed.

NOTE: If a required disk volume cannot be mounted because there is no empty drive, designate in the load initialize command that the compiler is to halt after it is loaded into memory. When the halt occurs, demount the disk volume that contains the compiler and mount the required volume on that drive.

FORTRAN Compiler File Requirements

Table III-6-1 describes the files used during loading and execution of the FORTRAN Compiler. Files are assigned in attach commands to the Command Processor, as described below.

Input to Command Processor Before FORTRAN Compiler is Loaded

Before the FORTRAN Compiler is loaded into memory, one or more commands must be entered to the Command Processor. Figure III-6-1 illustrates the input commands that can be entered to the Command Processor immediately before the compiler is loaded. The load command must always be entered. The other commands must be entered only if appropriate information does not exist in the loader communication area and attach table after the Command Processor is loaded into memory. In Figure III-6-1, the classification of commands as "required" applies only to those situations where the appropriate information does not already exist in memory. For detailed information regarding the elements in input commands to the Command Processor, see the Command Processor section of the Program Development Tools manual.

Table III-6-1. Files Used During Loading and Execution of FORTRAN Compiler

File	Device Types to Which the File Can Be Assigned	Comments
Program	Disk	Required file. The program file is the file from which the compiler is loaded. The name of this file is provided to the Disk Loader during bootstrapping. This file can be reassigned by the AT 00 command to the Command Processor. The member name FORTRAN must be specified in the load command to the Command Processor.
Input (Source Modules)	Disk	Required file. Assigned by AT 01 command to the Command Processor. A member of this file contains FORTRAN source statements.
List (Compiler Listing)	Disk Printer	Optional file. Assigned by AT 02 command to the Command Processor. If the list file is assigned to disk, it must not be in the same file as the output file; i.e., the file specified in the AT 02 command must not be the same as that specified in the AT 03 command.
Output (Object or Source Modules)	Disk	Optional file. Assigned by AT 03 command to the Command Processor. A member of this file contains object text that can be linked by the Linker, or assembly language source statements that can be processed by the Assembler. If the list file is assigned to disk, the file specified in the AT 03 command must not be the same as that specified in the AT 02 command.

Command Type	Command Format	Required or Optional
Load Initialize	LI normal options	Optional
Attach (Program File)	AT 00,PROGFILE,DSKcccc,file-name	Optional
Attach (Input File)	AT 01,DSKcccc,file-name	Required
Attach (List File)	AT 02,sdncccc[,file-name]	Optional
Attach (Output File)	AT 03,DSKcccc,file-name	Optional
Load	FORTRAN$\Delta arg_1 [arg_2]...$[a]	Required

[a] arg_1 designates the member name of the source module to be compiled. This member name must comprise one to six alphanumeric characters and not include the .F suffix. Optional arguments are listed in Table III-6-2.

Figure III-6-1. Input to Command Processor Before FORTRAN Compiler is Loaded

After the load command has been entered, pressing RETURN causes the program named in the load command (FORTRAN) to be loaded into memory.

Table III-6-2. FORTRAN Compiler Arguments

Argument	Meaning
AS	Assembler source output; default is object output.
LE	List errors only.
LO	List object output. Object text listings will be interspersed with source text listings.
NL	Suppress all listings. Default is a listing of the source module and error diagnostics.
NO	Suppress object output; need not be specified if AS is specified.
SI	Short integer and logical variables; each integer and logical variable is one word. Default is two words.
WRK=n	Object-time workspace for FORTRAN main programs; must be a 1- to 4-digit decimal number from 1 to 9999. For more detailed information, see Section 4 of the FORTRAN manual. Default is 256 words.
NOTE: FORTRAN Compiler arguments can be entered in any order.	

MESSAGES ISSUED BY FORTRAN COMPILER

The FORTRAN Compiler issues coded error messages through the system console; these messages are listed numerically and described in Table III-6-3. For a discussion of these messages, see "Messages Issued by System Programs" in Section 1 of this part, and Appendix G.

Table III-6-3. Error Messages Issued by FORTRAN Compiler

Message		Condition	Operator Action
Code	Other Information		
140001	FATAL ERR	Fatal error. Control returns to Command Processor.	1. Reload FORTRAN Compiler and retry. 2. If the error message is reissued, call your Honeywell service representative.
140002	MBR CONFLICT lfn	Input: File member name not found. Output: File member already exists. Control returns to Command Processor.	Input: Reenter the command, designating another file member name. Output: Delete the duplicate file member name by using the delete utility command described in the Utility Programs manual.
140003	FATAL ERR	Fatal error. Control returns to Command Processor.	See operator action for error code 140001.

Table III-6-3 (cont). Error Messages Issued by FORTRAN Compiler

Message		Condition	Operator Action
Code	Other Information		
140004	NO DATA SPACE lfn	No data space available in disk output file for either a new member or file. Control returns to Command Processor.	Reallocate and reinitialize the disk output file by using utility commands, and then reload the FORTRAN Compiler and retry. (Utility commands are described in the Utility Programs manual.)
140005	NO INDEX SPACE lfn	No index space available for either new members or files. Control returns to Command Processor.	Reallocate and reinitialize the disk file by using utility commands, and then reload the FORTRAN Compiler and retry. (Utility commands are described in the Utility Programs manual.)
140006	FATAL ERR	Fatal error. Control returns to Command Processor.	See operator action for error code 140001.
14000A	PART FILE NOT FND lfn	Designated partitioned file cannot be located in the disk on the channel number specified. Control returns to Command Processor.	Correct the file or mount correct disk volume and reload the FORTRAN Compiler.
14000C	FILE NOT PART lfn	File is not partitioned.	Either (1) specify a partitioned file in attach command that caused error, or (2) reallocate the file so that it is partitioned.
140101	cccc sswd	Channel whose number is designated in cccc is being used to process an outstanding request. Control returns to Command Processor.	Check validity of attach table entries. Reload the FORTRAN Compiler. (The attach table is described in Planning and Building an Online Application manual.)
140102	cccc sswd	Invalid physical address passed by component to driver. cccc is probably an invalid channel number. Control returns to Command Processor.	Check validity of attach table entries. Reload the FORTRAN Compiler. (The attach table is designated in Planning and Building Online Application manual.)
140104	cccc sswd	Invalid parameter passed by component to driver (internal error). Control returns to Command Processor.	Check validity of attach table entries. Reload the FORTRAN Compiler. (The attach table is described in Planning and Building an Online Application manual.)

Table III-6-3 (cont). Error Messages Issued by FORTRAN Compiler

Message		Condition	Operator Action
Code	Other Information		
140105	cccc sswd	Device specified by cccc is unavailable. If the device is a printer, a halt occurs; otherwise, control returns to Command Processor.	Check validity of attach table entries. If a printer is used, press Execute; otherwise, reload the FORTRAN Compiler. (The attach table is described in Planning and Building an Online Application manual.)
140106	cccc sswd	Time-out has occurred in the device on channel number cccc. If the device is a printer, a halt occurs; otherwise, control returns to Command Processor.	Check validity of attach table entries. If a printer is used, press Execute; otherwise, reload the FORTRAN Compiler. (The attach table is described in Planning and Building an Online Application manual.)
140107	cccc sswd	Hardware error in device on channel number cccc. Status bits specified by sswd. Control returns to Command Processor.	Check validity of attach table entries. Reload the FORTRAN Compiler. (The attach table is described in Planning and Building an Online Application manual.)
140F01		Member name not designated in load command line. Control returns to Command Processor.	Reload the FORTRAN Compiler, specifying the member name in the load command line.
140F02		Invalid argument specified in load command line. Control returns to Command Processor.	Reload the FORTRAN Compiler, specifying a correct argument in the load command line.
140F03		A file required by the compiler is not attached. Control returns to Command Processor.	Attach the required file by specifying the appropriate attach command. Reload the FORTRAN Compiler.
140F04		Insufficient memory (i.e., table overflow) for Compiler operation.	Reboot and reload FORTRAN Compiler, specifying more memory for use by FORTRAN Compiler.
140F05		Fatal compilation. Control returns to Command Processor.	Correct source code, reload FORTRAN Compiler, and recompile.

SECTION 7
COBOL COMPILER

OPERATING PROCEDURES FOR COBOL COMPILER

Prior to executing the COBOL Compiler, the following steps must be taken:

1. Prepare the disk devices for the compiler work files and the optional object output file. The compiler work files are relative files; the object output file is a partitioned file and must be formatted for partitioned access.

   When allocating space on a disk, the amount of space required is determined by the size and content of the COBOL program. An average COBOL program requires the following disk space for each 100 source lines:

   COBWRK - 50 sectors
   LABELS - 2 sectors
   OBJECT OUTPUT - 25 sectors

2. Mount the disk volume that contains the Command Processor and the COBOL Compiler. Depending on the amount of disk space available, this volume may also contain other files required by the compiler.
3. Perform the bootstrapping and loading procedure described in Section 1 of this part, if necessary. During this procedure, the Disk Loader loads the Command Processor into memory.
4. If the disk is not already mounted, mount the disk volume that contains the input to the compiler and other files required for compilation.
5. (Optional) Make available the device for the list file.
6. Enter commands to the Command Processor (see Figure III-7-1); this attaches the peripheral devices and causes the compiler to be loaded into memory and executed.

COBOL Compiler File Requirements

Table III-7-1 describes the files used during loading and execution of the COBOL Compiler. Files are assigned in attach commands to the Command Processor, as described below.

Input to Command Processor Before COBOL Compiler is Loaded

Before the COBOL Compiler is loaded into memory, one or more commands must be entered to the Command Processor. Figure III-7-1 illustrates the input commands that can be entered to the Command Processor immediately before the compiler is loaded. The load command must always be entered. The other commands must be entered only if appropriate information does not exist in the

loader communication area and attach table after the Command Processor is loaded into memory. In Figure III-7-1, the classification of commands as "required" applies only to those situations where the appropriate information does not already exist in memory. For detailed information regarding the elements in input commands to the Command Processor, see the Command Processor section of the Program Development Tools manual.

Table III-7-1. Files Used During Loading and Execution of COBOL Compiler

File	Device Types to Which the File Can Be Assigned	Comments
Program	Disk	Required file. The program file is the file from which the compiler is loaded. The name of this file is provided to the Disk Loader during bootstrapping. This file cannot be reassigned by the AT 00 command to the Command Processor. The member name COBOL must be specified in the load command to the Command Processor.
Input (Source Module)	Disk	Required file. Assigned by AT 01 command to the Command Processor. This file contains COBOL source statements.
List (Compiler Listing)	Disk Printer	Optional file. Assigned by AT 02 command to the Command Processor. If the list file is assigned to disk, it must be a member of a partitioned file and it must not be in the same file as the output file; i.e., the file specified in the AT 02 command must not be the same as that specified in the AT 03 command.
Output (Object Modules)	Disk	Optional file. Assigned by AT 03 command to the Command Processor. This file contains object modules that can be linked by the Linker. If the list file is assigned to disk, the file specified in the AT 03 command must not be the same as that specified in the AT 02 command.

Table III-7-1 (cont). Files Used During Loading and Execution of COBOL Compiler

File	Device Types to Which the File Can Be Assigned	Comments
COBWRK (Compiler Work File)	Disk	Required file. Assigned by AT 06 command to the Command Processor. This relative file contains the intermediate representation of the COBOL source program that is used during the compilation procedure.
LABELS (Compiler Work File)	Disk	Required file. Assigned by AT 05 command to the Command Processor. This relative file contains paragraph names used by several compiler phases.

Command Type	Command Format	Required or Optional
Load Initialize	LI normal options	Optional
Attach (Input File)	AT 01,DSKcccc,file-name	Required
Attach (List File)	AT 02,sdncccc[,file-name]	Optional
Attach (Output File)	AT 03,DSKcccc,file-name	Optional
Attach (LABELS File)	AT 05,DSKcccc,file-name	Required
Attach (COBWRK File)	AT 06,DSKcccc,file-name	Required
Load	COBOLΔarg$_1$ [arg$_2$] . . .[a]	Required

[a] arg$_1$ designates the member name of the source module to be compiled. This member name must comprise one to six alphanumeric characters and not include a suffix. Optional arguments are listed in Table III-7-2.

Figure III-7-1. Input to Command Processor Before COBOL Compiler is Loaded

After the load command has been entered, pressing RETURN causes the program named in the load command (COBOL) to be loaded into memory.

Table III-7-2. COBOL Compiler Arguments

Argument	Meaning
LE	List errors only.
LD	List data map, source text, and errors.
LO	List object text, data map, source text, and errors.
NL	Suppress all listings. Default is a listing of the source module and error diagnostics. In either case, there is a console typeout of the number of errors found.
NO	Suppress object output.
DB	Compile debugging lines as comments, ignoring the WITH DEBUGGING MODE clause.
NOTE: A load command may contain a maximum of four arguments, including the first argument, which indicates the member name of the source module to be compiled; additional arguments can be entered in any order. Only one of the following arguments may be entered: LE, LD, LO, NL.	

MESSAGES ISSUED BY COBOL COMPILER

The COBOL Compiler issues coded error messages and uncoded error and information messages through the system console; these messages are described in Tables III-7-3 and III-7-4, respectively. Coded messages are listed numerically; uncoded messages are listed alphabetically. For a discussion of these messages, see "Messages Issued by System Programs" in Section 1 of this part, and Appendix G.

Table III-7-3. Coded Error Messages Issued By COBOL Compiler

Message		Condition	Operator Action
Code	Other Information		
260001	lfn	Fatal error. Control returns to Command Processor.	1. Reload the COBOL Compiler and retry. 2. If the error message is reissued, call your Honeywell service representative.

Table III-7-3 (cont). Coded Error Messages Issued By COBOL Compiler

Message		Condition	Operator Action
Code	Other Information		
260002	lfn	Either (1) the member name specified in a load command cannot be found, or (2) the output member specified for the list or output file already exists. Control returns to Command Processor.	Use the list utility command to determine whether the source member is in the attached input file. If the list or output member name already exists, use the delete utility command to delete the member contained in the file specified in the AT 02 or AT 03 command. (Utility commands are described in the Utility Programs manual.)
260003	lfn	Fatal error. Control returns to Command Processor.	See operator action for error code 260001.
260004	lfn	List member or object output member has filled available data space in partitioned disk file. Control returns to Command Processor.	Reallocate and reinitialize the disk volume by using utility commands and then retry the compilation procedure. (Utility commands are described in the Utility Programs manual.)
260005	lfn	List member or object output member cannot be assigned to disk because there is no index space available.	Reallocate and reinitialize the disk volume by using utility commands, and then retry the compilation procedure. (Utility commands are described in the Utility Programs manual.)
260006	lfn	Fatal error. Control returns to Command Processor.	See operator action for error code 260001.
26000A	lfn	A partitioned file that has been attached cannot be found in the index of the designated disk volume. Control returns to Command Processor.	Obtain a listing of the files on the designated disk volume by using the list utility command. (See the Utility Programs manual.)
26000C	lfn	A file that has been attached is not partitioned. Control returns to Command Processor.	Be sure that the input and output files, and the list file (if assigned to disk) are partitioned. To determine whether a file is partitioned, use the list utility command. (See the Utility Programs manual.)
260101	lfn	Channel number designated for file attached to lfn is being used to process an outstanding request.	Retry the compilation procedure.

Table III-7-3 (cont). Coded Error Messages Issued By COBOL Compiler

Message		Condition	Operator Action
Code	Other Information		
260102	lfn	Invalid channel number specified for file attached to lfn. Control returns to Command Processor.	Retry the compilation procedure.
260104	lfn	Invalid parameter passed by compiler to driver (internal error). Control returns to Command Processor.	See operator action for error code 260001.
260105	lfn	Device to which lfn is attached is not ready. If the device is a disk, control returns to Command Processor. If the device is a printer, a halt occurs, followed by the message: R TO RETRY, A TO ABORT.	See operator action for error code 260106.
260106	lfn	Device time-out detected by disk driver or printer driver. If the device attached to lfn is a printer, a halt occurs, followed by the message: R TO RETRY, A TO ABORT.	If the attached device is a printer, prepare it for use and then type R (followed by return carriage) to continue, or type A (followed by return carriage) to return to command level. If the device is a disk, then a hardware failure has occurred; in this case, type A to return to command level.
260107	lfn	Hardware error on device attached to lfn.	Retry the compilation procedure; the hardware error may be intermittent.

Table III-7-4. Uncoded Error and Informational Messages Issued By COBOL Compiler

Message	Condition	Operator Action
BES2 COBOL VERSION nn	An informational message giving the numbered (nn) version of the compiler.	No operator action required.
INVALID ARG IGNORED	Invalid argument has been specified in the load command.	No operator action required.

Table III-7-4 (cont). Uncoded Error and Informational Messages Issued By COBOL Compiler

Message	Condition	Operator Action
lfn NOT ATTACHED	A file required by the compiler is not attached. lfn may have values 01, 02, 03, 05, or 06.	Attach the required file by entering the appropriate attach command and then reenter the load command.
NO ARGS SUPPLIED	No arguments were specified in the load command.	Reenter the load command, specifying the correct arguments.
NO CODE GENERATED	Fatal errors occurred during compilation; no object output is produced.	The results of compilation cannot be linked and executed.
nnnn ERRORS	A count of source program errors detected by the compiler is printed at the end of compilation.	No operator action required.
SOURCE NAME MAX IS 6 BYTES	The first argument in the load command is incorrect. It must be the name of the source member attached to lfn 01 and it must not exceed six characters.	Reenter the load command with the correct arguments.

SECTION 8
LINKER

OPERATING PROCEDURES FOR LINKER

Prior to executing the Linker, the following steps must be taken:

1. Mount the disk volume that contains the Command Processor and the Linker.
2. Perform the bootstrapping and loading procedure described in Section 1 of this part, if necessary. During this procedure, the Disk Loader loads the Command Processor into memory.
3. If the disk volume(s) are not already mounted, mount the disk volume(s) that contain some or all of the object modules to be linked. If there are not enough disk drives, demount a disk volume after it is used so that another volume can be mounted on that drive.
4. (Optional) Mount the disk volume(s) that will contain the Linker output. Both the object modules to be linked and the resulting load module(s) can be on the same disk volume.
5. Make available the devices for the command and list files, if applicable.
6. Enter commands to the Command Processor (see Figure III-8-1); this causes the peripheral devices to be attached and the Linker to be loaded into memory.

NOTE: If a required disk volume cannot be mounted because there is no empty drive, designate in the load initialize command that the Linker is to halt after it is loaded into memory. When the halt occurs, demount the disk volume that contains the Linker and mount the required volume on that drive. If there still are not enough drives, enter the appropriate Linker command (i.e., IN or OUT), demount a disk volume after it is used by the Linker, and use that drive.

Linker commands are described in the Program Development Tools manual.

Linker File Requirements

Table III-8-1 describes the files used during loading and execution of the Linker. Files are assigned to devices in attach commands to the Command Processor as described below.

Input to Command Processor Before Linker is Loaded

Before the Linker is loaded into memory, one or more commands must be entered to the Command Processor. Figure III-8-1 illustrates the input commands that can be entered to the Command Processor immediately before the Linker is loaded. The load command must always be entered. The other commands must be entered only if appropriate information does not exist in the loader communication area and attach table after the Command Processor is loaded into memory. In Figure III-8-1, the classification of commands as "required" applies only to those situations where the appropriate information does not already exist in memory. For detailed information regarding the elements in input commands to the Command Processor, see the Command Processor section of the Program Development Tools manual.

Table III-8-1. Files Used During Loading and Execution of Linker

File	Device Types to Which the File Can be Assigned	Comments
Program	Disk	Required file. The program file is the file from which the Linker is loaded. The name of this file is provided to the Disk Loader during bootstrapping. This file can be reassigned by the AT 00 command to the Command Processor. The member name LINKER must be specified in the load command to the Command Processor.
Input (Object Modules)	Disk	Required file. Assigned by AT 01 and AT 05 commands to the Command Processor. The AT 01 command is required; the other is optional. Each file can reside on the same disk volume or on different volumes. An IN command can be specified during execution of the Linker to change the name of any previously attached input file. The names of specific object modules to be linked are specified in LINKNΔ object-module and LINKΔ object-module commands to the Linker.
List (Link Maps)	System Console KSR or CRT Printer Disk	Optional file. Assigned by the AT 02 command to the Command Processor. If the list file is assigned to disk, use the print utility (PT) command to obtain a copy of the contents of the list file.
Output (Load Modules)	Disk	Optional file. Assigned by the AT 03 command to the Command Processor. The input and output files can be on the same disk volume. An OUT command to the Linker can be used to change the name of the output file.

Table III-8-1 (cont). Files Used During Loading and Execution of Linker

File	Device Types to Which the File Can be Assigned	Comments
Command	System Console KSR or CRT Disk Card Reader	The name of each load module that is not an overlay is declared in a separate NAME command to the Linker. The name of each load module that is an overlay is declared in a separate OVLY command to the Linker. Required file. The system console is assumed to be the command input device unless a different device is specified in an AT 04 command to the Command Processor. If a disk device or card reader is the command input device, messages from the Linker are issued through the system console. If a disk is the command input device, the member name of the command file must be designated in the load command.

Command Type	Command Format	Optional or Required
Load Initialize	LI normal options	Optional
Attach (Program File)	AT 00,DSKcccc,file-name	Optional
Attach (Input File)	AT 01,DSKcccc,file-name	Required
Attach (List File)	AT 02,sdncccc[,file-name]	Optional
Attach (Output File)	AT 03,DSKcccc,file-name	Optional
Attach (Command File)	AT 04,sdncccc[,file-name]	Optional
Attach (Optional Input File)	AT 05,DSKcccc,file-name	Optional
Load	LINKER[Δmember-name]	Required

NOTE: If the AT 04 command specifies a disk file, you must specify the name of the member that contains the Linker commands. The member name can be one to six characters and must not include a suffix.

Figure III-8-1. Input to Command Processor Before Linker is Loaded

After the load command has been entered, pressing RETURN causes the program named in the load command (LINKER) to be loaded into memory.

MESSAGES ISSUED BY LINKER

The Linker issues coded error messages and uncoded error and informational messages; these messages are described in Tables III-8-2 and III-8-3, respectively. Coded messages are listed numerically; uncoded messages are listed alphabetically. For a discussion of these messages, see "Messages Issued by System Programs" in Section 1 of this part, and Appendix G.

Table III-8-2. Coded Error Messages Issued by Linker

Message		Condition	Operator Action
Code	Other Information		
110001	FATAL ERROR	Fatal error.	Either (1) reload the Linker, and retry, or (2) reboot the system, reload the Linker, and retry. If error message is reissued, contact your Honeywell service representative.
110003	FATAL ERROR	Fatal error.	See operator action for error code 110001.
110004		No data space on output file. Linking continues, but no load module(s) are created.	Reallocate and reinitialize space in the output file by using utility commands, and then retry the linking procedure. (Utility Commands are described in the Utility Programs manual.)
110005		No index space available. Linking continues, but no load module(s) are created.	If additional load modules are desired, reallocate and reinitialize the file and retry the linking procedure.
110006	FATAL ERROR	Fatal error.	See operator action for error code 110001.
11000A	FILE NOT FOUND lfn	Designated file cannot be found in the disk on the channel number specified.	If the wrong disk is mounted, mount the correct disk and then continue entering commands. If the correct disk is mounted but an attach command was specified incorrectly, enter QT. After the typeout C?, reenter the attach command and reload the Linker.
11000C	FILE NOT PART lfn	File is not partitioned.	Either (1) enter QT and specify a partitioned file in attach command that caused error, or (2) reallocate the file so that it is partitioned.
110101	cccc sswd	Channel number designated in cccc is being used to process an outstanding request. Control returns to Command Processor.	Retry the linking procedure.
110102	cccc sswd	Invalid physical address passed by Linker to driver. cccc probably is an invalid channel number. Control returns to Command Processor.	Retry the linking procedure.

Table III-8-2 (cont). Coded Error Messages Issued by Linker

Message		Condition	Operator Action
Code	Other Information		
110104	cccc sswd	Invalid parameter(s) passed by Linker to driver (internal error). Control returns to Command Processor.	Retry the linking procedure.
110105	cccc sswd	Device on channel number cccc is unavailable. If the device on cccc is a printer, the READY PRINTER message is issued; otherwise, control returns to Command Processor.	If the device on cccc is a printer, prepare the printer so that it can be used and then take the appropriate action. If there is a console typeout C?, retry the linking procedure.
110106	cccc sswd	Time-out has occurred in the device on channel number cccc. If a printer is on cccc, the READY PRINTER message is issued; otherwise, control returns to Command Processor.	If the device on cccc is a printer, prepare the printer so that it can be used and then take the appropriate action. If there is a console typeout C?, retry the linking procedure.
110107	cccc sswd	Hardware error in device on channel number cccc. Status bits are specified by sswd. If the device on cccc is a printer, the READY PRINTER message is issued; otherwise, control returns to Command Processor.	If the device on cccc is a printer, prepare the printer so that it can be used and then take the appropriate action. If there is a console typeout C?, retry the linking procedure.

Table III-8-3. Uncoded Error and Informational Messages Issued By Linker

Message	Condition	Operator Action
ATTACH ERRΔ {01 / 05}	Either no file attached to 01, or one of the files attached to 01 or 05 not found. Control returns to Command Processor.	Make the necessary corrections and retry the link process.
ATTACH ERR 03, NO LINK-OUTPUT WILL BE CREATED	Either no file attached to 03, or file specified not found. Linking continues, but no load module(s) are created.	No operator action required.

Table III-8-3 (cont). Uncoded Error and Informational Messages Issued By Linker

Message	Condition	Operator Action
CMD ERR	Command error. If the command file is a disk or an imbedded command caused the error, the error message is preceded by a typeout of the command that caused the error; if multiple commands were on the line, the entire line is typed. If there are multiple commands on the command line in which the error occurred, preceding commands are processed, but the command that caused the error and subsequent commands are ignored.	If the command file is a disk or an imbedded command caused the error, no operator action is possible. If there are multiple commands on the command line, determine the cause of the error and re-enter the corrected command.
DISK LFN 02 IO ERR, NO LIST FILE	List file will not be created by Linker. Preceding error code designates cause of problem. Linking continues, but no maps can be created.	Look up the meaning of the error code in Table III-8-2.
FL file-name NOT FND	File specified in IN or OUT command not found. Linking continues, but the previously attached file is used.	No operator action required.
HMA mmmm EXCEEDED	Specified or default high memory address exceeded by load module.	No operator action required.
IMG UNDEF	Undefined symbols existed when the load module was closed. Special records that identify the undefined symbols are generated as part of the load module.	No operator action required.
IN IO ERR (preceded by coded error message)	Input file cannot be read. Error code indicates cause of problem. Control returns to Command Processor.	Look up the meaning of the error code in Table III-8-2 and perform the specified operator action.

Table III-8-3 (cont). Uncoded Error and Informational Messages Issued by Linker

Message	Condition	Operator Action
INV OBJ (preceded by first 15 characters of invalid record)	Assembler or compiler output module contains an invalid record. Control returns to Command Processor.	Reassemble/recompile module and retry linking process.
IST CMD ERR	External symbol name specified in IST command was not defined during linking procedure. The load module will not contain an initialization start address.	No operator action required.
LFN (02), (03) SAME FL. LIST FL NOT OPENED	List and output files, designated by LFN 02 and LFN 03, respectively, were assigned to the same file. Only the output file has been opened. No maps can be produced.	No operator action required. If maps are desired, respecify the AT 02 or AT 03 command and the load command.
MNT lfn FL file-name	An IN or OUT command was specified. A halt occurs. Linker is ready to process new file.	Mount disk volume that contains file designated in file-name. To continue, press Execute.
NO COMMAND FILE	Invalid device specified as command file in AT 04 command. Control returns to Command Processor.	Reenter corrected AT 04 command.
NO IMAGE (preceded by message 110004 or 110005)	Look up the meaning of error message 110004/ 110005 in Table III-8-2. Linking continues, but no load module(s) are created.	No operator action required.
NO LIST FILE-NO MAPS	AT 02 command was not specified; i.e., no list file was assigned.	No operator action required.
NO NAME-NO IMAGE	NAME command was not specified. Linking continues, but no load module will be created.	To create a load module, specify an END statement and then specify the NAME command and reenter the Linker command(s) that designate which modules will be linked.

Table III-8-3 (cont). Uncoded Error and Informational Messages Issued by Linker

Message	Condition	Operator Action
NO NAME-NO LIST FILE ON DISK (followed by NO NAME-NO IMAGE message)	No name specified, so list file will not be on disk.	No operator action required.
object module name NOT FND	Object module requested in command cannot be found.	If STEP mode is in effect, determine the correct object module name and reenter the command.
READY PRINTER, REPLY R TO RETRY, A TO ABORT, C TO CONTINUE WITHOUT LIST FL (preceded by coded error message)	Printer is not operational.	Take the appropriate action.
TBL OV	Symbol table overflow. Control returns to Command Processor.	Allocate more memory to the Linker, or purge symbols that are no longer required.
UNDEF OFFSET	External reference with offset, to a symbol not yet defined, was encountered in an object module being linked. Control returns to Command Processor.	Link the modules in a different order, or correct the source text and reassemble or recompile the object module.

SECTION 9

CROSS-REFERENCE PROGRAM

OPERATING PROCEDURES FOR CROSS-REFERENCE PROGRAM

Prior to executing the Cross-Reference Program, the following steps must be taken:

1. Mount the disk volume that contains the Command Processor and the Cross-Reference Program.
2. Perform the bootstrapping and loading procedure described in Section 1 of this part, if necessary. During this procedure, the Disk Loader loads the Command Processor into memory.
3. If the disk is not already mounted, mount the disk volume that contains the source module(s) to be processed.
4. Make available the device for the list file, if applicable.
5. Enter commands to the Command Processor (see Figure III-9-1); this attaches the peripheral devices and causes the Cross-Reference Program to be loaded into memory.

Cross-Reference Program File Requirements

Table III-9-1 describes the files used during loading and execution of the Cross-Reference Program. Files are assigned to devices in attach commands to the Command Processor, as described below.

Input to Command Processor Before Cross-Reference Program is Loaded

Before the Cross-Reference Program is loaded into memory, one or more commands must be entered to the Command Processor. Figure III-9-1 illustrates the input commands that can be entered to the Command Processor immediately before the Cross-Reference Program is loaded. The load command must always be entered. The other commands must be entered only if appropriate information does not exist in the loader communication area and attach table after the Command Processor is loaded into memory. In Figure III-9-1, the classification of commands as "required" applies only to those situations where the appropriate information does not already exist in memory. For detailed information regarding the elements in input commands to the Command Processor, see the Command Processor section in the Program Development Tools manual.

Table III-9-1. Files Used During Loading and Execution of Cross-Reference Program

File	Device Types to Which the File Can Be Assigned	Comments
Program	Disk	Required file. The program file is the file from which the Cross-Reference Program is loaded. The name of this file is provided to the Disk Loader during bootstrapping. This file can be reassigned by the AT 00 command to the Command Processor. The member name XREF must be specified in the load command to the Command Processor.
Input (Source Modules)	Disk	Required file. Assigned by AT 01 command to the Command Processor. Assembly language source modules to be processed by the Cross-Reference Program must reside on this file.
List (Cross-Reference Listing)	Printer System Console KSR or CRT	Required file. Assigned by AT 02 command to the Command Processor.

Command Type	Command Format	Required or Optional
Attach (Program File)	AT 00,DSKcccc,file-name	Optional
Attach (Input File)	AT 01,DSKcccc,file-name	Required
Attach (List File)	AT 02,sdncccc	Required
Load	XREF△member-name	Required

NOTE: In the load command you must specify the member name of the assembly language source module that will be read by the Cross-Reference Program. The member name must comprise one through eight alphanumeric characters.

Figure III-9-1. Input to Command Processor Before Cross-Reference Program is Loaded

After the load command has been entered, pressing RETURN causes the program named in the load command (XREF) to be loaded into memory.

MESSAGES ISSUED BY CROSS-REFERENCE PROGRAM

The Cross-Reference Program issues coded error messages; these messages are listed numerically and described in Table III-9-2. For a discussion of these messages, see "Messages Issued by System Programs" in Section 1 of this part, and Appendix G.

Table III-9-2. Coded Error Messages Issued by Cross-Reference Program

Message		Condition	Operator Action
Code	Other Information		
18nnnn	cccc sswd R TO RETRY, A TO ABORT	Printer on the specified channel is no longer operational. If the condition is corrected and R is entered, processing resumes where it stopped; if A is entered, no more output is produced for the current execution of the Cross-Reference Program and control returns to the Command Processor.	Determine and correct the problem; i.e., be sure that the printer is not out of paper. To retry, enter R. If the problem cannot be corrected, abort the current execution of the Cross-Reference Program by entering A.
180001	FATAL ERR lfn	Fatal error. Control returns to Command Processor.	Either (1) reload the Cross-Reference Program, and retry, or (2) reboot the system, reload the Cross-Reference Program, and retry. If error message is reissued, contact your Honeywell service representative.
180002	MBR NOT FND 01	Member not found in 01 (the attached input file). Control returns to Command Processor.	Reenter the AT 01 command and specify the correct member name.
180003	FATAL ERR lfn	Fatal error. Control returns to Command Processor.	See operator action for error code 180001.
180006	FATAL ERR lfn	Fatal error. Control returns to Command Processor.	See operator action for error code 180001.
18000A	FILE NOT FND 01	The designated file cannot be found in 01, or 01 is not partitioned. Control returns to Command Processor.	If the wrong disk is mounted, mount the correct disk and then enter the remainder of the commands that invoke the Cross-Reference Program. If the correct disk is mounted but the file is not partitioned, partition the file by using the initialize file function, which is described in the Utility Programs manual. If the correct disk is mounted but the AT 01 command was specified incorrectly, reenter the AT 01 command and reload the Cross-Reference Program.

Table III-9-2 (cont). Coded Error Messages Issued by Cross-Reference Program

Message		Condition	Operator Action
Code	Other Information		
18000C	FILE NOT PART lfn	The file found is not partitioned. Control returns to Command Processor.	Either (1) specify a partitioned file in attach command that caused error, or (2) reallocate the file so that it is partitioned.
180101	cccc sswd	Channel number designated in cccc is being used to process an outstanding request. Control returns to Command Processor.	May be an intermittent hardware error; to retry, reload the Cross-Reference Program.
180102	cccc sswd	Invalid physical address submitted by component to driver. cccc is probably an invalid channel number. Control returns to Command Processor.	May be an intermittent hardware error; reload the Cross-Reference Program.
180104	cccc sswd	Invalid parameter(s) passed by component to driver (internal error). Control returns to Command Processor.	May be an intermittent hardware error. Reload the Cross-Reference Program.
180105	cccc sswd	Device on channel number cccc is unavailable. If the device on cccc is a printer, a halt occurs. Otherwise, control returns to Command Processor.	If a halt occurs, press Execute to retry. After the typeout C?, reload the Cross-Reference Program.
180106	cccc sswd	Time-out has occurred in the device on channel number cccc. If the device on cccc is a printer, a halt occurs; otherwise, control returns to Command Processor.	If a halt occurs, press Execute to retry. After the typeout C?, reload the Cross-Reference Program.
180107	cccc sswd	Hardware error in device on channel number cccc. Status bits are specified by sswd. If the device on cccc is a printer, a halt occurs; otherwise, control returns to Command Processor.	If a halt occurs, press Execute to retry. After the typeout C?, reload the Cross-Reference Program.
180F01	TRUNC AFTER n CHARS	A line was read that contains more than 80 characters. Only 80 characters were read; subsequent characters were truncated.	No operator action required.

Table III-9-2 (cont). Coded Error Messages Issued by Cross-Reference Program

Message		Condition	Operator Action
Code	Other Information		
180F02	MEMORY FULL	There is no available space in memory. There is a typeout on the system console of the contents of memory that have already been processed by the Cross-Reference Program and then control returns to Command Processor.	No operator action required.
180F03	NO INPUT	Load command did not include valid member name of source module to be processed by Cross-Reference Program. Control returns to Command Processor.	Reenter load command, including member name of source module to be processed.

SECTION 10
UTILITY SETS 1, 2, AND 3

Utility Sets 1, 2, and 3 are three utility programs, each consisting of a number of utility functions, that are described in the Utility Programs manual.

OPERATING PROCEDURES FOR UTILITY SETS 1, 2, AND 3

Before you load any one of the three utility programs, take the following steps:

1. Mount the disk volume containing the Command Processor and the utility program, if the volume is not already mounted.
2. Perform the bootstrapping and loading procedure for system programs described in Section 1 of this part, if necessary. During this procedure, the Disk Loader loads the Command Processor into memory.
3. Make available any volumes/files required for the particular utility functions to be performed (see "Volume/File Requirements for Utility Sets 1, 2, and 3" below). Specify these volumes/files in attach commands to the Command Processor.
4. Enter the commands to the Command Processor required before the utility program is loaded into memory (see "Input to Command Processor Before Utility Set 1, 2, or 3 is Loaded" below); this attaches the peripherals and causes the utility program to be loaded into memory.

During execution of a utility program, submit commands to the program through the specified command input device. The utility program commands are specified in the Utility Programs manual.

Volume/File Requirements for Utility Sets 1, 2, and 3

Volume and file requirements for Utility Sets 1, 2, and 3 depend on the particular function or functions to be performed when the utility program is loaded. The volume and file assignments specific to the individual functions of each utility program can be determined from the format requirements for each utility program command, as specified in the Utility Programs manual.

Input to Command Processor Before Utility Set 1, 2, or 3 Is Loaded

Logical file number 00 is restricted in meaning for Utility Sets 1, 2, and 3; it is used to attach the program file containing the utility program, as follows:

```
AT 00,DSKcccc,file-name
```

The AT 00 command need be specified only if the appropriate information does not already exist in the loader communication area of memory.

The load command must be submitted for each utility program. It specifies the member name of the utility program, as listed below.

Member Name	Utility Program
UTILL1	Utility Set 1
UTILL2	Utility Set 2
UTILL3	Utility Set 3

Utility Sets 1, 2, and 3 do not recognize logical file numbers 01 through 99 as having any special significance; any volume or device can be attached to any of these logical file numbers. For logical file numbers 01 through 99 the volume name (the volume-id field in the volume label) must be specified in each attach command to the Command Processor that specifies a disk, with the exception that the file name must be specified if the command input device is a disk.

The command file for Utility Sets 1, 2, and 3 can be assigned to the system console, a card reader, or a disk. If the command input device is a disk, the commands are stored in a partitioned file member on the disk volume.

To assign the command file to a disk or card reader:

1. Assign the device to any logical file number from 01 through 99 in an attach command to the Command Processor. If the device is a disk, specify in the attach command the name of the file that contains the member in which the commands are stored.
2. Specify the logical file number described in step 1 above as the first argument of the load command to the Command Processor for that utility program. If the device is a disk, specify the name of the member containing the commands as the second argument of the load command. The format for the load command if the command file is on disk or cards is:

$$\text{UTILL} \left\{\begin{matrix}1\\2\\3\end{matrix}\right\} \Delta\text{lfn}[,\text{member-name}]$$

If no logical file number appears in the load command, the command file is assigned, by default, to the system console.

The format of input commands to the Command Processor is specified in Section 2 of this part. Detailed information on commands to the Command Processor is given in the Command Processor section of the Program Development Tools manual.

MESSAGES ISSUED BY UTILITY SETS 1, 2, AND 3

The three utility sets issue coded error messages and uncoded error and information messages; these messages are listed in Tables III-10-1 and III-10-2, respectively. Coded messages are arranged numerically; uncoded messages are arranged alphabetically.

For a discussion of these messages, see "Messages Issued by System Programs" in Section 1 of this part. Also see Appendix G, "Explanation of Error Message Codes."

Table III-10-1. Coded Error Messages Issued by Utility Sets 1, 2, and 3

Message		Condition	Operator Action
Code	Other Information		
120001	SYSTEM ERROR	Utility program error. Program returns to command level.	Either (1) retry the command; or (2) enter QT, reload utility program, and retry; or (3) reboot the system, reload the program, and retry. If error message is reissued, contact your Honeywell service representative.
120003	SYSTEM ERROR	Utility program error. Program returns to command level.	See the operator action for error code 120001.
120005	SYSTEM ERROR	Utility program error. Program returns to command level.	See the operator action for error code 120001.
120006	SYSTEM ERROR	Utility program error. Program returns to command level.	See the operator action for error code 120001.
120101	cccc sswd	Channel number designated in cccc is being used to process an outstanding request. Program returns to command level.	Retry the command.
120102	cccc sswd	Invalid physical address passed by component to driver. cccc is probably an invalid channel number. Program returns to command level.	Retry the command.
120104	cccc sswd	Invalid parameter passed by component to driver (internal error). Program returns to command level.	Retry the command.

Table III-10-1 (cont). Coded Error Messages Issued by Utility Sets 1, 2, and 3

Message		Condition	Operator Action
Code	Other Information		
120105	cccc sswd	Device on channel number cccc is unavailable. If it is a line printer or card reader, program issues informational message and waits for reply. If it is a disk, program returns to command level.	If line printer or card reader, ready the device and reply to message. If disk, retry the command.
120106	cccc sswd	Time-out has occurred in the device on channel number cccc. If it is a printer or card reader, program issues informational message and waits for reply. If it is a disk, program returns to command level.	If printer or card reader, ready the device and reply to message. If a disk, retry the command.
120107	cccc sswd	Hardware error in device on channel number cccc. Status bits are specified by sswd. Program returns to command level.	Retry the command.
120F11		Attempt was made to reference a sector beyond the bounds of the unit (file or volume).	Correct and resubmit the entire command.
120F13		A directory entry containing a file-type code unknown to the system has been encountered.	Correct and resubmit the entire command.

Table III-10-2. Uncoded Error and Informational Messages Issued by Utility Sets 1, 2, and 3

Message	Related Utility Command(s)	Condition	Operator Action
DATA WILL BE TRUNCATED	CP	The destination file is not large enough to store the data contained in the source file. The end-of-data pointer is set to the physical end of the destination file.	No operator action required. For information only.
DECIMAL FIELD ERROR	All	Nonnumeric character encountered in a numeric field, or the field exceeds 65,535.	Resubmit the entire command.
DEFECTIVE SECTOR INDEX OVERFLOW	IN,AL,DL	More than 64 sectors have been flagged as being defective.	No operator action required. For information only.

Table III-10-2 (cont). Uncoded Error and Informational Messages Issued by Utility Sets 1, 2, and 3

Message	Related Utility Command(s)	Condition	Operator Action
DEVICE TYPES DIFFER	CM,CP	The devices specified in the command are of different types. Program returns to command level.	Correct and resubmit the entire command.
END OF COMMAND FILE	All	End of file indicator has been encountered on the command file. Control returns to Command Processor.	No operator action required. For information only.
FATAL HARDWARE ERROR		Follows all 1201xx error messages.	See appropriate 1201xx message for operator action.
FILE ALLOCATED	AL	The request to allocate a file by the previous AL command has been completed.	No operator action required. For information only.
FILE ALREADY EXISTS	AL,RN	An AL or RN request specifies a resultant file name that already exists on the specified volume.	Correct and resubmit the entire command.
FILE COPIED	CP	The request to copy a file by the previous CP command has been completed.	No operator action required. For information only.
FILE DELETED	DL	The request to delete a file by the previous DL command has been completed.	No operator action required. For information only.
FILE INITIALIZED	IN	The request to initialize a file by the previous IN command has been completed.	No operator action required. For information only.
FILE LENGTHS DIFFER	CM	The files being compared contain a different number of sectors. Only the number of sectors in the smaller file are compared.	No operator action required. For information only.
FILE RENAMED	RN	The request to rename a file by the previous RN command has been completed.	No operator action required. For information only.
FILE TYPES DIFFER	CM,CP	The command cannot be performed because the specified files are of different types.	Correct and resubmit the entire command.

Table III-10-2 (cont). Uncoded Error and Informational Messages Issued by Utility Sets 1, 2, and 3

Message	Related Utility Command(s)	Condition	Operator Action
FILE xxxxxxxxxxxx NOT FOUND	All commands accepting a file name as an operand	The file specified in the message was not found on the specified volume.	Correct and resubmit the entire command.
FILE xxxxxxxxxxxx SECTOR xxxx... UNLIKE FILE xxxxxxxxxxxx SECTOR xxxx...	CM	The specified relative sector(s) are nonmatching between the two specified files.	No operator action required. For information only.
FILES COMPARED	CM	The request to compare files by the previous CM command has been completed.	No operator action required. For information only.
INDEX SPACE NOT AVAILABLE	CP,XF	Index space needed to accept the required number of member names is not available.	Correct and resubmit the entire command.
INVALID COMMAND	All	The first two characters are not valid utility commands for this utility set.	Load the appropriate utility set and/or resubmit the corrected command.
INVALID COMMAND FILE	All	Logical file number in load command is >99, or device associated with lfn is neither disk nor card reader.	Correct and resubmit entire command.
INVALID FIELD LENGTH	All	Alphabetic field exceeds maximum allowable length.	Resubmit the corrected command.
INVALID FILE TYPE	LD,IN; all commands operating on partitioned files	File was not partitioned when it should have been, was a partitioned file specified in an LD command, or was a relative file with deletable records specified and was to be initialized for partitioned access.	Correct and resubmit the entire command.
INVALID LFN	All	The logical file number is not in the range of 0 through 99, or is not attached.	Resubmit the entire command.
INVALID OPERAND TYPE	All	Invalid logical file name for the command.	Resubmit the entire corrected command.

Table III-10-2 (cont). Uncoded Error and Informational Messages Issued by Utility Sets 1, 2, and 3

Message	Related Utility Command(s)	Condition	Operator Action
INVALID RECORD LENGTH	AL,IN	Either (1) the record length was greater than 65,535 bytes for a relative file as specified in an AL command, or (2) the IN command specified a file whose record length was not equal to the size of one sector.	Correct and resubmit the entire command.
MEMBER ALREADY EXISTS	CP,XF,RN	The file already contains the member name specified as output in the command.	Correct and resubmit the entire command.
MEMBER COMPARED	CP	The request to compare individual members by the previous CM command has been completed.	No operator action required. For information only.
MEMBER COPIED	CP	The request to copy an individual member by the previous CP command has been completed.	No operator action required. For information only.
MEMBER DELETED	DL	The request to delete a member by the previous DL command has been completed.	No operator action required. For information only.
MEMBER NOT FOUND	Any command accepting an input member name	The member specified as an input operand cannot be found in the input file.	Correct and resubmit the entire command.
MEMBER RENAMED	RN	The request to rename a member by the previous RN command has been completed.	No operator action required. For information only.
MEMBER xxxxxxxx BEING COMPARED	CM	The named members of the files specified in the CM command are being compared.	No operator action required. For information only.
MEMBER xxxxxxxx BEING COPIED	CP	The member named is being copied to the output file.	No operator action required. For information only.
MEMBER xxxxxxxx BEING IGNORED	CP	The member named is in the input file and is not being copied to the output file because a version of the member already exists and the replace option was not specified in the CP command.	No operator action required. For information only.

Table III-10-2 (cont). Uncoded Error and Informational Messages Issued by Utility Sets 1, 2, and 3

Message	Related Utility Command(s)	Condition	Operator Action
MEMBER xxxxxxxx BEING REPLACED	CP	The member named is being replaced in the output file by the version from the input file because the replace option was specified in the CP command.	No operator action required. For information only.
MEMBER xxxxxxxx NOT ON FILE xxxxxxxxxxxx	CM	The named member cannot be found in the named file although it exists in the other file specified in the CM command.	No operator action required. For information only.
MEMBERS DO NOT MATCH	CM	The file members specified in the CM command do not contain identical data records.	No operator action required. For information only.
MEMBERS xxxxxxxx AND xxxxxxxx COMPARE	CM	The named members of the files specified in the CM command are identical in contents. If all members of two files were compared, this message is issued for each pair of members with like names that have identical contents.	No operator action required. For information only.
MEMBERS xxxxxxxx AND xxxxxxxx DIFFER	CM	The named members of the files specified in the CM command are not identical in contents. If all members of two files were compared, this message is issued for each pair of members with like names that do not have identical contents.	No operator action required. For information only.
MEMBERS xxxxxxxx DO NOT MATCH	CM	The named members of the files specified in the CM command do not contain identical data records.	No operator action required. For information only.
NON-HONEYWELL VOLUME MOUNTED	All	An attempt has been made to access a non-Honeywell volume.	Mount Honeywell volume and resubmit the entire command.
NUMERIC FIELD ERROR	All	Non-hexadecimal digit in a numeric field or the range exceeds FFFF.	Resubmit the entire command.

Table III-10-2 (cont). Uncoded Error and Informational Messages Issued by Utility Sets 1, 2, and 3

Message	Related Utility Command(s)	Condition	Operator Action
READY DEVICE REPLY R TO RETRY, A TO ABORT		The printer device or card reader is not operational.	Ready the printer or card reader, if possible. Reply "R" to retry the request; "A" to return to command level.
RECORD LENGTHS DIFFER	CP	The input and output files have different record lengths.	No operator action required. For information only.
REMOTE EXTENT BLOCK ERROR	CM,CP,DL,LD, LS,PT	A volume directory entry which should contain a remote extent block does not.	No operator action required. For information only.
SECTOR xxxx UNREADABLE	All commands accepting disk input	The sector specified (by either a volume-relative or file-relative sector number, depending on the command) cannot be read.	No operator action required. The utility program continues unless the problem is in the system-maintained area. When the request has been completed, the command can be resubmitted.
SECTOR xxxx UNWRITABLE	All commands with disk output	The sector specified (by either a volume-relative or file-relative sector number, depending on the command) cannot be written.	The utility program returns to command level. Resubmit the command.
SECTORS xxxx... DEFECTIVE	IN,AL,DL	The volume-relative sectors specified in the error message are defective.	No operator action required. For information only.
SPACE NOT AVAILABLE	IN	Sufficient data space is not available to accommodate at least one sector of data for each member.	Correct and resubmit the entire command.
SPACE NOT AVAILABLE	CP,XF	Sufficient data space in the output file is not available to accomplish the request.	Correct and resubmit the entire command.
UNABLE TO FIND COMMAND FILE	All	The file named in the attach command for the command file could not be found, or the file member named in the load command could not be found.	Correct and resubmit entire command.

Table III-10-2 (cont). Uncoded Error and Informational Messages Issued by Utility Sets 1, 2, and 3

Message	Related Utility Command(s)	Condition	Operator Action
UNABLE TO FORMAT VOLUME	IN	The volume cannot be formatted in its entirety. The utility program returns to the command level.	Resubmit the command.
UNABLE TO READ COMMAND FILE	All	A read error occurred while the command file was being read. Control returns to Command Processor.	Correct the error. Reload the utility program and retry the command.
UNABLE TO READ VOLUME BIT MAP	AL,DL,IN,LS	The volume bit map cannot be read. The utility program returns to the command level.	Resubmit the command.
UNABLE TO READ VOLUME DIRECTORY		The volume directory cannot be read. Program returns to command level.	Resubmit the command.
UNABLE TO READ VOLUME LABEL	All	The volume label cannot be read. The utility program returns to command level.	Resubmit the command.
UNABLE TO WRITE VOLUME BIT MAP	AL,DL	The volume bit map cannot be written. The utility program returns to command level.	Resubmit the command.
UNABLE TO WRITE VOLUME LABEL	IN	The volume label cannot be written. The utility program returns to command level.	Resubmit the command.
UTILITY SET n - RELEASE rrr UTIL READY		The named utility program has been loaded and is ready for a command.	Submit a command.
VOLUME COPIED	CP	The request to copy a volume by the previous CP command has been completed.	No operator action required. For information only.
VOLUME INITIALIZED	IN	The request to initialize a volume by the previous IN command has been completed.	No operator action required. For information only.
VOLUME RENAMED	RN	The request to rename a volume by the previous RN command has been completed.	No operator action required. For information only.

Table III-10-2 (cont). Uncoded Error and Informational Messages Issued by Utility Sets 1, 2, and 3

Message	Related Utility Command(s)	Condition	Operator Action
VOLUME xxxxxx NOT MOUNTED	All	A request has been made to access a volume that has not been mounted or whose volume-id does not match the volume name specified in the attach entry.	Mount the correct volume or correct the attach entry and resubmit the entire command.
VOLUMES COMPARED	CP	The request to compare volumes by the previous CP command has been completed.	No operator action required. For information only.

SECTION 11
BOOTSTRAP GENERATOR

Bootstrap Generator is a utility program whose functions are described in the Utility Programs manual. The two types of output produced by Bootstrap Generator are:

- A bootstrap record containing preset parameter values, loaded into memory during bootstrapping and loading of a program or component. (A bootstrap record created by Bootstrap Generator must be placed on each disk or paper tape on which a loadable program or component is stored.)
- A special bootstrap record named MDUMP, used to obtain a memory dump on a diskette for later printing.

This section describes the operating procedures for loading Bootstrap Generator. The use of a bootstrap record in bootstrapping and loading a particular program or component into memory is described in the section of this manual that presents operating procedures for that program or component.

The steps in using the MDUMP bootstrap record to perform a memory dump are given in Appendix E.

OPERATING PROCEDURES FOR BOOTSTRAP GENERATOR

Before executing Bootstrap Generator, take the following steps:

1. Mount the disk volume containing the Command Processor and Bootstrap Generator, if the volume is not already mounted.
2. Perform the bootstrapping and loading procedure for system programs specified in Section 1 of this part, if necessary. During this procedure, the Disk Loader loads the Command Processor into memory.
3. Mount the volume containing the input file for Bootstrap Generator, if an input file is required.
4. Mount the volume or make available the device for the output file for Bootstrap Generator.
5. Make available the devices for the list and command files, as needed.
6. Enter the commands to the Command Processor required before Bootstrap Generator is loaded into memory (see "Input to Command Processor Before Bootstrap Generator Is Loaded" below); this attaches the peripherals and causes Bootstrap Generator to be loaded into memory.

During execution of Bootstrap Generator, submit commands to the program through the specified command input device. The Bootstrap Generator commands are specified in the Utility Programs manual.

Volume/File Assignments for Bootstrap Generator

The volume/file assignments used during execution of Bootstrap Generator are described in Table III-11-1. These volumes and files are assigned to devices in attach commands to the Command Processor. (See the Program Development Tools manual for a description of assignment of files using the Command Processor.)

Table III-11-1. Volumes/Files Used By Bootstrap Generator

File	Device Types to Which the File Can Be Assigned	Comments
Program	Disk	The file from which Bootstrap Generator is loaded. An attach (AT) command to the Command Processor for logical file number 00 is needed only if the program file name and channel number are not specified in the loader communication area. The member name BTGEN must be specified in the load command to the Command Processor. If output is to an ASR, Bootstrap Generator will automatically call and load Utility Set 2.
Input	Disk	Optional volume. Assigned by AT command to logical file number 01. An input volume containing the member TRPHND (the offline trap handler) must be specified if the output is assigned to an ASR attached to logical file number 03. No input volume or file is required if the output is assigned to a disk.
List	System Console KSR or CRT Printer	Required only if the DFT parameter is given the value Y or N; in this case, the list file is used to print the status of the specialized bootstrap record being created by the Bootstrap Generator. Assigned by AT command to the Command Processor to logical file number 02. If the AT 02 command is not specified, the list file is assigned to the system console, whose channel number is specified in the loader communication area.

Table III-11-1 (cont). Volumes/Files Used By Bootstrap Generator

File	Device Types to Which the File Can Be Assigned	Comments
Output	Disk ASR	Required volume/file. Assigned by AT 03 command to the Command Processor. If the output is assigned to a disk, the volume name (the volume-id field in the volume directory) must be specified in the AT 03 command; the output will be a specialized bootstrap record stored at the beginning of that volume. If the bootstrap record MDUMP is to be created, the output must be assigned to a diskette. If the output file is on an ASR, the output punched on paper tape consists of a specialized bootstrap record followed by the Paper Tape Loader.
Command	System Console KSR or CRT Card Reader Disk	Required file. The system console whose channel number is in the loader communication area is assumed to be the command input device unless another device is specified in an AT 04 command to the Command Processor. When the command file is assigned to a disk, the name of the member that contains the commands must be specified in the load command.

Input to Command Processor Before Bootstrap Generator Is Loaded

Figure III-11-1 illustrates the full set of commands that can be submitted to the Command Processor immediately before Bootstrap Generator is loaded. The load command must always be entered. The other commands should be entered, as required, provided appropriate information does not exist in the loader communication area and attach table after the Command Processor is loaded into memory.

If the bootstrap record being created is to be placed on a disk, the volume name is specified in the AT 03 command to the Command Processor.

If the command file is assigned to a disk, the load command line for Bootstrap Generator must contain an argument specifying the name of the command file member that contains the commands to be submitted to Bootstrap Generator. The format for this load command line is:

BTGENΔmember-name

Command Type	Command Format	Optional or Required
Load Initialize	LI normal options	Optional
Attach (Program File)	AT 00,DSKcccc,file-name	Optional
Attach (Input File)	AT 01,DSKcccc,volume-id	Optional
Attach (List File)	AT 02,sdncccc	Optional
Attach (Output File)	AT 03,sdncccc[,volume-id]	Required
Attach (Command File)	AT 04,sdncccc[,file-name]	Optional
Load	BTGEN[Δmember-name]	Required

Figure III-11-1. Input to Command Processor Before Bootstrap Generator Is Loaded

For detailed information on input commands to the Command Processor, see the Command Processor section of the Program Development Tools manual.

MESSAGES ISSUED BY BOOTSTRAP GENERATOR

Bootstrap Generator issues coded error messages and uncoded error and informational messages; these messages are listed in Tables III-11-2 and III-11-3, respectively. Coded messages are arranged numerically; uncoded messages are arranged alphabetically.

For a discussion of these messages, see "Messages Issued By System Programs" in Section 1 of this part. Also see Appendix G, "Explanation of Error Message Codes."

Table III-11-2. Coded Error Messages Issued By Bootstrap Generator

Message		Condition	Operator Action
Code	Other Information		
300001		Bootstrap Generator error. Program returns to Command Processor.	Either reload Bootstrap Generator and retry; or reboot the system, reload Bootstrap Generator, and retry. If error message is reissued, contact your Honeywell service representative.
300003		Bootstrap Generator error. Program returns to Command Processor.	See the operator action for error code 300001.

Table III-11-2 (cont). Coded Earor Messages Issued By Bootstrap Generator

Message		Condition	Operator Action
Code	Other Information		
300006		Bootstrap Generator error. Program returns to Command Processor.	See the operator action for error code 300001.
300101	cccc sswd	Channel number designated in cccc is being used to process an outstanding request. Program returns to Command Processor.	Reload Bootstrap Generator and retry the command.
300102	cccc sswd	Invalid physical address passed by component to driver. cccc is probably an invalid channel number. Program returns to Command Processor.	Reload Bootstrap Generator and retry the command.
300104	cccc sswd	Invalid parameter passed by component to driver (internal error). Program returns to Command Processor.	Reload Bootstrap Generator and retry the command.
300105	cccc sswd	Device on channel number cccc is unavailable. Program returns to Command Processor.	Reload Bootstrap Generator and retry the command.
300106	cccc sswd	Time-out has occurred in device on channel number cccc. Program returns to Command Processor.	Reload Bootstrap Generator and retry the command.
300107	cccc sswd	Hardware error in device on channel number cccc. Status bits are specified by sswd. Program returns to Command Processor.	Reload Bootstrap Generator and retry the command.

Table III-11-3. Uncoded Error and Informational Messages Issued By Bootstrap Generator

Message	Condition	Operator Action
ASR BOOTSTRAP BLOCK TOO LARGE	Fatal error. Bootstrap block is larger than 250 words. Control returns to Command Processor.	No operator action possible.
CMD MEMBER NAME MISSING	The member name of the command file was not entered in load command.	Enter correct load command.
DUMP FILE NOT FOUND	File named DUMPFILE is not allocated in the volume named DMPVOL.	Allocate DUMPFILE on DMPVOL and reload Bootstrap Generator.
GO?	Request for response to question: is bootstrap record status listing acceptable?	Reply N, not acceptable; Y, acceptable (create bootstrap record); or Q, return to Command Processor (without creating a bootstrap record).
INPUT FILE NOT ATTACHED	No volume is attached to lfn 01.	Attach input volume to lfn 01. Output file must be an ASR. Reload Bootstrap Generator.
INVALID INPUT DEVICE	Unsupported device is attached to lfn 01.	Attach correct device to lfn 01. Reload Bootstrap Generator.
INVALID LIST FILE	Device to which list file assigned is not a KSR, CRT or line printer.	Attach list file to valid device. Reload Bootstrap Generator.
INVALID OUTPUT DEVICE	Unsupported device is attached to lfn 03.	Attach correct device to lfn 03. Reload Bootstrap Generator.
INVALID OUTPUT FILE	Invalid file is attached to lfn 03.	Attach correct file to lfn 03. Reload Bootstrap Generator.
INVALID PARAMETER(S)	Invalid parameters entered. Control does not return to Command Processor if command input device is a KSR or CRT.	If the command input device is a KSR or CRT, reenter parameters. Otherwise, correct the command file and reload Bootstrap Generator.
INVALID VOLUME NAME	The volume named DMPVOL is not mounted.	Mount DMPVOL and reload Bootstrap Generator.
OUTPUT FILE NOT ATTACHED	No file is attached to lfn 03.	Attach output file to lfn 03. Reload Bootstrap Generator.
PARAM?	Request for entry of parameter values.	Enter parameter values.

Table III-11-3 (cont). Uncoded Error and Informational Messages Issued By Bootstrap Generator

Message	Condition	Operator Action
PARAM. NOT ENTERED	Parameters have not been submitted to Bootstrap Generator.	If the command input device is a KSR or CRT, enter the parameters. Otherwise, correct the command file and reload Bootstrap Generator.
PARAMETER(S) ERROR $P_n[,P_n]...$	Parameter(s) listed were incorrect as entered; n represents any number from 1 through 9.	Resubmit corrected parameter(s).
VOLUME NOT MOUNTED	No volume is mounted with a volume name corresponding to the volume-id specified in an attach command.	Mount correct volume and reload Bootstrap Generator.

SECTION 12
DUMP EDIT

Dump Edit is a utility program whose functions are described in the Utility Programs manual.

OPERATING PROCEDURES FOR DUMP EDIT

Before executing Dump Edit, take the following steps:

1. Mount the disk volume containing the Command Processor and Dump Edit, if the volume is not already mounted.
2. Perform the bootstrapping and loading procedure for system programs specified in Section 1 of this part, if necessary. During this procedure, the Disk Loader loads the Command Processor into memory.
3. Mount the volume containing the input file for Dump Edit; this volume must be named DMPVOL and must contain the file DUMPFILE in which a memory dump is stored.
4. Make available the devices for the list and command files, as required.
5. Enter the commands to the Command Processor required before Dump Edit is loaded into memory (see "Input to Command Processor Before Dump Edit Is Loaded" below); this attaches the peripherals and causes Dump Edit to be loaded into memory.

During execution of Dump Edit, submit commands to the program through the specified command input device. The Dump Edit commands are specified in the Utility Programs manual.

File Assignments for Dump Edit

The file assignments used during execution of Dump Edit are described in Table III-12-1. These files are assigned to devices in attach commands to the Command Processor. (See the Program Development Tools manual for a description of assignment of files using the Command Processor.)

Input to Command Processor Before Dump Edit is Loaded

Figure III-12-1 illustrates the full set of commands that can be submitted to the Command Processor immediately before Dump Edit is loaded. The load command must always be entered. The other commands should be entered, as required, provided appropriate information does not exist in the loader communication area and attach table after the Command Processor is loaded into memory.

Table III-12-1. Files Used by Dump Edit

File	Device Types to Which the File Can Be Assigned	Comments
Program	Disk	The file from which Dump Edit is loaded. An attach (AT) command to the Command Processor for logical file number 00 is needed only if the program file name and channel number are not specified in the loader communication area. The member name DPEDIT must be specified in the load command to the Command Processor.
Input	Diskette	Required file. Assigned by AT command to logical file number 01. The file name DUMPFILE must be specified in the AT 01 command.
List	Line Printer Serial Printer	Required file. Assigned by AT 02 command to the Command Processor.
Command	System Console KSR or CRT	Required file. Specified in an AT 04 command to the Command Processor. If no AT 04 command is specified, the system console whose channel number is in the loader communication area is assumed to be the command input device.

Command Type	Command Format	Optional or Required
Load Initialize	LI normal options	Optional
Attach (Program File)	AT 00,DSKcccc,file-name	Optional
Attach (Input File)	AT 01,DSKcccc,DUMPFILE	Required
Attach (List File)	AT 02,sdncccc	Required
Attach (Command File)	AT 04,KSRcccc	Optional
Load	DPEDIT	Required

Figure III-12-1. Input to Command Processor Before Dump Edit Is Loaded

The file name DUMPFILE must be specified in the AT 01 command to the Command Processor. The input diskette whose channel number is specified in the AT 01 command must contain the bootstrap record MDUMP and the volume DMPVOL, in which DUMPFILE is located. A memory dump to DUMPFILE must have been made before Dump Edit is loaded.

For detailed information on input commands to the Command Processor, see the Command Processor section of the Program Development Tools manual.

MESSAGES ISSUED BY DUMP EDIT

Dump Edit issues coded error messages and uncoded error and informational messages; these messages are listed in Tables III-12-2 and III-12-3, respectively. Coded messages are arranged numerically; uncoded messages are arranged alphabetically.

Table III-12-2. Coded Error Messages Issued By Dump Edit

Message		Condition	Operator Action
Code	Other Information		
250001	FATAL ERROR lfn	Fatal error. Control returns to Command Processor.	Reload Dump Edit and retry.
250003	FATAL ERROR lfn	Fatal error. Control returns to Command Processor.	Reload Dump Edit and retry.
25000A	FILE NOT FOUND 01	DUMPFILE cannot be found on the diskette. Control returns to Command Processor.	Correct the error, reload Dump Edit, and retry.
250102	cccc sswd	Invalid physical address passed by Dump Edit to driver (internal error). Control returns to Command Processor.	Reload Dump Edit and retry.
250104	cccc sswd	Invalid parameter(s) passed by Dump Edit to driver (internal error). Control returns to Command Processor.	Reload Dump Edit and retry.
250105	cccc sswd	Device (either diskette or KSR) is unavailable. Control returns to Command Processor. (Message not applicable to printer.)	Reload Dump Edit and retry.
250106	cccc sswd	Time-out has occurred in the device on channel number cccc. Control returns to Command Processor.	Reload Dump Edit and retry.
250107	cccc sswd	Hardware error other than read, search or seek error in device on channel number cccc. Status bits are specified by sswd. Control returns to Command Processor.	Reload Dump Edit and retry.

Table III-12-2 (cont). Coded Error Messages Issued by Dump Edit

Message		Condition	Operator Action
Code	Other Information		
250F01	ATTACH ERROR lfn	The device assigned to the specified logical file number is not attached or is attached incorrectly. Control returns to Command Processor.	Attach the device correctly, reload Dump Edit, and retry.
250F02	OPEN ERROR lfn	Unable to open the device assigned to the specified logical file number. Control returns to Command Processor.	Correct the error, reload Dump Edit, and retry.

Table III-12-3. Uncoded Error and Informational Messages Issued by Dump Edit

Message	Condition	Operator Action
DPEDIT OF nnK DUMP, READY FOR "GO"	Dump Edit is ready to accept a GO command; nn is a decimal number representing the size of the memory area dumped.	Submit a GO or QT command.
DUMP COMPLETE	Printing of memory dump is completed.	No operator action required. For information only.
INCORRECT ENTRY -- START OVER	Syntax error. Invalid GO command was submitted or invalid nonhexadecimal value was entered for size of trap save area or contents of P- or S-register.	After Dump Edit issues a READY FOR "GO" message, resubmit corrected entry.
READY PRINTER -- REPLY C TO CONTINUE, R TO RESTART, A TO ABORT	The printer device is not operational.	Ready printer if possible. Enter C to continue printing from current position; R to reprint entire dump (starting with page 1); or A to return to command level (where GO or QT can be submitted).

For a discussion of these messages, see "Messages Issued by System Programs" in Section 1 of this part. Also see Appendix G, "Explanation of Error Message Codes."

SECTION 13
PROGRAM PATCH

Program Patch is a utility program whose functions are described in the Utility Programs manual.

OPERATING PROCEDURES FOR PROGRAM PATCH

Before executing Program Patch, take the following steps:

1. Mount the disk volume containing the Command Processor and Program Patch, if the volume is not already mounted.
2. Perform the bootstrapping and loading procedure for system programs specified in Section 1 of this part, if necessary. During this procedure, the Disk Loader loads the Command Processor into memory.
3. Mount the volume containing the input file for Program Patch and any volume containing patch members that may be required.
4. Mount the volume containing the output file for Program Patch, if the file is required and is on a different volume from the input file.
5. Make available the devices for the list and command files, as needed.
6. Enter the commands to the Command Processor required before Program Patch is loaded into memory (see "Input to Command Processor Before Program Patch Is Loaded" below); this attaches the peripherals and causes Program Patch to be loaded into memory.

During execution of Program Patch, submit commands to the utility program as specified in "Program Patch" in the Utility Programs manual.

Program Patch File Requirements

Table III-13-1 lists and describes the files used by Program Patch. These files are assigned to devices in attach commands to the Command Processor, as described below.

Table III-13-1. Files Used During Loading and Execution of Program Patch

File	Device Types to Which the File Can Be Assigned	Comments
Program	Disk	The file from which Program Patch is loaded. An attach (AT) command to the Command Processor for logical file number 00 is needed only if the program file name and channel number are not specified in the loader communication area. The member name PATCH must be specified in the load command to the Command Processor.
Input (Object or Image Modules)	Disk	Required file. Assigned by AT command to any logical file number except 00, 02, 04. File containing member to be patched or from which patches are to be deleted. If an AP command is to be executed, a second input file (containing a patch member) must be assigned to a different logical file number. Each file can reside on a separate disk pack.
List	System Console KSR or CRT Printer	Required only if an LP command is to be executed. Assigned to logical file number 02 by AT command to the Command Processor; if this command is unspecified, the list file is assigned to the system console, whose channel number is specified in the loader communication area.
Output (Object or Image Module)	Disk	Required file. Assigned by AT command to either: 1. The same logical file number as the input file, or 2. Any other logical file number except 00, 02, 04. If the output is assigned to a separate logical file number, the file can reside on a separate disk pack.
Command	System Console KSR or CRT Card Reader Disk	Required file. Assigned to logical file number 04 by AT command to the Command Processor; this AT command is not needed if the command file is assigned to the system console on the channel number specified in the loader communications area. If a disk is the command input device, the member name of the command file must be specified in the load command.

Input to Command Processor Before Program Patch is Loaded

Figure III-13-1 illustrates the full set of commands that can be submitted to the Command Processor immediately before Program Patch is loaded. The load command must always be entered. The other commands should be entered, as required, provided appropriate information does not exist in the loader communication area and attach table after the Command Processor is loaded into memory.

If a file is assigned to a disk, the file name must be specified in the appropriate attach command to the Command Processor.

Command Type	Command Format	Optional or Required
Load Initialize	LI normal options	Optional
Attach (Program File)	AT 00,DSKcccc,file-name	Optional
Attach (Input File or Files)	AT nn,DSKcccc,file-name	Required
Attach (List File)	AT 02,sdncccc	Optional
Attach (Output File)	AT nn,DSKcccc,file-name	Optional
Attach (Command File)	AT 04,sdncccc	Optional
Load	PATCH[△member-name]	Required

Figure III-13-1. Input to Command Processor Before Program Patch is Loaded

If the command file is assigned to a disk, the load command line for Program Patch must contain an argument specifying the name of the command file member that contains the patch commands, separated from the name PATCH by a space.

For detailed information on input commands to the Command Processor, see the Command Processor section of the Program Development Tools manual.

MESSAGES ISSUED BY PROGRAM PATCH

Program Patch issues coded error messages and uncoded error and informational messages; these messages are listed in Tables III-13-2 and III-13-3, respectively. Coded messages are arranged numerically; uncoded messages are arranged alphabetically.

For a description of system programs messages, see "Messages Issued by System Programs" in Section 1 of this part. Also see Appendix G.

Table III-13-2. Coded Error Messages Issued By Program Patch

Message		Condition	Operator Action
Code	Other Information		
210001	FATAL ERR	Program Patch error. Patch returns to Command Processor level.	1. Reload Patch and retry. 2. If the error message is reissued, call your Honeywell service representive.
210002	MEMBER NOT FOUND	Specified member not found in input file. Patch returns to Command Processor level.	Correct the error and reload Program Patch.

Table III-13-2 (cont). Coded Error Messages Issued By Program Patch

Message		Condition	Operator Action
Code	Other Information		
210002	DUPLICATE MEMBER	Duplicate member exists in output file. Patch returns to Command Processor level.	Correct the error and reload Program Patch.
210003	FATAL ERR	Program Patch error. Patch returns to Command Processor level.	See the operator action for error code 210001.
210004	NO DATA SPACE	Output file has no data space available for the patch. Patch returns to Command Processor level.	Correct the file and reload Program Patch.
210005	NO INDEX SPACE	Output file has no index space available for additional member names. Patch returns to Command Processor level.	Correct the file and reload Program Patch.
210006	FATAL ERR	Program Patch error. Patch returns to Command Processor level.	See operator action for error code 210001.
21000A	FILE NOT FND lfn	File specified by lfn is not a partitioned disk file or is not found on disk device. Patch returns to Command Processor level.	Correct the file or mount correct disk volume, and reload Program Patch.
21000C	FILE NOT PART lfn	The file found is not partitioned.	Either (1) enter QT and specify a partitioned file in the attach command that caused error, or (2) reallocate the file so that it is partitioned.
210101	cccc sswd	Channel number designated in cccc is being used to process an outstanding request. Patch returns to Command Processor level.	Reload Program Patch.

Table III-13-2 (cont). Coded Error Messages Issued By Program Patch

Message		Condition	Operator Action
Code	Other Information		
210102	cccc sswd	Invalid physical address passed by component to driver, cccc is probably an invalid channel number. Patch returns to Command Processor level.	Reload Program Patch.
210104	cccc sswd	Invalid parameter passed by component to driver (internal error). Patch returns to Command Processor level.	Reload Program Patch.
210105	cccc sswd	Device on channel number cccc is unavailable. If device is line printer or card reader, Patch halts. If device is disk, Patch returns to Command Processor level.	If line printer or card reader, press Ready, then Execute. If disk, reload Program Patch.
210106	cccc sswd	Time-out has occurred in the device on channel number cccc. If device is printer or card reader, Patch halts. If device is disk, Patch returns to Command Processor level.	If printer or card reader, press Ready, then Execute. If disk, reload Program Patch.
210107	cccc sswd	Hardware error in device on channel number cccc. Status bits are specified by sswd. Patch returns to Command Processor level.	Reload Program Patch.

Table III-13-3. Uncoded Error and Informational Messages Issued By Program Patch

Message	Related Utility Command(s)	Condition	Operator Action
DUPLICATE PATCH ID	AP,HP	The patch-id specified by the previous command already exists in the input member. This patch is ignored and Program Patch continues.	No operator action required. Message for information only.
INPUT FILE NOT ATTACHED		Required input file not attached. Patch returns to Command Processor level.	Program Patch terminates. Attach input file and reload Program Patch.
INVALID COMMAND	All	An invalid Patch command was submitted or a slash character (/) was not submitted as the first character on the HP data line.	Correct the command and resubmit it.
INVALID HEX DIGIT	HP	An invalid hexadecimal digit was specified as part of an HP command.	Correct the command and resubmit it.
INVALID ITEM TYPE	All	An invalid object or load module is encountered during creation of the output module. The invalid item type is in the original member and is probably a previous patch. Patch returns to Command Processor level.	Re-create the input module with original patches and reload Program Patch.
INVALID KEYWORD	EP	-ALL was not specified correctly in the EP command. Patch returns to command level.	Correct the appropriate command and resubmit it.
INVALID LFN	AP, initial patch command	The logical file number (lfn) specified in the add patch command or the initial patch command (output member) is not in the range of 01 $\leq$ 99.	Correct the appropriate command and resubmit it.

Table III-13-3 (cont). Uncoded Error and Informational Messages Issued By Program Patch

Message	Related Utility Command(s)	Condition	Operator Action
INVALID PARAMETER	All	One of the parameters (operands) submitted in a Patch command is invalid.	Correct the parameter (operand) and resubmit the command.
INVALID TEXT TYPE	AP	The module containing the patches does not contain the same type of text (object/ image) as the member to be patched. Patch returns to Command Processor level.	Correct the module if necessary and reload Program Patch.
OUTPUT FILE NOT ATTACHED		Required output file is not attached. Patch returns to Command Processor level.	Program Patch is terminated. Attach output file and reload Program Patch.
PATCH ID TABLE OVERFLOW	All	More than 50 patches are applied to one module. Patch returns to Command Processor level.	Remove patches as necessary. Reload Program Patch.
TRANSACTION FILE NOT ATTACHED	AP	Transaction file containing member(s) required by add patch command is not attached. Patch returns to Command Processor level.	Program Patch is terminated. Attach required file and reload Program Patch.

SECTION 14

DEBUGGER (OFFLINE)

The Debugger is an offline utility program whose functions are described in the Utility Programs manual.

The Debugger can be loaded in one of three forms:

1. As a separate load module, not linked to a user-written offline application program.
2. As part of a load module in which it is prelinked to a user-written offline application program, with the starting address of the Debugger being the initial address of the load module. The starting address of the Debugger must have been specified in the START command to the Linker when the load module was created.
3. As part of a load module in which it is prelinked to a user-written offline application program, with the starting address of the user-written program being the initial address of the load module. The starting address of the user-written program must have been specified in the START command to the Linker when the load module was created.

The procedure for loading the Debugger in each of the above forms is described below.

LOADING THE OFFLINE DEBUGGER AS A SEPARATE LOAD MODULE

To load the Debugger when it is not prelinked to a user-written offline application program, take the following steps:

1. Mount the disk volume containing the Debugger, if the volume is not already mounted.
2. Perform the bootstrapping and loading procedure for system programs as described in Section 1 of this part, if necessary. During this procedure, the Disk Loader loads the Command Processor into memory.
3. Enter the commands to the Command Processor that cause the Debugger to be loaded (see "Input to Command Processor Before Debugger Is Loaded" below). Specify the halt option in the load initialize command to the Command Processor.
4. The Debugger is then loaded and the loader halts in response to the halt option specified in the load initialize command. Enter the following before the Debugger begins execution:
   a. Set register R1 to contain the channel number of the KSR-like device to be used to submit commands to the Debugger. (This device can be the system console or any other KSR or CRT used as a command input device.)

b. Set register R2 to contain the channel number of a line or serial printer.

c. Set the P-register to the starting address of the Debugger.

5. To cause the Debugger to begin execution, press Execute.

During execution of the Debugger, submit commands to the Debugger as specified in "Debugger" in the Utility Programs manual.

Input to Command Processor Before Debugger Is Loaded

The only commands to the Command Processor required for loading the Debugger are the load initialize command, the attach command that attaches to logical file number 00 the program file containing the Debugger, and the load command DEBUG. The AT 00 command need be specified only if the program file name is not already stored in the loader communication area.

If you have not attached the program file, enter the following commands to the Command Processor to load the Debugger.

```
LI H[,rel]
AT 00,DSKcccc,file-name
DEBUG
```

Setting Registers Before Debugger Begins Execution

It is necessary to specify the halt option (H) in the load initialize command so that you can set registers R1 and R2 to the channel numbers of the KSR and line or serial printer, respectively, and set the P-register to the starting address of the Debugger, before the Debugger begins execution.

LOADING THE PRELINKED DEBUGGER WHEN IT IS THE INITIAL COMPONENT OF THE LOAD MODULE

When the Debugger is linked to a user-written offline application program and the starting address of the Debugger is the initial address of the load module, the loading procedure is the same as though the Debugger were to be loaded as a separate load module, not linked to the user-written program (see "Loading the Offline Debugger as a Separate Load Module" above). The load command to the Command Processor must specify the name of the load module.

To start the user-written program once the Debugger has begun execution, use the debugging command $R, the restart command.

LOADING THE PRELINKED DEBUGGER WHEN IT IS NOT THE INITIAL COMPONENT OF THE LOAD MODULE

When the Debugger is linked to a user-written offline application program and the starting address of the user-written program is the initial address of the load module, the Debugger can be entered only through the program to which it is linked. The following call to the Debugger must be written in the program:

```
LNJ $B5,ZHDBGI
```

Before the program calls the Debugger, it must enter the channel numbers of the KSR and line or serial printer in registers R1 and R2, respectively.

The procedure for loading the user-written offline application program is given in Part V of this manual.

MESSAGES ISSUED BY THE OFFLINE DEBUGGER

The Debugger issues only two messages, as specified in Table III-14-1. These messages are issued on the system console or other KSR-like command input device, whichever is on the channel whose number is specified in register R1.

Table III-14-1. Messages Issued By Offline Debugger

Message	Condition	Operator Action
DEBUG READY	The Debugger is ready to accept commands.	Submit debugging commands, beginning on the next line after the message.
U?	The Debugger detects an error related to the last command submitted.	Correct the error and resubmit the command.

PART IV

OPERATING PROCEDURES FOR ONLINE APPLICATIONS

SECTION 1
METHODS OF LOADING ONLINE APPLICATIONS

Configuring and loading an online application program is a continuous process, consisting of the following steps:

- The loader loads the Configuration Load Manager (CLM).
- The CLM configures the online application in response to commands submitted to the CLM.
- The CLM loads the load modules to be used in the online application and initializes the system.

The CLM may be stored either on a disk or on paper tape. Loading the CLM from a disk is described in Section 2 of this part. Included in the section are separate procedures for the following:

- Bootstrapping and loading the CLM from a disk using the Disk Loader, Command Processor, and operator's console
- Bootstrapping and loading the CLM from a disk using the Disk Loader and control panel (including basic control panel)

Section 3 covers the third alternative procedure:

- Bootstrapping and loading the CLM from paper tape, using the Paper Tape Loader and control panel

The process by which the CLM configures and constructs an online application program is described in detail in the Planning and Building an Online Application manual. The format descriptions of commands to be submitted to the CLM are included in that manual. The initial assignment of the CLM command file to a specific device occurs during bootstrapping and loading of the CLM (as described in Sections 2 and 3 of Part IV).

Control panel operations specified in this part are described in detail under "Control Panel," Section 1 of Part II of this manual. All registers used in the following procedures are identified by selection code in Table II-1-4, "Register Selection Codes."

Loader halts specified in the following procedures are described in Appendix A, "Disk Loader," and Appendix B, "Paper Tape Loader," as appropriate.

Error messages issued to the operator by the CLM are listed in Section 4 of Part IV; online error messages are listed in Section 5.

SECTION 2
BOOTSTRAPPING AND LOADING ONLINE APPLICATIONS FROM DISK

The procedures for bootstrapping and loading the Configuration Load Manager (CLM) using the Disk Loader are described in this section. The two ways to load the CLM from a disk are:

- Using the operator's console and the Command Processor
- Using the control panel, without the Command Processor

Each of the above bootstrapping and loading procedures is discussed separately below.

Once the CLM is loaded, it accepts commands from a command input stream that cause it to load the CLM load modules, configure the online application, and then load the load modules that constitute the online application: the Executive modules, I/O drivers, and user-written application program.

BOOTSTRAPPING AND LOADING CLM FROM DISK USING CONSOLE

Bootstrapping and loading the CLM into memory using the Disk Loader and the operator's console involves:

- Invocation of the bootstrapping procedure which causes the boot loader to be loaded into memory
- Use of the boot loader to load the Disk Loader, which loads the Offline Trap Handler and the Command Processor into memory
- Submission of commands to the Command Processor that contain information to be used in loading and running the CLM
- Loading of the CLM by the Disk Loader, which then turns control over to the CLM

Parameter values preset in the bootstrap record and boot loader when they were created by Bootstrap Generator are used during bootstrapping and loading. The values for the high memory address and operator's console channel number that were preset in the bootstrap record and boot loader are referred to as "default" values in the bootstrapping and loading procedures that follow, because they can be modified through the control panel. The indicator setting for loader halt 1601 that was preset in the boot loader cannot be modified during bootstrapping and loading.

Bootstrapping and loading the CLM from disk using the operator's console, summarized in Figure IV-2-1, consists of these phases (as described in the following subsections):

- Mounting the required disk volume(s)
- Performing one of two possible bootstrapping procedures
- Performing loading of the CLM

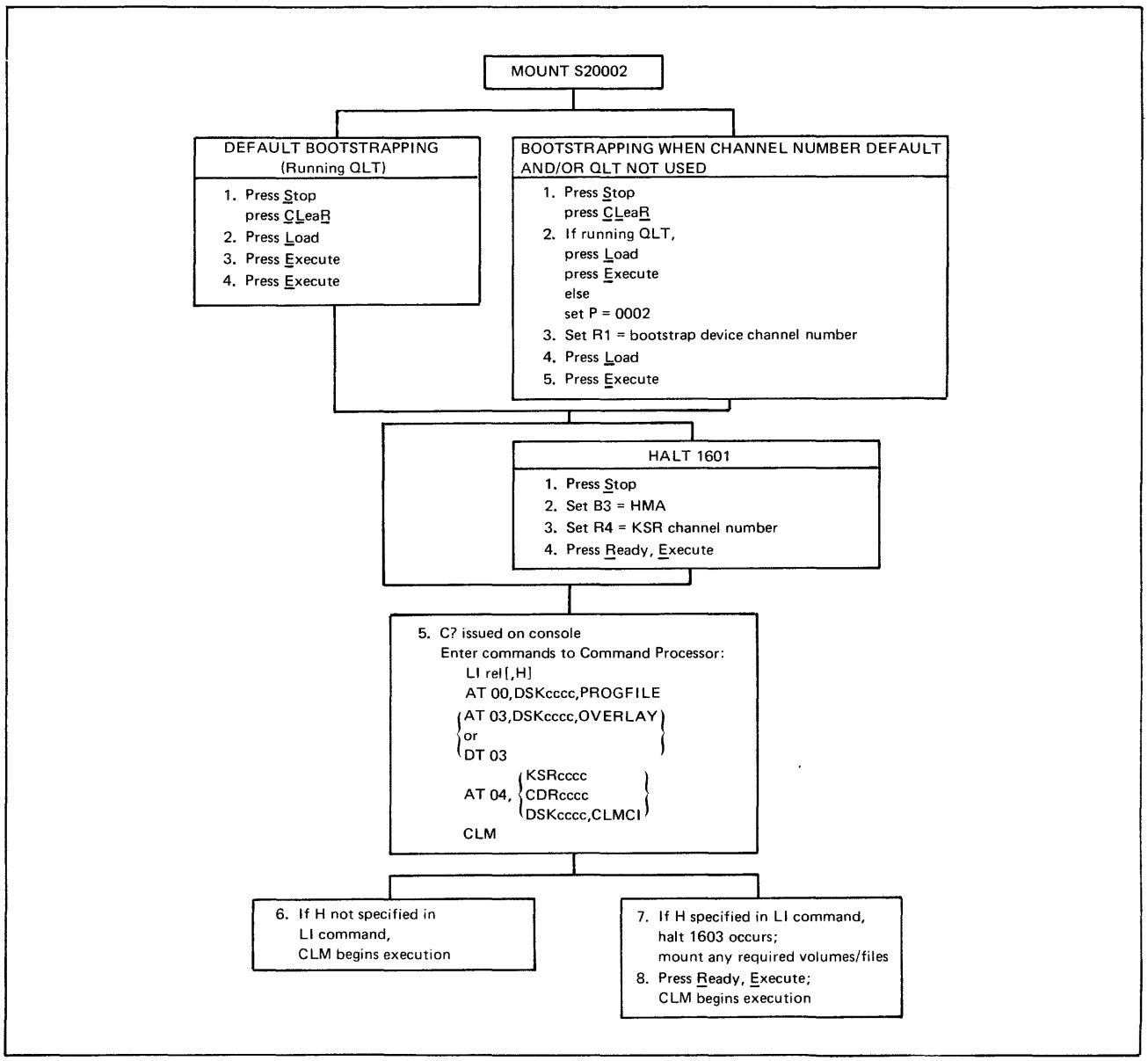


Figure IV-2-1. Procedure for Bootstrapping and Loading CLM From Disk Using Console

Mounting Required Disk Volume(s)

Before you bootstrap and load the CLM, mount the disk volume initially identified by the volume name S20002 either on a device whose channel number is 0400_{16} (the default bootstrap device channel number) or on another appropriate device to be used in bootstrapping. This volume includes the bootstrap record and boot loader and the file named PROGFILE that contains the Disk Loader, the

Offline Trap Handler, the Command Processor, and the CLM load modules, and device drivers. If any of the load modules required during the configuration phase (LACT command) is on a separate disk, mount that disk on the appropriate channel.

Bootstrapping Phase for CLM on Disk

Control panel operation causes the bootstrap record to be read into memory from the peripheral device on which it is stored; the bootstrap record then loads the boot loader. The LOAD indicator lamp goes on when the Load key is pressed, and will remain on until the bootstrapping routine jumps to the first location of the bootstrap record.

Perform one of the following two bootstrapping procedures as described in the next two subsections:

- Default bootstrapping: Use this procedure if you wish to perform the Quality Logic Test (QLT) and use the default bootstrap device channel number (0400_{16}). (The QLT clears memory; therefore, do not perform the QLT if you need to preserve the contents of memory.)
- Bootstrapping when channel number default and/or QLT not used: Perform this procedure if you wish to change the bootstrap device channel number and/or omit the QLT.

DEFAULT BOOTSTRAPPING

The default bootstrapping procedure is as follows:

1. Perform master clear operation, automatically setting the P-register to 0000_{16}.
2. Press Load.
3. Press Execute. The QLT will be run, setting register R1 to the default bootstrap device channel number and the P-register to 0002_{16}.
4. Press Execute, causing the bootstrap record to be loaded into memory. The bootstrap record loads the boot loader, which places the default high memory address in register B3 and the default operator's console channel number in register R4. The boot loader halts if the indicator for loader halt 1601 was set in the boot loader when it was created by Bootstrap Generator. At loader halt 1601, proceed to step 1 of the loading phase. If no halt occurs, the boot loader automatically loads the Disk Loader as in step 5 of the loading phase.

BOOTSTRAPPING WHEN CHANNEL NUMBER DEFAULT AND/OR QLT NOT USED

The bootstrapping procedure described below permits you to use a different bootstrap device channel number and/or omit the QLT.

1. Perform master clear operation.
2. If you are running the QLT, press Load and then Execute (the QLT sets the P-register to 0002_{16}); otherwise, set the P-register to 0002_{16}.
3. Set the bootstrap record device channel number in register R1 as follows:
   a. If you are using a channel number other than the default channel number, place the desired channel number in register R1.

b. If you are using the default channel number but are not running the QLT, place the default channel number 0400_{16} in register R1. (If you are using the default channel number and are running the QLT, the QLT places the default channel number in register R1.)

4. Press Load.

5. Press Execute, causing the bootstrap record to be loaded into memory. The bootstrap record loads the boot loader, which places the default high memory address in register B3 and the default operator's console channel number in register R4.

NOTE: Although the bootstrap record on a fixed cartridge disk cannot be bootstrapped directly into memory, this device can be used as the system load device by invoking the following procedure:

At step 3a above, enter in register R1 the cartridge disk unit channel number, specifying a 1 in the low-order byte (e.g., 0401_{16}). Pressing Load and Execute causes the bootstrap record and boot loader to be read from the removable cartridge disk. (All other programs, including the Disk Loader, Offline Trap Handler, and the program specified in the removable cartridge disk boot loader, will be searched for and loaded from the fixed cartridge disk.)

The boot loader halts if the indicator for loader halt 1601 was preset in the boot loader. At loader halt 1601, proceed to step 1 of the loading phase. If no halt occurs, the boot loader automatically loads the Disk Loader, as in step 5 of the loading phase.

Loading Phase for CLM From Disk Using Console

After you have completed one of the two alternative bootstrapping procedures described above, perform the following loading procedure:

1. If loader halt number 1601 occurs, press Stop, placing the central processor in the stop state.

2. Enter in register B3 the high memory address, unless you are using the default value.

3. Enter in register R4 the channel number of the KSR-like device you are using as the operator's console, unless you are using the default value.

4. If you have performed steps 1 through 3, press Ready, then Execute.

5. The Disk Loader is loaded into high memory. The Disk Loader loads the Offline Trap Handler, and then loads the Command Processor, which issues the system console message C? through the operator's console, indicating that it is ready to receive input.

   Enter commands to the Command Processor through the operator's console (or any other device specified as the command input device), to load the CLM. See Figure IV-2-2 for the commands to be submitted to the Command Processor before the CLM is loaded. You must specify a CLM relocation factor in the load initialize (LI) command. See the Release Bulletin for the value of the CLM relocation factor. The overlay file need be assigned only if overlaying is to be performed. Specify the halt option (H) in the load initialize command if all required volumes/files have not been made available (see "Making the Appropriate Load Modules Available for an Online Application" below).

6. If you do not specify the halt option (H) in the load initialize command to the Command Processor, the Disk Loader will load the CLM and turn control over to it.

7. If you specify the halt option (H) in the load initialize command to the Command Processor, the Disk Loader will load the CLM and then halt (halt number 1603 in register R1). Mount the appropriate volume or device required to satisfy the attach command to the Command Processor for the CLM command file (logical file number 04) in Figure IV-2-2. Make available on the appropriate channels any load modules required for configuring the online application that have not already been mounted (see "Making Appropriate Load Modules Available for Online Application" below).
8. Press Ready, then Execute. Control is turned over to the CLM.

Command Type	Command Format	Comments
Load Initialize	LI rel [,H]	rel - CLM relocation factor[a]
Attach (Program File)	AT 00,DSKcccc,PROGFILE	cccc - channel number of program file disk
Attach (Overlay File) or Detach (Overlay File)	AT 03,DSKcccc,OVERLAY DT 03	cccc - channel number of overlay file disk Required only if previously attached
Attach (Command File)	AT 04, {KSRcccc / CDRcccc / DSKcccc,CLMCI}	cccc - channel number of device containing command input file
Load	CLM	

[a]See the Release Bulletin for the value of the relocation factor.

Figure IV-2-2. Input to Command Processor Before Configuration Load Manager is Loaded

Making Appropriate Load Modules Available for Online Application

Before the CLM begins execution, you should mount the disk volume(s) containing the load modules to be loaded by the CLM; these load modules are the ones specified in the ADMOD, DEVICE, COMM, TTY, VIP and BSC commands to be submitted to the CLM for that particular online application. Certain load modules, if required, must be located in the file PROGFILE (that includes the CLM) on the disk on the same channel from which CLM was loaded; these modules are:

- Communications module ZGQCDS (CLM data structures/cleanup)
- DEVICE, COMM, TTY, VIP and BSC drivers (for which no additional ADMOD commands were issued)

All other load modules to be used in configuring the online application must be on the same type of device as is used for bootstrapping, but can be on devices on different channels.

The load modules are loaded by the CLM in the order in which they are specified in the ADMOD, DEVICE, COMM, TTY, VIP and BSC commands. See the Planning and Building an Online Application manual for a detailed description of the commands to the CLM.

BOOTSTRAPPING AND LOADING CLM FROM DISK USING CONTROL PANEL

The CLM can be loaded from a disk into memory using a control panel and the Disk Loader. The full control panel must be used if operator intervention is required during bootstrapping and loading. If no operator intervention is required, loading can be performed using either the full control panel, or, if

available, the basic control panel. (See Appendix F for a description of loading using the basic control panel.)

Bootstrapping and loading the CLM into memory using the control panel and the Disk Loader, involves:

- Invocation of the bootstrapping procedure which causes the boot loader to be loaded into memory
- Using the boot loader to load the Disk Loader which loads the Offline Trap Handler. The Disk Loader then enters into memory information required to load the CLM.
- Loading of the CLM by the Disk Loader which then turns control over to the CLM.

Parameter values preset in the bootstrap record and boot loader when they were created by Bootstrap Generator are used in loading the CLM from disk. The file name PROGFILE and member name CLM must be preset in the bootstrap record.

If loading of the CLM without operator intervention is to occur (i.e., none of the parameter values preset in the bootstrap record and boot loader are to be modified, and no halts are required), only the following actions are needed to load the CLM from disk using the full control panel:

- Mounting the required disk volume(s)
- Performing one of two possible bootstrapping procedures (as described below), causing the CLM to be loaded without additional operator intervention

(See the Planning and Building an Online Application manual for a description of the bootstrap record used in loading of the CLM from disk without operator intervention.)

If operator intervention is permitted during loading, the following parameter values that were stored in the bootstrap record and boot loader can be modified through the full control panel at loader halt 1601:

- High memory address
- Relocation factor
- Program loading device channel number
- Load and halt option (indicator setting for loader halt 1603)

The values preset in the bootstrap record and boot loader for these modifiable parameters are referred to as "default" values in the bootstrapping and loading procedures given below.

The complete operation for bootstrapping and loading the CLM from disk using the control panel, summarized in Figure IV-2-3, consists of these phases (as described in the following subsections):

- Mounting the required disk volume(s)
- Performing one of two possible bootstrapping procedures
- Performing loading of the CLM

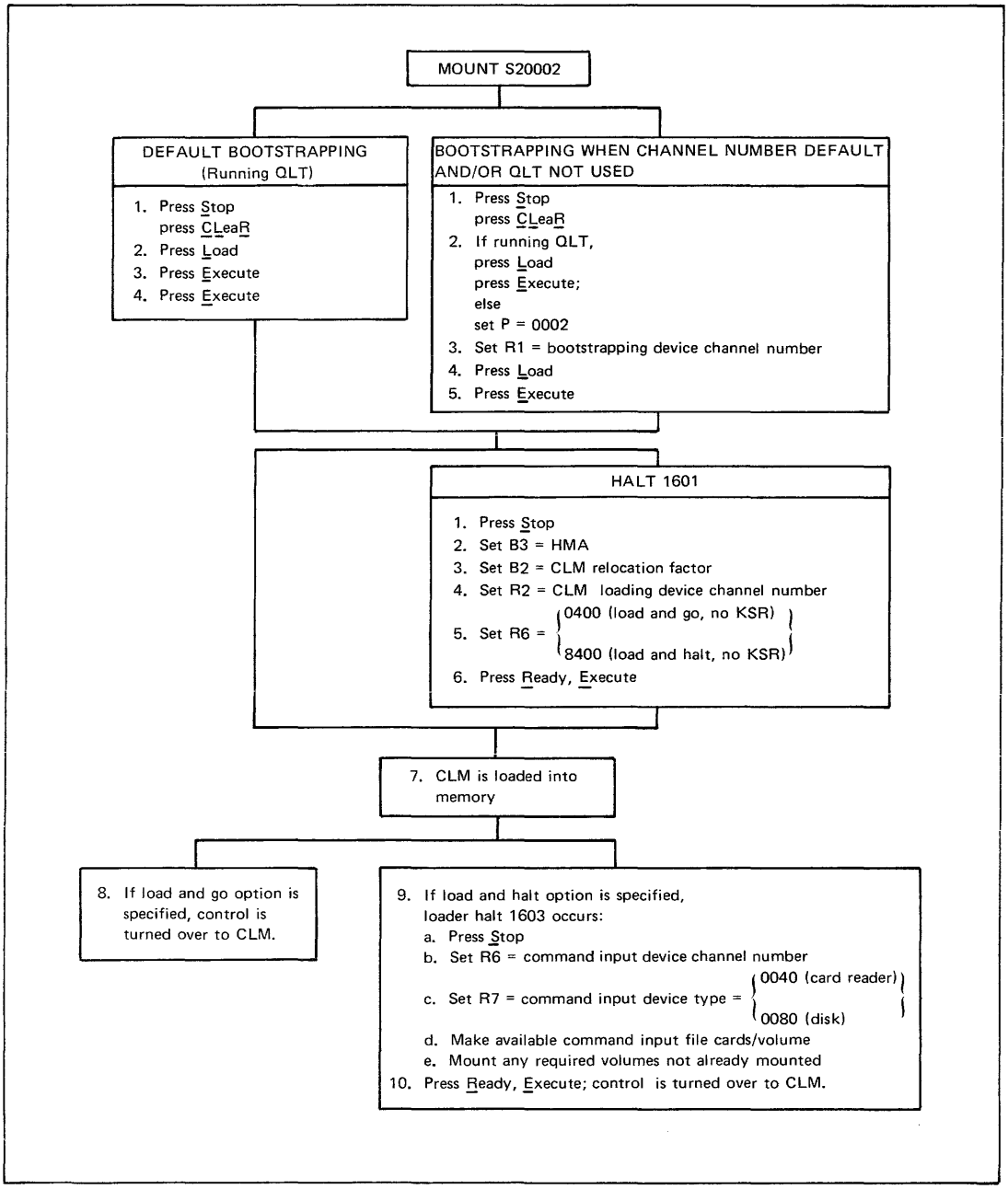


Figure IV-2-3. Procedure for Bootstrapping and Loading CLM From Disk Using Control Panel

Mounting Required Disk Volume(s)

Before you bootstrap and load the CLM, mount the disk volume initially identified by the volume name S20002 either on a device whose channel number is 0400_{16} (the default bootstrap device channel number) or on another appropriate device to be used in bootstrapping. This volume includes the bootstrap record and boot loader and the file named PROGFILE that contains the Disk Loader, the Offline Trap Handler, the Command Processor, and the CLM load modules and device drivers. If any of the load modules required during the configuration phase (LACT command) is on a separate disk, mount that disk on the appropriate channel. If no operator intervention is expected during CLM operation, the CLM command file and the overlay file (if needed) must be on the same disk as CLM.

Bootstrapping Phase for CLM on Disk

Control panel operation causes the bootstrap record to be read into memory from the peripheral device on which it is stored; the bootstrap record then loads the boot loader. The LOAD indicator lamp goes on when the Load key is pressed, and will remain on until the bootstrapping routine jumps to the first location of the bootstrap record.

Perform one of the following two bootstrapping procedures as described in the next two subsections:

- Default bootstrapping: Use this procedure if you wish to perform the Quality Logic Test (QLT) and use the default bootstrap device channel number (0400_{16}). (The QLT clears memory; therefore, do not perform the QLT if you need to preserve the contents of memory.)
- Bootstrapping when channel number default and/or QLT not used: Perform this procedure if you wish to change the bootstrap device channel number and/or omit the QLT.

DEFAULT BOOTSTRAPPING

The default bootstrapping procedure is as follows:

1. Perform master clear operation, automatically setting the P-register to 0000_{16}.
2. Press Load.
3. Press Execute. The QLT will be run, setting register R1 to the default bootstrap device channel number and the P-register to 0002_{16}.
4. Press Execute, causing the bootstrap record to be loaded into memory. The bootstrap record loads the boot loader, which places the default high memory address in register B3. The boot loader halts if loader halt 1601 was preset in the boot loader when it was created by Bootstrap Generator. At loader halt 1601, proceed to step 1 of the loading phase. If no halt occurs, the boot loader automatically loads the Disk Loader as in step 7 of the loading phase.

BOOTSTRAPPING WHEN CHANNEL NUMBER DEFAULT AND/OR QLT NOT USED

The bootstrapping procedure described below permits you to use a different bootstrap device channel number and/or omit the QLT.

1. Perform master clear operation.
2. If you are running the QLT, press Load and then Execute (the QLT sets the P-register to 0002_{16}); otherwise, set the P-register to 0002_{16}.
3. Set the bootstrap record device channel number in register R1, as follows:
   a. If you are using a channel number other than the default channel number, place the desired channel number in register R1.
   b. If you are using the default channel number but are not running the QLT, place the default channel number 0400_{16} in register R1. (If you are using the default channel number and are running the QLT, the QLT places the default channel number in register R1.)
4. Press Load.
5. Press Execute, causing the bootstrap record to be loaded into memory. The bootstrap record loads the boot loader, which places the default high memory address in register B3.

   NOTE: Although the bootstrap record on a fixed cartridge disk cannot be bootstrapped directly into memory, this device can be used as the system load device by invoking the following procedure: At step 3a above, enter in register R1 the cartridge disk unit channel number, specifying a 1 in the low-order byte (e.g., 0401_{16}). Pressing Load and Execute causes the bootstrap record and boot loader to be read from the removable cartridge disk. (All other programs, including the Disk Loader, Offline Trap Handler, and the program specified in the removable cartridge disk boot loader, will be searched for and loaded from the fixed cartridge disk.)

   The boot loader halts if the indicator for loader halt 1601 was preset in the boot loader. At loader halt 1601, proceed to step 1 of the loading phase. If no halt occurs, the boot loader automatically loads the Disk Loader, as in step 5 of the loading phase.

Loading Phase for CLM From Disk Using Control Panel

After you have completed one of the two alternative bootstrapping procedures described above, perform the following loading procedure:

1. If loader halt 1601 occurs, press Stop to place the central processor in the stop state.
2. Enter the high memory address in register B3, unless you are using the default value.
3. Enter in register B2 the relocation factor to be added to all relocatable addresses in the CLM, unless you wish to use the relocation factor that was set in the bootstrap record by the Bootstrap Generator. See the Release Bulletin for the value of the CLM relocation factor.
4. Enter in register R2 the channel number of the device from which the CLM is to be loaded unless you are using the default value.
5. Specify either the load and go or load and halt option unless you are using the default value. Enter one of the following bit masks for the loader indicator word in register R6:

   0400 - Load and go, no KSR

   8400 - Load and halt, no KSR

   The load and halt option permits you to mount the command input device on a different channel from the one being used to load the CLM (see step 9 below).

6. Press Ready, then Execute.
7. The boot loader loads the Disk Loader which loads the Offline Trap Handler and the CLM.
8. If the load and go option was specified, control is turned over to the CLM.
9. If the load and halt option was specified, the loader halts (halt number 1603 in register R1):
   a. Press Stop.
   b. Enter in register R6 the channel number of the device containing the command input file.
   c. Enter in register R7 the command input device type as follows:
      0040_{16} - Card reader
      0080_{16} - Disk
   d. If necessary, make the cards containing the command file available on the card reader, or mount the volume containing the command file, as appropriate.
   e. Make available on the appropriate channels any load modules required for configuring the online application that have not already been mounted (see "Making Appropriate Load Modules Available for Online Application" below).
10. Press Ready and then Execute. Control is turned over to the CLM.

Making Appropriate Load Modules Available for Online Application

Before the CLM begins execution, you should mount the disk volume(s) containing the load modules to be loaded by the CLM; these load modules are the ones specified in the ADMOD, DEVICE, COMM, TTY, VIP and BSC commands to be submitted to the CLM for that particular online application. Certain load modules, if required, must be in the file PROGFILE (that contains the CLM) on the device on the same channel from which CLM was loaded; these modules are:

- Communications module ZGQCDS (CLM data structures/cleanup)
- DEVICE, COMM, TTY, VIP and BSC drivers (for which no additional ADMOD commands were issued)

All other load modules to be used in configuring the online application must be on the same type of device as is used for bootstrapping, but can be on devices on different channels.

The load modules are loaded by the CLM in the order in which they are specified in the ADMOD, DEVICE, COMM, TTY, VIP and BSC commands. See Planning and Building an Online Application for a detailed description of the commands to the CLM.

SECTION 3

BOOTSTRAPPING AND LOADING ONLINE APPLICATIONS FROM PAPER TAPE

The procedures for bootstrapping and loading the Configuration Load Manager (CLM) from paper tape using the Paper Tape Loader and the control panel are described in this section. Once loaded, the CLM accepts commands from a command input stream that cause it to:

- Configure the online application
- Load the load modules that constitute the online application (the Executive modules, I/O drivers, and user-written application program)

Bootstrapping and loading the CLM from paper tape into memory, using the Paper Tape Loader, involves:

- Invocation of the bootstrapping procedure, which causes the bootstrap record to be loaded into memory
- Use of the bootstrap record to load the Paper Tape Loader, which loads the Offline Trap Handler
- Loading of the CLM by the Paper Tape Loader, which then turns control over to the CLM

If operator intervention is required during bootstrapping and loading from paper tape, the full control panel must be used. If no operator intervention is required, loading can be performed using either the full control panel, or, if available, the basic control panel. (See Appendix F for a description of loading using the basic control panel.)

Parameter values preset in the bootstrap record when it was created by Bootstrap Generator are used in loading the CLM from paper tape.

If the CLM is to be loaded without operator intervention (i.e., none of the parameter values preset in the bootstrap record are to be modified, and no halts are required), only the following actions are needed to load the CLM from paper tape using the full control panel:

- Mounting the required load modules on the ASR
- Performing one of two possible bootstrapping procedures (as described below), causing the CLM to be loaded without additional operator intervention.

(See the Planning and Building an Online Application manual for a description of the bootstrap record used in loading the CLM from paper tape without operator intervention.)

If operator intervention is permitted during loading, the following parameter values stored in the bootstrap record can be modified through the control panel during loading if loader halt 1601 occurs:

- High memory address
- Operator's console channel number

If loader halt 1602 occurs, these parameter values can be modified through the control panel:

- Relocation factor
- Load and halt option (indicator setting for loader halt 1603)

The values that were preset in the bootstrap record for these modifiable parameters are referred to as "default" values in the bootstrapping and loading procedures given below.

The complete operation for bootstrapping and loading the CLM from paper tape, summarized in Figure IV-3-1, consists of these phases (as described in the following subsections):

- Making the required load modules available on the paper tape reader (ASR)
- Performing one of two possible bootstrapping procedures
- Performing loading of the CLM

MAKING LOAD MODULES REQUIRED FOR ONLINE APPLICATION AVAILABLE ON PAPER TAPE

Before you perform the bootstrapping and loading procedure, you must make the information required for the online application available on a paper tape reader (ASR) connected either to channel number 0400_{16} (the default channel number for bootstrapping) or to another appropriate channel number to be used in bootstrapping. The paper tapes must appear in the following order in the reader:

1. Bootstrap record followed by Paper Tape Loader and Offline Trap Handler
2. Configuration Load Manager (CLM) and CLM modules (in the order specified in the LACT commands)
3. The system load modules and online application program load modules (in the order specified in the ADMOD, DEVICE, COMM, TTY, VIP and BSC commands)

You must specify the CLM and system load modules that you wish to use in your online application in the ADMOD, DEVICE, COMM, TTY, VIP, and BSC commands you submit to the CLM. All load modules used in configuring the online application must be on the same type of device as is used in bootstrapping.

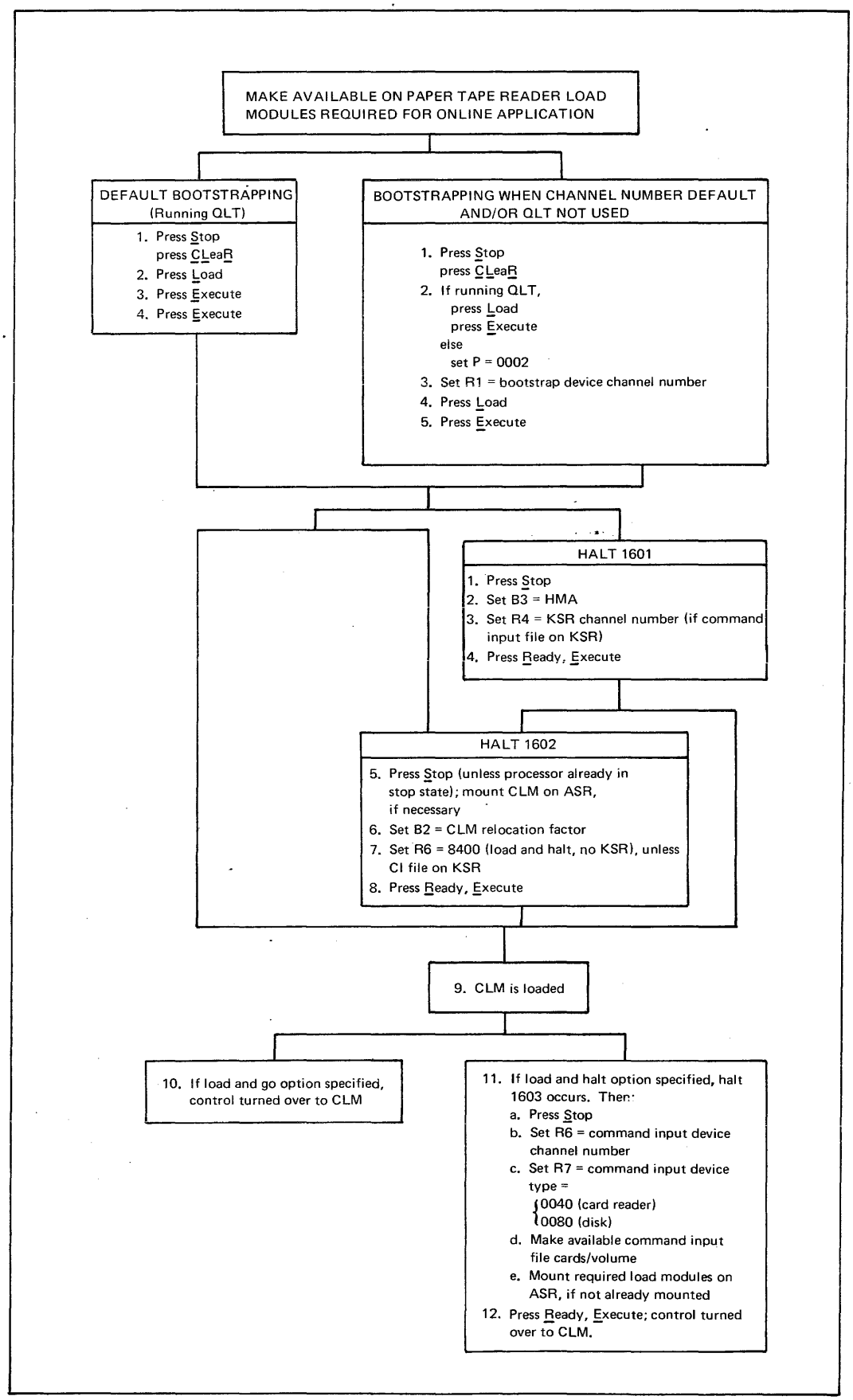


Figure IV-3-1. Procedure for Bootstrapping and Loading CLM From Paper Tape Using Control Panel

Because the paper tape reader is a serial input device, you must mount the CLM and system load modules and application program load module in the order in which the CLM is to load them. The CLM loads load modules in the order in which they are specified in the ADMOD, DEVICE, COMM, TTY, VIP, and BSC commands. Refer to "Order of Loading CLM Modules" below for the order in which CLM modules are loaded.

BOOTSTRAPPING PHASE FOR CLM ON PAPER TAPE

Control panel operation causes the bootstrap record to be read into memory from the peripheral device on which it is stored. The LOAD indicator lamp goes on when the Load key is pressed, and will remain on until the bootstrapping routine jumps to the first location of the bootstrap record.

Perform one of the following two bootstrapping procedures as described in the next two subsections:

- Default bootstrapping: Use this procedure if you wish to perform the Quality Logic Test (QLT) and use the default bootstrap device channel number (0400_{16}). (The QLT clears memory; therefore, do not perform the QLT if you need to preserve the contents of memory.)
- Bootstrapping when channel number default and/or QLT not used: Perform this procedure if you wish to change the bootstrap device channel number and/or omit the QLT.

Default Bootstrapping

The default bootstrapping procedure is as follows:

1. Perform master clear operation, automatically setting the P-register to 0000_{16}.
2. Press Load.
3. Press Execute. The QLT will be run, setting register R1 to the default bootstrap device channel number and the P-register to 0002_{16}.
4. Press Execute, causing the bootstrap record to be loaded into memory. The bootstrap record places the default high memory address in register B3 and the default operator's console channel number in register R4. The bootstrap record automatically loads the Paper Tape Loader, which in turn loads the Offline Trap Handler, unless loader halt 1601 occurs. If halt 1601 occurs, proceed to step 1 of the loading phase. If halt 1602 occurs, proceed to step 5. If no halt occurs, proceed to step 9. (Loader halt numbers are stored in register R1.)

Bootstrapping When Channel Number Default and/or QLT Not Used

The bootstrapping procedure described below permits you to use a different bootstrap device channel number and/or omit the QLT.

1. Perform master clear operation.
2. If you are running the QLT, press Load and then Execute (the QLT sets the P-register to 0002_{16}); otherwise, set the P-register to 0002_{16}.

3. Set the bootstrap record device channel number in register R1, as follows:

   a. If you are using a channel number other than the default channel number, place the desired channel number in register R1.

   b. If you are using the default channel number but are not running the QLT, place the default channel number 0400_{16} in register R1. (If you are using the default channel number and are running the QLT, the QLT places the default channel number in register R1.)

4. Press Load.

5. Press Execute, causing the bootstrap record to be loaded into memory. The bootstrap record places the default high memory address in register B3 and the default operator's console channel number in register R4. The bootstrap record automatically loads the Paper Tape Loader, which in turn loads the Offline Trap Handler, unless loader halt 1601 occurs. If halt 1601 occurs, proceed to step 1 of the loading phase. If halt 1602 occurs, proceed to step 5. If no halt occurs, proceed to step 9. (Loader halt numbers are stored in register R1.)

LOADING PHASE FOR CLM FROM PAPER TAPE

After you have completed one of the two alternative bootstrapping procedures described above, perform the following loading procedure:

1. If loader halt 1601 occurs, press Stop, placing the central processor in the stop state; perform steps 2 through 4.

2. Enter in register B3 the high memory address, unless you are using the default value.

3. If you are using the KSR as an operator's console, enter in register R4 the KSR channel number, unless you are using the default value.

4. If you have performed steps 1 through 3, press Ready, then Execute, causing the Paper Tape Loader and Offline Trap Handler to be loaded into high memory.

5. If loader halt 1602 occurs, press Stop, unless the processor is already in the stop state (see step 1 above). Mount the CLM on the ASR, if it is not already mounted. Perform steps 6 through 9.

6. Enter in register B2 the relocation factor to be added to all relocatable addresses in the program being loaded, unless you wish to use the default value preset in the bootstrap record. See the Release Bulletin for the value of the CLM relocation factor.

7. If you are not using a KSR-like device as an operator's console for the submission of commands to the CLM, specify the load and halt option by entering the following bit mask for the loader indicator word in register R6:

   8400_{16} - Load and halt, no KSR

   The load and halt option permits you to specify the command input file device type and channel number after the CLM has been loaded (see step 12 below).

8. If you have performed steps 5 through 7, press Ready, then Execute.

9. The Paper Tape Loader loads the CLM, which is the next load module on the paper tape reader.
10. If the load and go indicator is set, the loader turns control over to the CLM.
11. If the load and halt indicator is set, the loader halts (halt 1603 in register R1). Then:
    a. Press Stop
    b. Enter in register R6 the channel number of the device containing the command input file
    c. Enter in register R7 the command input device type as follows:
       0040_{16} - Card reader
       0080_{16} - Disk
    d. Make the cards containing the command input file available on the card reader, or mount the volume containing the command input file, as appropriate
    e. Mount on the paper tape reader the load modules required for the online application, if they are not already mounted
12. Press Ready, then Execute. Control is transferred to the CLM, which loads the remaining load modules required to create the online application.

If all load modules are not on the same paper tape, continue to mount tapes and restart the paper tape reader manually when the end of tape of the previous load module causes the reader to stop.

ORDER OF LOADING CLM MODULES

The order in which CLM modules are loaded depends on whether the high memory address (HMA) is 1FFF (8K) or greater. The two cases are described below. In the following description, EXTEN designates a CLM module you can design to interpret commands that are specific to your application.

Loading Order When HMA Is 1FFF

The order of loading CLM modules when HMA is 1FFF is given in Figure IV-3-2. In this figure, the CLM modules following D$EXTENn are loaded only as needed and in the order specified in the LACT commands. Therefore, commands to the CLM must be issued in groups (as in Table IV-3-1) in the order of the LACT commands. If HMA is 1FFF, the C$ load modules are loaded in order after the QUIT command is issued.

Loading Order When HMA Is Greater Than 1FFF

The order of loading CLM modules when HMA is greater than 1FFF is given in Figure IV-3-3. All the CLM load modules are loaded before the CLM requests any system configuration commands; therefore, system commands need not be grouped.

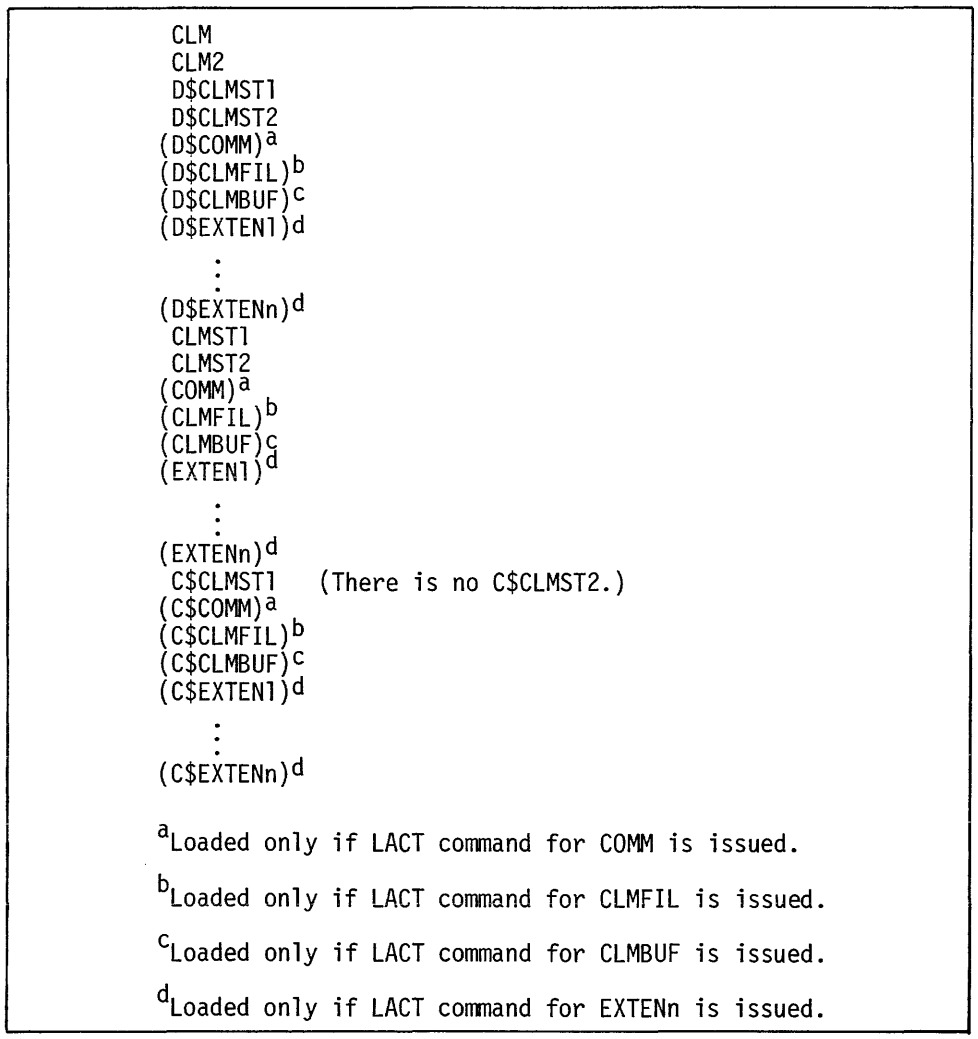


Figure IV-3-2. Order of Loading CLM Modules When HMA Is 1FFF

Table IV-3-1. CLM System Commands Grouped By Interpretive Module

Interpretive Module	CLM Commands	Interpretive Module	CLM Commands
CLMST1	SYS	CLMST2	IOS
	OIM		EQLRN
	TSA		ATLRN
	CLOCK		
	DATE	CLMBUF[a]	BUFSPACE
	TASK		
	DEVICE	COMM[a]	COMM
	TRAP		TTY
			VIP
CLMFIL[a]	FILMGR		BSC
	DEVFILE		MODEM
	ATFILE		LTPDEF
	FMDISK		LTP0
			LTP1
CLMST2	ADMOD		LTP2
	PRMOD		LTP3
	ELOC		STATION
	EVAL		

[a]Name specified in LACT command.

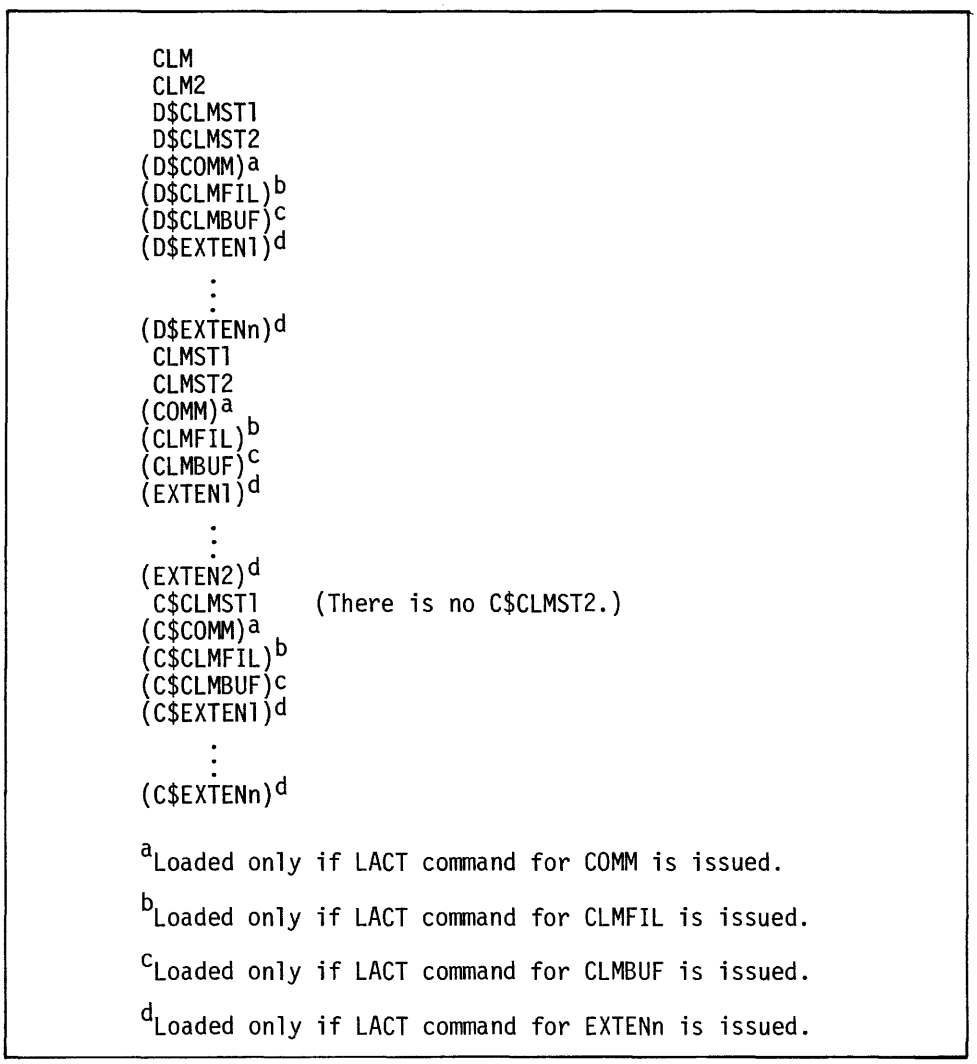


Figure IV-3-3. Order of Loading CLM Modules When HMA Is Greater Than 1FFF

SECTION 4

ERROR REPORTING BY CONFIGURATION LOAD MANAGER

The Configuration Load Manager (CLM) detects command errors and load erros. Messages issued by the CLM are described in Table IV-4-1. Each error code that begins with CFGLD is a command error; all other errors are load errors.

If a command error occurs and an operator's console is available, an error message is issued through the console. If a device other than the operator's console is the command input device, the error message is issued through the operator's console and is preceded by a typeout of the command that caused the error.

If a command error occurs and no operator's console is available, register R1 contains the error code, and register R2 may contain additional information.

If a load error occurs, the system halts and does not cause a typeout of an error code. For each load error, register R1 contains the error code, and register R2 may contain additional information. Register B4 contains a pointer to the high memory address of the loader. HMA-6 and HMA-3 contains the name of the module to be loaded, and HMA-12 to HMA-7 contains the name of the file in which the module is located.

After determining the error code, determine the cause of the error using Table IV-4-1.

Symbols and abbreviations used in CLM error messages include:

Δ - Space

n - Number of operand that contains error (consists of four hexadecimal digits).

sswd - Software status word (consists of four hexadedimal digits)

The software status word is described in the Executive and Input/Output manual.

Table IV-4-1. Error Messages Issued By Configuration Load Manager

Message		Condition
Code	Other Information	
0106		Operator's console or another KSR-like device has been waiting approximately 4 minutes for input. System halts. To continue, press Ready and then press Execute.
CFGLDΔ1301		Command mnemonic was misspelled, or it does not begin in the first position of the line.
CFGLDΔ1302	n	Nondecimal digit specified in nth operand, or, if a KSR-like device is the command input device, a command line contained more than 40 characters.
CFGLDΔ1303	n	Decimal number specified in nth operand is too large; must be ≤ 65529.
CFGLDΔ1304	n	Nonhexadecimal digit specified in nth operand.
CFGLDΔ1305	n	Hexadecimal number in nth operand is too large; single-word (X-type) hexadecimal number contains more than four digits, or double-word (D-type) hexadecimal number contains more than eight digitis.
CFGLDΔ1306	n	ASCII character string in nth operand is too long; the end of the line has been reached, or more than 64 characters were specified without specifying a string termination character (a single quote).
1307		Command cannot be read. System halts. Determine whether the command input device is operational. To retry, press Ready and then press Execute.
1309		Fatal error; system halts.
130A	sswd	Command input device cannot be opened. System halts. The command input device type probably is incorrect.
130B		Operator's console not operational so error code cannot be issued through it. System halts. Register R2 contains the error code that designates the cause of the error. Reload the Configuration Load Manager and retry.
130C	sswd	File that contains overlay cannot be opened. System halts. Register R2 contains error code: 0 indicates that AT 03 entry missing from attach table; any other value indicates a disk error.
CFGLDΔ130F		Either one or more operands are missing, or an operand was specified incorrectly. An operand must comprise strings of ASCII characters, or decimal or hexadecimal integers.

Table IV-4-1 (cont). Error Messages Issued By Configuration Load Manager

Message		Condition
Code	Other Information	
CFGLDΔ1310		SYS command error. Highest logical resource number too large; value specified in hilrn operand must be ≤ 255.
CFGLDΔ1311		SYS command error. Lowest priority level too large; value specified in lolevel operand must be ≤ 62.
CFGLDΔ1312		SYS command error. Lowest priority level too small; value specified in lolevel operand must be ≥ 6.
CFGLDΔ1313		SYS command error. Value other than SAF designated as model.
CFGLDΔ1314		SYS command error. High memory address too large; address designated in himem operand is higher than ending address of the loader.
CFGLDΔ1315		SYS command error. himem operand specified incorrectly; must be a double-word hexadecimal integer.
CFGLDΔ1316		SYS command error. SYS must precede all CLM commands except for IOS, LACT, and ELACT.
CFGLDΔ1318		TRAP command error. Invalid trap vector number specified in trap number operand; must be ≤46.
CFGLDΔ1319		CLOCK command error. Invalid line frequency specified in hz operand; must be 50 to 60.
CFGLDΔ131A		CLOCK command error. Invalid value specified in scan-cycle operand; must be 8, 16, 25, 33, 50, or 100.
[CFGLDΔ]131F		Multiply-defined symbol; it is either a load error or an ELOC or EVAL command error. If a load error occurs, the system halts. To ignore the error, press Ready, then Execute.
CFGLDΔ1320		ELOC command error. Address specified incorrectly in absolute-address operand; must be a double-word hexadecimal integer.
1321		Configuration work area overlap; data structures require more memory than is available. System halts. Reallocate memory so that more memory is available for data structures, or use fewer data structures.
1323		Block size of zero was requested. System halts. Register B5 contains location of the request.

Table IV-4-1 (cont). Error Messages Issued By Configuration Load Manager

Message		Condition
Code	Other Information	
CFGLDΔ1324		DEVICE command error. Invalid value specified in device-type operand; must be one of the following: KSR, ASR, LPT, SPT, CDR, DSK, FCD, or RCD.
CFGLDΔ1325		Logical resource number too high; value specified in lrn operand of DEVICE, STATION, TASK, ATLRN, EQLRN, TTY, VIP, BSC, or LTPn command exceeds value specified in hilrn operand of SYS command.
CFGLDΔ1326		Level too large. Value specified in level operand of DEVICE, TASK, ATLRN, TTY, VIP, BSC, or LTPn command exceeds value specified in lolevel operand of SYS command.
CFGLDΔ1327		Level too small. Value specified in level operand of DEVICE, TASK, ATLRN, TTY, VIP, BSC, or LTPn command is less than 5.
CFGLDΔ1328		Duplicate logical resource number. Value specified in lrn operand of DEVICE, STATION, TASK, ATLRN, EQLRN, TTY, VIP, BSC, or LTPn command has already been assigned to another device of a different type.
CFGLDΔ1329		Duplicate level. Level specified in DEVICE, TASK, ATLRN, TTY, VIP, BSC, or LTPn command was assigned in a previous command.
CFGLDΔ132A		DEVICE command error. Duplicate channel; the channel specified has already been assigned to another device in a TTY, VIP, BSC, DEVICE, or LTPn command.
CFGLDΔ132B		Priority level specified in OIM, DEVICE, TASK, ATLRN, TTY, VIP, BSC, or LTPn command has been reserved for exclusive use by another task.
CFGLDΔ132C		DEVICE command error. Reference logical resource number does not match logical resource number specified in previous DEVICE command with the same level and channel number.
CFGLDΔ132D		The reference logical resource number (designated in label operand) matches a logical resource number in previous DEVICE command that also includes a label.
132E		System halts after configuration loading is completed so that desired changes can be made; this halt occurs only if QUIT HLT was specified. To execute the loaded program(s), press Ready and then press Execute.
CFGLDΔ132F		Command error. A file name, module name, or label string is too long.

Table IV-4-1 (cont). Error Messages Issued By Configuration Load Manager

Message		Condition
Code	Other Information	
CFGLDΔ1330		BUFSPACE command error; system halts. More than 32K words of memory requested for blocks.
CFGLDΔ1331		DEVICE command error. DEVICE command that contains reference logical resource number is not preceded by a DEVICE command including that logical resource number.
CFGLDΔ1332		EQLRN command error. The old logical resource number was not assigned in a previous TASK, DEVICE, ATLRN, EQLRN, TTY, VIP, BSC, LTPn, or STATION command.
1333		Load error. Overlays cannot be loaded because a disk device was not specified in a DEVICE command. System halts. To ignore the message (and not load overlays), press Ready, then Execute.
1334		Load error. Wrong overlay device channel number specified; overlays can reside only on disk devices. To ignore the message (and not load overlays), press Ready, then Execute.
1335		Load error. System encountered unresolved symbol that was defined using P-relative addressing. System halts.
1340		Load error; requested module cannot be located. System halts. Verify that the correct module name and file name were designated. (File name is not applicable if loading from cards or paper tape.) HMA-12 to HMA-7 contain the file name; HMA-6 to HMA-3 contain the module name. Make the necessary correction(s). The procedure for resuming loading is determined by the medium from which the module is being loaded, and whether it was necessary to correct the file and/or module names. Medium / Module and File Names Were Correct / Module or File Name Was Incorrect Cards / Cards for a load module are out of order. Place the cards in the correct order, precede them with an EOF card, and put them back into card reader. Press Ready and then press Execute. / Precede card just read (should be sequence number 1) with an EOF card, and put those cards back into card reader. Press Ready and then press Execute.

Table IV-4-1 (cont). Error Messages Issued By Configuration Load Manager

Message		Condition		
Code	Other Information			
		Medium	Module and File Names Were Correct	Module or File Name Was Incorrect
		Paper Tape	Position tape to beginning of load module. Press Ready and then press Execute.	Position tape to beginning of load module. Press Ready and then press Execute.
		Disk	Mount disk volume that contains load module to be loaded into memory. Press Ready and then press Execute.	Press Ready and then press Execute.
1341		Load error; unresolved symbol(s). System halts. This error message is issued each time an undefined symbol is encountered and after loading is completed. To ignore the message, press Ready and then press Execute. If QUIT was specified, execution begins after loading is completed. If QUIT HLT was specified, after loading is completed the error code 132E is issued and a halt occurs.		
1342		Load error. Residual main memory not large enough to include buffer(s) defined in BUFSPACE command(s). System halts.		
CFGLDΔ1343		FILMGR command error. FILMGR command was issued after a DEVFILE, FMDISK, or ATFILE command.		
CFGLDΔ1344		FILMGR command error. Value specified in max-lfn operand exceeds 255.		
CFGLDΔ1345		DEVFILE command error. Invalid value specified in device-name operand; must be one of the following: KSR, ASR, TTR, TTP, LPT, SPT, CDR, KSI, KSO, ASI, ASO, TTY, TTYI, TTYO, VIP, VIPI, VIPO, BSC, BCI, BCO.		
CFGLDΔ1346		DEVFILE command error. Invalid value specified in double operand.		
CFGLDΔ1347		DEVFILE command error. Invalid value specified in share operand.		
CFGLDΔ1348		FMDISK command error. Logical resource number too high; value specified in lrn operand exceeds value in hilrn operand of SYS command.		
CFGLDΔ1349		FMDISK command error. Invalid value specified for disk-type operand.		

Table IV-4-1 (cont). Error Messages Issued By Configuration Load Manager

Message Code	Message Other Information	Condition
CFGLDΔ134A		ATFILE command error. Logical file number too high; value specified in lfn operand exceeds value specified in max-lfn operand of FILMGR command.
134B		Current load module's initialization code detected an error; load module cannot be initialized. Register R2 contains error identifier. Correct the error and then reload the system. Contents of R2 — Meaning 1301 — OIM command missing. 8005 — Requested MLCP is unavailable. 8006 — MLCP time-out. 8007 — MLCP hardware error. 8008 — User-written LTP has called the CLM communication initialization subroutines. $R1 contains an invalid function number; must be from 0 through 4. 800B — There is not enough room in the MLCP RAM to load the requested module. 800C — Line adapter type id does not match the protocol configured on that line; e.g., an asynchronous device was configured on a synchronous communications line adapter. 8040 — The TTY line-type processor was loaded into memory, but no TTY's were configured. 8050 — A VIP was configured as a full duplex device.
CFGLDΔ134C		BUFSPACE command error. No value specified for number element of "size, number" operand.
CFGLDΔ134D		BUFSPACE command error. No value specified for size operand.
CFGLDΔ134E		FILMGR command error. Value specified in concurrent calls operand cannot be 0.
CFGLDΔ134F		FILMGR command error. Value specified in concurrent opens operand cannot be 0.

Table IV-4-1 (cont). Error Messages Issued By Configuration Load Manager

Message		Condition
Code	Other Information	
CFGLDΔ1350		DEVFILE command error. Logical resource number (lrn) not specified in previous DEVICE, TTY, VIP, BSC, LTPn, or STATION command.
CFGLDΔ1351		DEVFILE command error. Duplicate logical resource number (lrn); lrn already designated in a TASK command.
CFGLDΔ1352		DEVFILE command error. Incorrect logical resource number (lrn) specified for device name. To determine correct lrn, check corresponding DEVICE, TTY, VIP, BSC, LTPn, or STATION command.
CFGLDΔ1353		BUFSPACE command error. "Size, number" operand pairs omitted.
CFGLDΔ1354		BUFSPACE command error. Duplicate name specified in space-name operand.
CFGLD1355		DATE command error. Invalid operand specified; either a nonnumeric character was entered, or the incorrect number of characters were entered.
CFGLD1356		DEVFILE command error. Logical resource number (lrn) too large; must be equal or less than the value specified in hilrn operand of SYS command.
CFGLDΔ1357		LACT command error. Extension module size greater than 1K words.
1358		Load error. Overlay load module contains initialization code; this is not permitted. To ignore the message, press Ready, then Execute.
1359	sswd	Load error. An error occurred while trying to write an overlay on a disk. System halts. Register R2 contains the status. 000F indicates that the file is too small to contain the overlay; reallocate the file so it is larger. Any other value indicates disk error.
135A		Load error. The last two characters of an overlay name are not decimal digits. System halts.
135B		Load error. Overlay contains an undefined symbol. System halts.

Table IV-4-1 (cont). Error Messages Issued By Configuration Load Manager

Message		Condition
Code	Other Information	
CFGLDΔ1380	sswd	Communication configuration table command error. Register R2 contains the status, which is described below: Contents of R2 — Meaning 1 — COMM command was specified more than once. 2 — Interrupt priority level specified in the COMM command is not in the range permitted or was specified in a previous command. 3 — A CLM communication command was issued before the COMM command. The COMM command must be issued and must precede all of the following commands: TTY, MODEM VIP, LTPDEF BSC, LTPn STATION 4 — Invalid lrn specified in a communication device command. 5 — Invalid priority level specified in a communication device command. 6 — Priority level specified in a communication device command is out of the allowable range or was previously specified in a DEVICE, TASK, COMM, or OIM command. 7 — Invalid channel number specified in a communication device command. 8 — Modem type number specified in a MODEM, TTY, VIP, BCS, or LTP (0-3) command is less than 0 or greater than 15, or the modem type number specified in a communication device command (TTY, VIP, BSC, or LTP (0-3)) is not 0, 1, 2, or previously defined by a MODEM command. 9 — Speed operand of a TTY or LTP (0-3) command is not one of the supported speeds.

Table IV-4-1 (cont). Error Messages Issued By Configuration Load Manager

Message		Condition
Code	Other Information	
		Contents of R2 — Meaning A — Character set operand of a BSC command is not ASCII, EBSCID, or TEBCDIC. B — Duplex operand of a LTP (0-3) command is not FDX or HDX.
CFGLDΔ1381 • • • CFGLDΔ13FF	sswd	User extension errors.

SECTION 5
ERROR REPORTING BY ONLINE APPLICATIONS

Each coded error message issued through the operator's console has the format xxyyzz, where

xx - Component code
yy - Error category
zz - Specific error code

These elements are described in more detail in Appendix G.

If the operator's console is not available, error messages may be issued through registers R1 and R3. The contents of these registers correspond to the error code format listed above, as follows:

R1 - yyzz
R3 - 00xx

Some error messages contain the following addtional information:

levl - Priority level (two decimal digits)
lfn - Logical file number (two decimal digits)
cccc - Channel number (four hexadecimal digits)
sswd - Software status word (four hexadecimal digits)
dswd - Device specific word (four hexadecimal digits)

If the message is being issued through registers, see Appendix C of the Executive and Input/Output manual to determine the registers through which this additional information is issued.

The software status word and the device specific word are described in the Executive and Input/Output manual.

After determining the error code, determine the cause of the error by looking up the error code in the appropriate online error code table. There is one set of tables for interpreting online error messages, regardless of whether messages are issued through the operator's console or through registers. No operator action is indicated in any of the tables in this section; error messages should be given to the programmer. Except where specified otherwise in the following paragraphs, online error messages are issued through the operator's console, if available; otherwise, they are issued through registers.

FORTRAN error messages are described in Tables IV-5-1 and IV-5-2; each code begins with component number 15. Table IV-5-1 describes codes issued by FORTRAN run-time I/O routines. These messages are not issued if the optional error specified (ERR=) was specified in an I/O statement. Table IV-5-2 describes codes issued by FORTRAN intrinsic functions.

COBOL error messages issued at run-time are described in Table IV-5-3.

File Manager coded error messages are described in Table IV-5-4. Uncoded error messages issued by the File Manager are described in Table IV-5-5. File Manager error messages are visible to the operator only if the application program calls the File Manager entry of the Executive error reporter.

Physical I/O error messages are described in Table IV-5-6. The first two digits (the component number) of each physical I/O error code are variable; component number 22 designates that the File Manager is reporting the error; component number 80 through FF designates which user component is reporting the error.

NOTES:
1. Since the File Manager (designated by component number 22) may report a physical I/O error or the application may report a file management error, the value of yy (the error category) designates whether the error code is in Table IV-5-4 or IV-5-6. If component number 22 is followed by 01, refer to the physical I/O table (Table IV-5-6); if the error category code is 02, refer to the File Manager table (Table IV-5-4). Physical I/O error messages beginning with 2201 are always reported by the File Manager.
2. Error codes for errors detected by user components should begin with component number 80 through FF, followed by 0F and the specific error code (see Table IV-5-6). It is recommended that user components follow this convention for reporting errors.

Overlay Loader error messages are described in Table IV-5-7. These messages are issued only through registers.

Operator Interface Manager error messages are described in Table IV-5-8.

Online Debugger error messages are described in Table IV-5-9.

Table IV-5-1. Coded Error Messages Issued By FORTRAN Run-Time Input/Output Routines

Message	Condition
150201	Illegal file name.
150202	Concurrent open commands exceed number of save areas.
150203	Illegal operation.
150204	File is in busy state.
150205	Illegal parameter.
150206	Illegal logical file number.
150207	File not opened.
150208	File already opened.
150209	File not located in referenced volume.
15020A	File incorrectly positioned for requested operation.
15020B	Hardware error generated from referenced device.
15020C	Requested volume not present.
15020D	Maximum record limit exceeded.
150F01	Record length exceeds available buffer space.
150F02	Record length specified in FORTRAN OPEN statement does not match actual physical record length.
150F03	End of file reached, but no end path is specified in FORTRAN program.
150F04	Record type conflict: formatted versus unformatted.
150F05	Invalid count parameter specified in BACKSPACE statement.
150F06	Input/output list (iolist) demands exceed record length.
150F07	Media error during previous I/O request in which specified unit number was requested.
150F08	Unformatted WRITE without any iolist.
150F21	Illegal format character.
150F22	Illegal format character sequence.
150F23	Unequal number of matching parentheses.
150F24	Integer constant missing from Hollerith type descriptor.
150F25	Input of Hollerith type data is illegal.
150F26	Input of apostrophe type data is illegal.
150F27	Invalid data for E or F format descriptor.
150F28	Integer constant missing for X-field descriptor.
150F29	Data type does not correspond with indicated data.
150F2A	Integer width is 0.
150F2B	Logical field contains blank characters.
150F2C	Logical field is not either true or false.
150F2D	Integer or real data value too large.
150F2E	Illegal value for exponent.
150F2F	Integer value specified in FORMAT statement is too large.
150F30	Too many embedded parentheses.

Table IV-5-2. Coded Error Messages Issued By FORTRAN Intrinisic Functions

Message	Condition
150F41	Value exceeds maximum ranges for EXPonent.
150F42	Zero (0) argument for intrinsic function ALOG.
150F43	Negative argument for intrinsic function ALOG.
150F44	Argument too large for intrinsic function SIN.
150F45	Negative argument for intrinsic function SQRT.
150F46	Origin cannot be an argument for ATAN2.
150F47	The second argument for the intrinsic function MOD cannot be zero.
150F48	The second argument for the intrinsic function AMOD cannot be zero.
150F49	The second argument for the intrinsic function ISION cannot be zero.
150F4A	The second argument for the intrinsic function SIGN cannot be zero.

Table IV-5-3. Error Message Issued By COBOL AT Run Time

Message	Condition
CONV ERR ON SOURCE LINE nnnnn	The result of converting a numeric field to binary is either negative or exceeds 32,767. The condition generally occurs during conversion of an identifier used in the following: (1) a PERFORM statement, (2) a SET statement, (3) subscripting, and (4) relative I/O statement (converting the RELATIVE KEY).

Table IV-5-4. Coded Error Messages Issued By File Manager

Message		Condition
Code	Other Information	
xx0000		No error.
xx0201	levl lfn	Invalid path name specified in attach file (ATFILE) command. Path name includes invalid characters or the incorrect number of characters. Correct application code.
xx0202	levl lfn	Too many files opened. Number of files opened cannot exceed value specified in concurrent opens operand of file manager (FILMGR) command. (1) Program may retry later, and/or (2) operator can respecify FILMGR command and reload.
xx0203	levl lfn	Function error. Nonexistent function specified, or function not valid for type of file referenced.
xx0204	levl lfn	File busy. File specified to be opened is nonshareable and already opened. Program should retry after the file is closed.
xx0205	levl lfn	Invalid operand other than lfn or function. Correct application code.
xx0206	levl lfn	Logical file number too large. Correct application code.
xx0207	levl lfn	Invalid logical file number; lfn references a file that is not opened, so designated function cannot be performed. Open file or reenter command, designating a different lfn.
xx0208	levl lfn	File designated by lfn already opened. Program should retry after file is closed.
xx0209	levl lfn	File not found, or another file create function was issued for the same file, so the file already exists.
xx020A	levl lfn	Position error. Reference to illegal location in file; i.e., beyond end of data or end of allocated space.
xx020B	levl lfn	Media error; a nonrecoverable physical I/O error occurred during the last I/O operation completed.
xx020C	levl lfn	Drives busy; there are no available drives on which to mount the specified disk volume. Retry later.
xx020D	levl lfn	Position error. Requested function references a record that is located beyond maximum number of bytes established for this usage of the file.
xx020E	levl lfn	Deleted record detected.
xx020F	levl lfn	Position error. Reference to location beyond the end of allocated space.

NOTE: Physical I/O error codes issued by the File Manager are described in Table IV-5-6.

Table IV-5-5. Uncoded Error Message Issued By File Manager

Message	Condition
MOUNT {DSK / RCD / FCD} volume ON cccc ANS: GO, WT, NO	Mount the specified disk volume on channel number cccc; answer GO, wait (WT), or NO.

Table IV-5-6. Physical I/O Error Messages

Message		Condition
Code	Other Information	
xx0{1/F}00	levl cccc sswd dswd	No error.
xx0{1/F}01	levl cccc sswd dswd	Request block already busy. Program logic error in task code.
xx0{1/F}02	levl cccc sswd dswd	Logical resource number (lrn) invalid; may be a configuration error. Correct configuration or task code.
xx0{1/F}03	levl cccc sswd dswd	Illegal wait. Program logic error in task code.
xx0{1/F}04	levl cccc sswd dswd	Invalid parameter(s) passed by component to driver, (internal error). Program logic error in task code.
xx0{1/F}05	levl cccc sswd dswd	Device not ready; prepare device so that it can be used.
xx0{1/F}06	levl cccc sswd dswd	Device time-out. Program should retry later.
xx0{1/F}07	levl cccc sswd dswd	Hardware error. Status bits specified by sswd.
xx0{1/F}08	levl cccc sswd dswd	Device disabled. Program logic error.
xx0{1/F}09	levl cccc sswd dswd	File mark encountered.
xx0{1/F}0A	levl cccc sswd dswd	Controller unavailable. Run controller test and verification.
xx0{1/F}0B	levl cccc sswd dswd	Device unavailable.
xx0{1/F}0C	levl cccc sswd dswd	Inconsistent request.

NOTE: xx - Designates which component is reporting the error.
Possible values are:

22 - File Manager
80-FF - User component

Table IV-5-7. Overlay Loader Error Messages

Message	Condition
0000	No error.
0F01	Invalid function code. Function code not in the range $0 \leq FC \leq 3$.
0F02	Invalid overlay ID. Overlay ID not in the range $0 \leq ID \leq MAX \leq 99$. or requested overlay not configured.
0F03	Illegal parameter. Overlay control table could not be located. Register B3 identifies the relative location zero of the root code.
0F04	Illegal load address. Attempt made to implicitly load a floatable overlay.
0F05	Illegal start address. Start address not within the bounds of the overlay.
0F06	Relocation error. Attempt made to relocate a nonrelocatable overlay.
0F07	Nonrecoverable media error.
0F08	Start error. Attempt made to start an overlay that has not been loaded.

Table IV-5-8. Operator Interface Manager Error Messages

Message	Condition
QUERY STALLED, NO FREE ID	Ten ID's have already been specified.
NO QUERY FOR ANSWER n	Message identifier incorrect. (See the Executive and Input/ Output manual for OIM dialog.)
n INVALID	Invalid message identifier specified. (See the Executive and Input/Output manual for OIM dialog.)

Table IV-5-9. Online Debugger Error Messages

Message	Condition
ACTIVE BP	Breakpoint specified in Sn command has already been set.
ADDR ERR	Address specified in DP, DH, Sn, or CH command is higher than the highest memory address configured. DP or DH command - Display includes last full print line that did not exceed highest memory address configured. CH command - Specified values placed in memory through the last configured word. Sn command - Breakpoint not set.
CMD ERR	Illegal operation code specified, invalid parameters specified, or En command is imbedded in a predefined command line Dn.
CMD INACTIVE	En, Pn, Ln, or P* command error; specified or all command lines are null.
CMD IO ERR	En command error; an I/O error occurred while attempting to read specified command line. The command line will not be executed.
DISK INACTIVE	No disk work file location specified in an SF command.
EXP ERR	Part of an expression is illegal.
ILL INST	GO command error; breakpoint was set at an invalid location. To proceed, clear the breakpoint and enter the GO command.
INACTIVE BP	Cn command error; specified breakpoint is not active, or L* command error; no breakpoints are active, or GO command error; the active level is not at a breakpoint.
INV LEVEL	AS command error; the active level has no ISA to use in referencing registers.
INV LRN	Illegal logical resource number (lrn) specified in an SF command, or an lrn expression was specified incorrectly.

Table IV-5-9 (cont). Online Debugger Error Messages

Message	Condition
IO ERR	Input/output error occurred while executing a command.
OVLY ERR	Error occurred while trying to read the overlay required by a command, or the incorrect overlay was loaded.
TOO LONG	Sn, Dn, or DT command line exceeds 126 characters, or VH command error; an expression was too long to be printed on one line.

PART V

OPERATING PROCEDURES FOR OFFLINE APPLICATIONS

SECTION 1

METHODS OF LOADING OFFLINE APPLICATIONS

Before a user-written offline application program can be loaded, it must be linked with the appropriate offline drivers, if they are required, to form a load module. The load module can be stored on either a disk or paper tape. The process of building an offline application program is described in the Executive and Input/Output manual.

Depending on the medium on which the offline application load module is stored, it can be loaded into memory by either the Disk Loader or the Paper Tape Loader. Two possible ways of loading the offline application from a disk are specified in Section 2 of this part, which includes separate procedures for each case, as follows:

- Bootstrapping and loading the offline application from a disk, using the Disk Loader, Command Processor, and operator's console
- Bootstrapping and loading the offline application from a disk, using the Disk Loader and control panel (including basic control panel)

Section 3 of this part specifies the third procedure:

- Bootstrapping and loading the offline application from paper tape, using the Paper Tape Loader and control panel

Control panel operations specified in this part are described in detail under "Control Panel" in Section 1 of Part II of this manual. All registers used in the following procedures are identified by selection code in Table II-1-4, "Register Selection Codes."

Loader halts specified in the following procedures are described in Appendix A, "Disk Loader," and Appendix B, "Paper Tape Loader," as appropriate.

During program execution, the operating procedures for the user-written offline application are determined by the program itself.

SECTION 2

BOOTSTRAPPING AND LOADING OFFLINE APPLICATIONS FROM DISK

The procedures for bootstrapping and loading an offline application program using the Disk Loader are described in this section. The two ways to load an offline application from a disk are:

- Using the operator's console and the Command Processor
- Using the control panel, without the Command Processor

Use of the Command Processor and operator's console in loading an offline application program permits flexibility in file assignment during development and test of the program. If file attachments are not required after the program has been tested, the program can be loaded from a disk using only the control panel, provided the program name was stored in the bootstrap record on the disk when the record was created by the Bootstrap Generator utility program. (Bootstrap Generator is described in the Utility Programs manual.)

Each of the above bootstrapping and loading procedures is discussed separately below.

BOOTSTRAPPING AND LOADING OFFLINE APPLICATIONS FROM DISK USING CONSOLE

Bootstrapping and loading an offline application program into memory using the Disk Loader and the operator's console involves:

1. Invocation of the bootstrapping procedure, which causes the boot loader to be loaded into memory
2. Use of the boot loader to load the Disk Loader, which loads the Offline Trap Handler and the Command Processor into memory
3. Submission of commands to the Command Processor that contain information to be used in loading and running the offline application program
4. Loading of the offline application program by the Disk Loader, which then turns control over to the program

Bootstrapping and loading an offline application, as summarized in Figure V-2-1, consists of these phases (as described in the following subsections):

- Mounting the required disk volume(s)
- Performing one of two possible bootstrapping procedures
- Performing loading of the offline application program

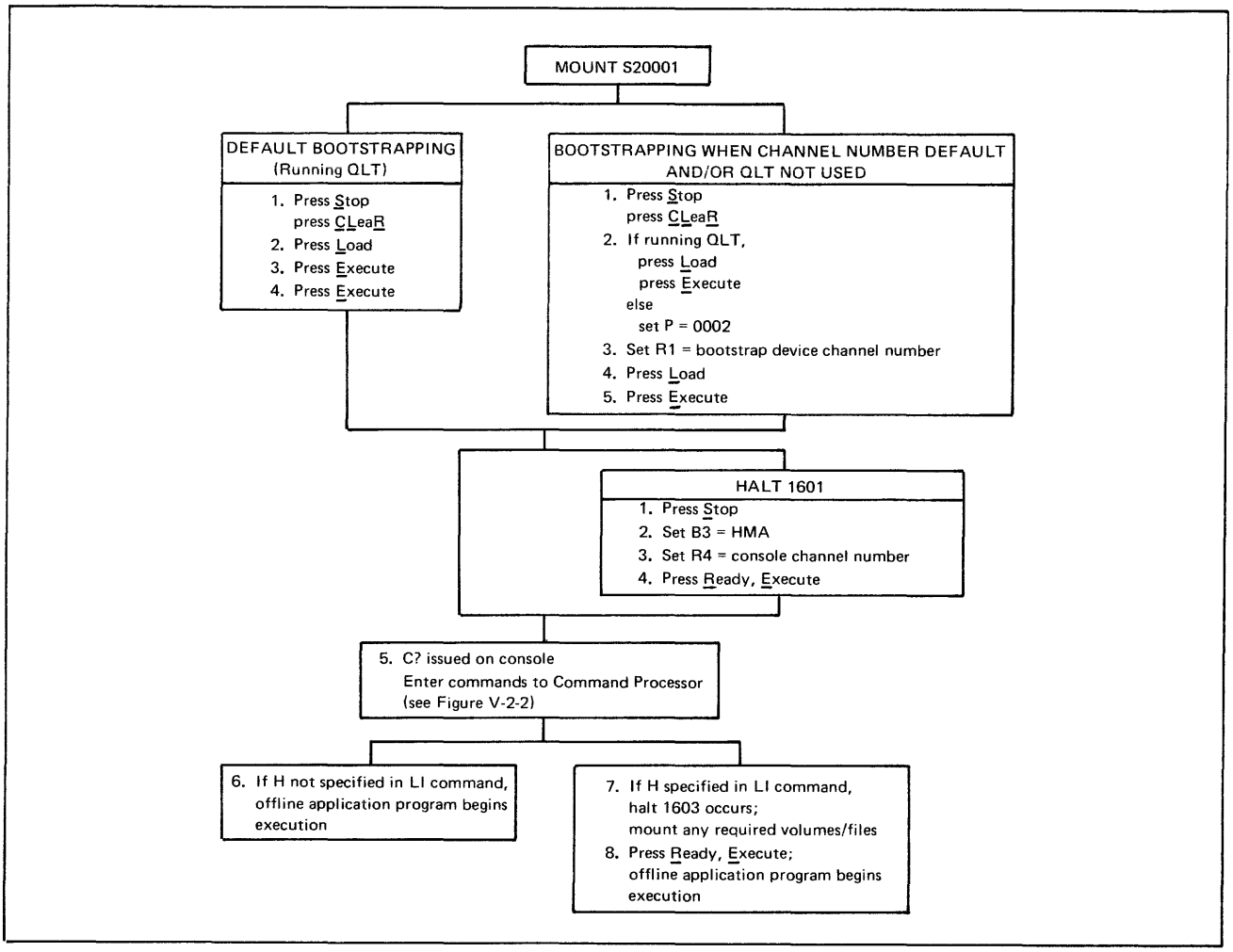


Figure V-2-1. Procedure for Bootstrapping and Loading Offline Application From Disk Using Console

Parameter values preset in the bootstrap record and boot loader when they were created by Bootstrap Generator are used during bootstrapping and loading. The values for the high memory address and operator's console channel number that were preset in the bootstrap record and boot loader are referred to as "default" values in the bootstrapping and loading steps that follow, because they can be modified through the control panel. The indicator setting for loader halt 1601 as preset in the boot loader during bootstrap generation cannot be modified during bootstrapping and loading.

Mounting Required Disk Volumes

Before you bootstrap and load the offline application program, determine whether the disk volume initially identified by the volume name S20001 is mounted. This volume includes the bootstrap record and boot loader and the file named PROGFILE that contains the Disk Loader, the Offline Trap Handler, and the Command Processor. If the volume is not mounted, mount it either on a device whose channel number is 0400_{16} (the default bootstrap device channel number) or on another appropriate device to be used in bootstrapping.

When the offline application program you are loading resides on a disk volume other than volume S20001, mount that volume before you reach the loading phase of the bootstrapping and loading procedure. This disk volume must be of the same type as the bootstrap device. Before executing the program, determine that all files specified in commands to the Command Processor during loading have been mounted on the appropriate channels.

Bootstrapping Phase for Offline Application on Disk

Control panel operation causes the bootstrap record to be read into memory from the peripheral device on which it is stored; the bootstrap record then loads the boot loader. The LOAD indicator lamp goes on when the Load key is pressed, and will remain on until the bootstrapping routine jumps to the first location of the bootstrap record.

Perform one of the following two bootstrapping procedures as described in the next two subsections:

- Default bootstrapping: Use this procedure if you wish to perform the Quality Logic Test (QLT) and use the default bootstrap device channel number (0400_{16}). (The QLT clears memory; therefore, do not perform the QLT if you need to preserve the contents of memory.)
- Bootstrapping when channel number default and/or QLT not used: Perform this procedure if you wish to change the bootstrap device channel number and/or omit the QLT.

DEFAULT BOOTSTRAPPING

The default bootstrapping procedure is as follows:

1. Perform master clear operation, automatically setting the P-register to 0000_{16}.
2. Press Load.
3. Press Execute. The QLT will be run, setting register R1 to the default bootstrap device channel number and the P-register to 0002_{16}.
4. Press Execute, causing the bootstrap record to be loaded into memory. The bootstrap record loads the boot loader, which places the default high memory address in register B3 and the default operator's console channel number in register R4. The boot loader halts if loader halt 1601 was preset in the boot loader when it was created by the Bootstrap Generator. At loader halt 1601, proceed to step 1 of the loading phase. If no halt occurs, the boot loader automatically loads the Disk Loader as in step 5 of the loading phase.

BOOTSTRAPPING WHEN CHANNEL NUMBER DEFAULT AND/OR QLT NOT USED

The bootstrapping procedure described below permits you to use a different bootstrap device channel number and/or omit the QLT.

1. Perform master clear operation.
2. If you are running the QLT, press Load and then Execute (the QLT sets the P-register to 0002_{16}); otherwise, set the P-register to 0002_{16}.

3. Set the bootstrap record device channel number in register R1, as follows:

   a. If you are using a channel number other than the default channel number, place the desired channel number in register R1.

   b. If you are using the default channel number but are not running the QLT, place the default channel number 0400_{16} in register R1. (If you are using the default channel number and are running the QLT, the QLT places the default channel number in register R1.)

4. Press Load.

5. Press Execute, causing the bootstrap record to be loaded into memory. The bootstrap record loads the boot loader, which places the default high memory address in register B3 and the default operator's console channel number in register R4. The boot loader halts if loader halt 1601 was preset in the boot loader when it was created by the Bootstrap Generator. At loader halt 1601, proceed to step 1 of the loading phase. If no halt occurs, the boot loader automatically loads the Disk Loader as in step 5 of the loading phase.

Loading Phase for Offline Application From Disk Using Console

After you have completed one of the two alternative bootstrapping procedures described above, perform the following loading procedure:

1. If loader halt number 1601 occurs, press Stop, placing the central processor in the stop state.

2. Enter in register B3 the high memory address, unless you are using the default value.

3. Enter in register R4 the channel number of the KSR-like device you are using as the operator's console, unless you are using the default value.

4. If you have performed steps 1 through 3, press Ready, then Execute.

5. The Disk Loader is loaded into high memory. The Disk Loader loads the Offline Trap Handler, and then loads the Command Processor, which issues the message C? through the operator's console, indicating that it is ready to receive input.

   Enter commands to the Command Processor through the operator's console (or any other device specified as the command input device), as required to load the offline application program you wish to run. See below, "Input to Command Processor Before Offline Application Is Loaded," for a full set of commands that can be submitted to the Command Processor before the program is loaded. Specify the halt option (H) in the load initialize (LI) command if any required volumes/files have not been made available.

6. If you do not specify the halt option (H) in the load initialize command to the Command Processor, the Disk Loader will load the offline application program and turn control over to it to start program execution.

7. If you specify the halt option (H) in the load initialize command to the Command Processor, the Disk Loader will load the offline application program and then halt (halt number 1603 in register R1). Make any other required volumes or files that are not already mounted available to the offline application.

8. Press Ready, then Execute. The offline application will begin execution.

Input to Command Processor Before Offline Application Program Is Loaded

Figure V-2-2 illustrates the input commands to the Command Processor that can be entered through the operator's console before an offline application program is loaded. The load command must always be entered. Other commands should be entered only if appropriate information does not exist in the loader communication area or attach table area after the Command Processor is loaded into memory. For detailed information regarding the format of commands to the Command Processor, see "Command Processor" in the Program Development Tools manual.

Command Type	Command Format
Load Initialize	LI [rel] [,H]
Attach (Program File)	AT 00,DSKcccc,file-name
Attach (Other Files)	AT lfn,sdncccc[,file-name]
Load	member-name

Figure V-2-2. Input to Command Processor Before Offline Application Program Is Loaded

BOOTSTRAPPING AND LOADING OFFLINE APPLICATION FROM DISK USING CONTROL PANEL

An offline application program can be loaded into memory from a disk, using a control panel and the Disk Loader. The full control panel must be used if operator intervention is required during bootstrapping and loading. If no operator intervention is required, loading can be performed using either the full control panel, or, if available, the basic control panel. (See Appendix F for a description of loading using the basic control panel.)

Bootstrapping and loading an offline application program into memory, using the control panel and the Disk Loader, involves:

- Invocation of the bootstrapping procedure which causes the boot loader to be loaded into memory.
- Using the boot loader to load the Disk Loader, which loads the Offline Trap Handler. The Disk Loader then enters into memory information required to load the offline application.
- Loading of the application program by the Disk Loader, which then turns control over to the program.

Parameter values preset in the bootstrap record and boot loader when they were created by Bootstrap Generator are used in loading the offline application from disk. The program must have been identified by file and member name in the boot loader when the boot loader was created by Bootstrap Generator, and the KSR parameter in the bootstrap record (representing the operator's console channel number) must have been preset to zero.

If loading without operator intervention is to occur (i.e., none of the parameter values in the bootstrap record and boot loader are to be modified, and no halts are required), only the following actions are needed to load the offline application from disk using the full control panel:

- Mounting the required disk volume(s)
- Performing one of two possible bootstrapping procedures (as described below), causing the program to be loaded without additional operator intervention

During loading, if loader halt 1601 occurs, the following parameter values that were stored in the bootstrap record and boot loader can be modified through the full control panel:

- High memory address
- Relocation factor
- Program loading device channel number
- Load and halt option (indicator setting for loader halt 1603)

The values that were preset in the bootstrap record and boot loader for these modifiable parameters are referred to as "default" values in the bootstrapping and loading procedures that follow.

The complete operation for bootstrapping and loading an offline application from disk using the control panel, summarized in Figure V-2-3, consists of these phases (as described in the following subsections):

- Mounting the required disk volume
- Performing one of two possible bootstrapping procedures
- Performing loading of the offline application program

Mounting Required Disk Volume

Before you bootstrap and load the offline application, determine whether a disk volume that includes the bootstrap record and boot loader and a file named PROGFILE containing the Disk Loader and the Offline Trap Handler is mounted. If the volume is not mounted, mount it either on a device whose channel number is 0400_{16} (the default bootstrap device channel number) or on another appropriate device to be used in bootstrapping.

Bootstrapping Phase for Offline Application on Disk

Control panel operation causes the bootstrap record to be read into memory from the peripheral device on which it is stored; the bootstrap record then loads the boot loader. The LOAD indicator lamp goes on when the Load key is pressed, and will remain on until the bootstrapping routine jumps to the first location of the bootstrap record.

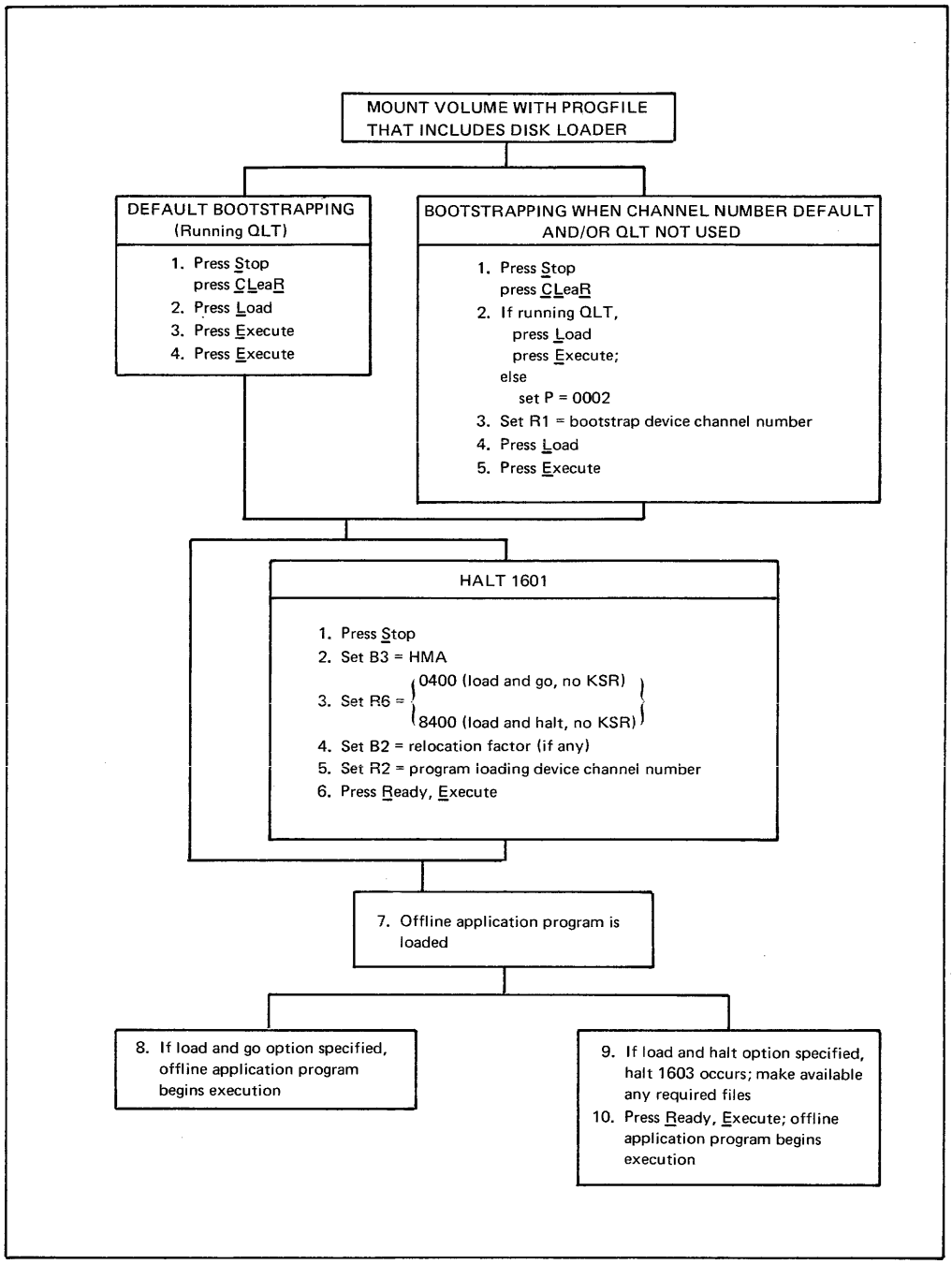


Figure V-2-3. Procedure for Bootstrapping and Loading Offline Application From Disk Using Control Panel

Perform one of the following two bootstrapping procedures as described in the next two subsections:

- Default bootstrapping: Use this procedure if you wish to perform the Quality Logic Test (QLT) and use the default bootstrap device channel number (0400_{16}). (The QLT clears memory; therefore, do not perform the QLT if you need to preserve the contents of memory.)
- Bootstrapping when channel number default and/or QLT not used: Perform this procedure if you wish to change the bootstrap device channel number and/or omit the QLT.

DEFAULT BOOTSTRAPPING

The default bootstrapping procedure is as follows:

1. Perform master clear operation, automatically setting the P-register to 0000_{16}.
2. Press Load.
3. Press Execute. The QLT will be run, setting register R1 to the default bootstrap device channel number and the P-register to 0002_{16}.
4. Press Execute, causing the bootstrap record to be loaded into memory. The bootstrap record loads the boot loader, which places the default high memory address in register B3. The boot loader halts if loader halt 1601 has been specified in the boot loader when it was created by Bootstrap Generator. At this halt, proceed to step 1 of the loading phase. If no halt occurs, the boot loader automatically loads the Disk Loader as in step 7 of the loading phase.

BOOTSTRAPPING WHEN CHANNEL NUMBER DEFAULT AND/OR QLT NOT USED

The bootstrapping procedure described below permits you to use a different bootstrap device channel number and/or omit the QLT.

1. Perform master clear operation.
2. If you are running the QLT, press Load and then Execute (the QLT sets the P-register to 0002_{16}); otherwise, set the P-register to 0002_{16}.
3. Set the bootstrap record device channel number in register R1 as follows:
   a. If you are using a channel number other than the default channel number, place the desired channel number in register R1.
   b. If you are using the default channel number but are not running the QLT, place the default channel number 0400_{16} in register R1. (If you are using the default channel number and are running the QLT, the QLT places the default channel number in register R1.)
4. Press Load.
5. Press Execute, causing the bootstrap record to be loaded into memory. The bootstrap record loads the boot loader, which places the default high memory address in register B3. The boot loader halts if loader halt 1601 has been specified in the boot loader when it was created by Bootstrap Generator. At this halt, proceed to step 1 of the loading phase. If no halt occurs, the boot loader automatically loads the Disk Loader as in step 7 of the loading phase.

Loading Phase for Offline Application From Disk Using Control Panel

After you have completed one of the two alternative bootstrapping procedures described above, perform the following loading procedure:

1. If loader halt number 1601 occurs, press Stop to place the central processor in the stop state.
2. Enter in register B3 the high memory address, unless you are using the default value.
3. Enter in register R6 the appropriate bit mask of the loader indicator word, unless you are using the default indicator setting for loader halt 1603 that was preset in the boot loader by the Bootstrap Generator. The loader indicator word is described in Appendix A. Bit 5 of the word must be set to 1 to indicate that the operator's console is not present. Enter one of the following in register R6:

   0400 - Load and go, no KSR

   8400 - Load and halt, no KSR

   The load and halt option causes loader halt 1603 to occur; the halt can be used to make available any volumes/files not already mounted (see step 9).
4. Enter in register B2 the relocation factor to be added to all relocatable addresses in the program being loaded, if you wish to relocate the program in memory. If the relocation factor is not entered, the default value that was set in the boot loader is assumed.
5. Enter in register R2 the channel number of the device from which the program is to be loaded, unless you are using the default value.
6. If you have performed steps 1 through 5, press Ready, then Execute.
7. The boot loader reads the Disk Loader into memory. The loader loads the Offline Trap Handler and then places the file name, member name, program device channel number, relocation factor, and loader indicator word in the loader communication area. The loader then loads the offline application program.
8. If the load and go indicator is set, the loader turns control over to the offline application program which begins execution.
9. If the load and halt indicator is set, the loader halts (halt number 1603 in register R1). Make available to the program any other required files that are not already mounted.
10. After loader halt number 1603, press Ready, then Execute. The offline application program will begin execution.

SECTION 3
BOOTSTRAPPING AND LOADING OFFLINE APPLICATIONS FROM PAPER TAPE

The procedures for bootstrapping and loading an offline application program from paper tape using the Paper Tape Loader and the control panel are described in this section. The Command Processor is not used when the offline application is loaded from paper tape.

Bootstrapping and loading the offline application from paper tape into memory, using the Paper Tape Loader, involves:

- Invocation of the bootstrapping procedure, which causes the bootstrap record to be loaded into memory
- Use of the bootstrap record to load the Paper Tape Loader, which loads the Offline Trap Handler
- Loading of the offline application by the Paper Tape Loader, which then turns control over to the offline application

Parameter values preset in the bootstrap record when it was created by Bootstrap Generator are used in loading the offline application from paper tape.

If loading without operator intervention is to occur (i.e., the parameter values preset in the bootstrap record during bootstrap generation are to be used, and no halts are to occur), only the following actions are required to load an offline application from paper tape using a full control panel:

- Making the required load modules available on the ASR
- Performing one of two possible bootstrapping procedures (as described below), causing the program to be loaded without additional operator intervention

A basic control panel, if available, can be used to load an offline application, provided no operator entries are required during bootstrapping and loading (see Appendix F).

During loading, if loader halt 1601 occurs, the following parameter values stored in the bootstrap record can be modified through the full control panel:

- High memory address
- Operator's console channel number

If loader halt 1602 occurs, the preset values of the following parameters can be modified:

- Relocation factor
- Indicator setting for loader halt 1603 (the load and halt option)

The values that were preset in the bootstrap record for these modifiable parameters are referred to as "default" values in the bootstrapping and loading procedures that follow.

The complete operation for bootstrapping and loading an offline application from paper tape, summarized in Figure V-3-1, consists of these phases (as described in the following subsections):

- Making the required load modules available on the ASR
- Performing one of two alternative bootstrapping procedures
- Performing loading of the offline application

MAKING REQUIRED OFFLINE APPLICATION LOAD MODULE AVAILABLE ON PAPER TAPE

Before you perform the bootstrapping and loading procedure, you must make the following available on paper tape:

1. Bootstrap record followed by Paper Tape Loader and Offline Trap Handler
2. User-written offline application program

The paper tape must be mounted on a paper tape reader (ASR) connected to channel number 0400_{16} (the default bootstrap device channel number) or to another appropriate channel number to be used in bootstrapping. The load modules are loaded sequentially from the paper tape reader.

BOOTSTRAPPING PHASE FOR OFFLINE APPLICATION ON PAPER TAPE

Control panel operation causes the bootstrap record to be read into memory from the peripheral device on which it is stored. The LOAD indicator lamp goes on when the Load key is pressed, and will remain on until the bootstrapping routine jumps to the first location of the bootstrap record.

Perform one of the following two bootstrapping procedures as described in the next two subsections:

- Default bootstrapping: Use this procedure if you wish to perform the Quality Logic Test (QLT) and use the default bootstrap device channel number (0400_{16}). (The QLT clears memory; therefore, do not perform the QLT if you need to preserve the contents of memory.)
- Bootstrapping when channel number default and/or QLT not used: Perform this procedure if you wish to change the bootstrap device channel number and/or omit the QLT.

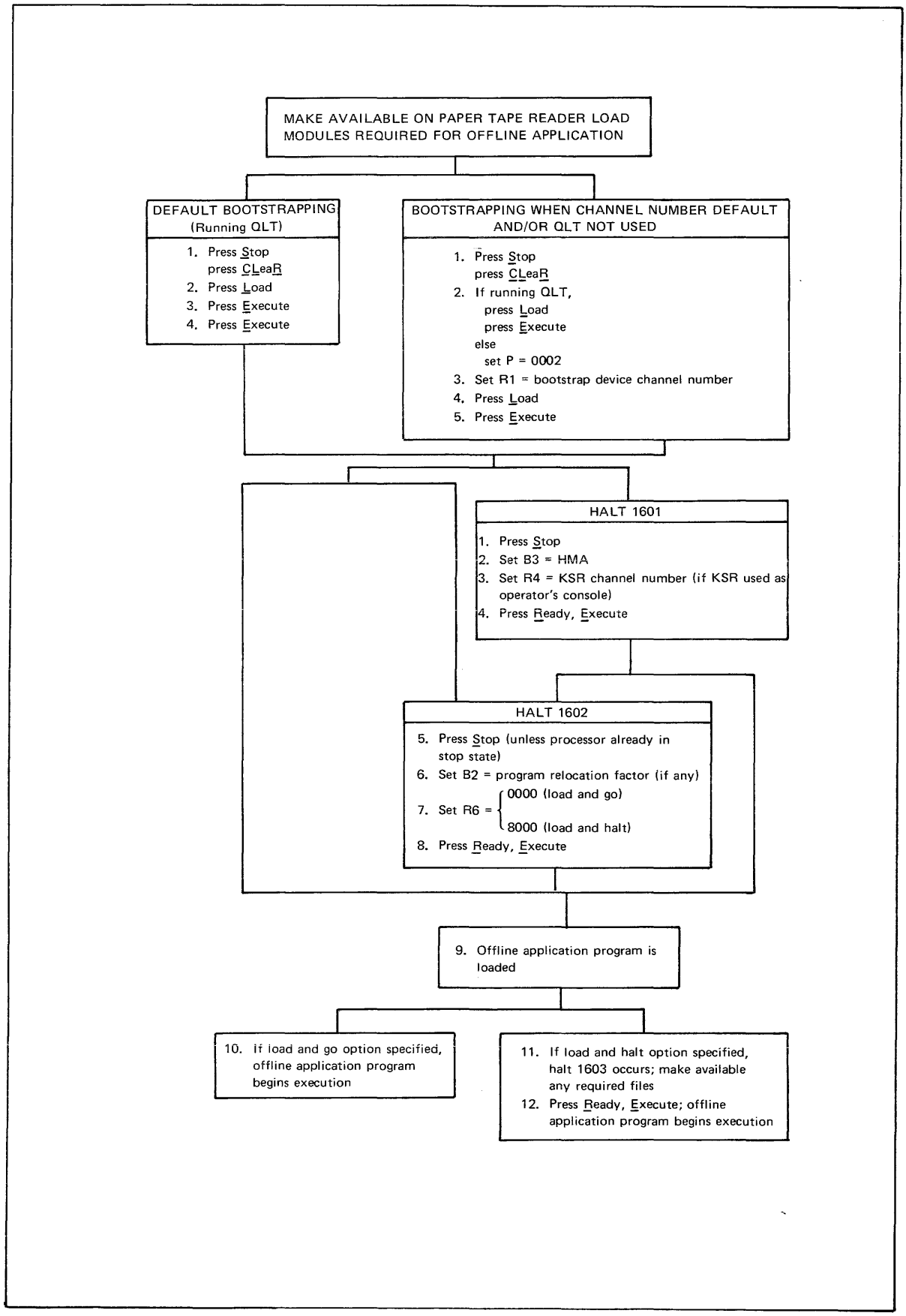


Figure V-3-1. Procedure for Bootstrapping and Loading Offline Application From Paper Tape Using Control Panel

Default Bootstrapping

The default bootstrapping procedure is as follows:

1. Perform master clear operation, automatically setting the P-register to 0000_{16}.
2. Press Load.
3. Press Execute. The QLT will be run, setting register R1 to the default bootstrap device channel number and the P-register to 0002_{16}.
4. Press Execute, causing the bootstrap record to be loaded into memory. The bootstrap record places the default high memory address in register B3 and the default operator's console channel number in register R4. The bootstrap record automatically loads the Paper Tape Loader and the loader loads the Offline Trap Handler, unless a loader halt 1601 occurs. If halt 1601 occurs, proceed to step 1 of the loading phase. If halt 1602 occurs, proceed to step 5. If no halt occurs, proceed to step 9. (Halt numbers appear in register R1.)

Bootstrapping When Channel Number Default and/or QLT Not Used

The bootstrap procedure described below permits you to use a different bootstrap device channel number and/or omit the QLT.

1. Perform master clear operation.
2. If you are running the QLT, press Load and then Execute (the QLT sets the P-register to 0002_{16}); otherwise set the P-register to 0002_{16}.
3. Set the bootstrap record device channel number in register R1, as follows:
   a. If you are using a channel number other than the default channel number, place the desired channel number in register R1.
   b. If you are using the default channel number but are not running the QLT, place the default channel number 0400_{16} in register R1. (If you are using the default channel number and are running the QLT, the QLT places the default channel number in register R1.)
4. Press Load.
5. Press Execute, causing the bootstrap record to be loaded into memory. The bootstrap record places the default high memory address in register B3 and the default operator's console channel number in register R4. The bootstrap record automatically loads the Paper Tape Loader and the loader loads the Offline Trap Handler, unless a loader halt 1601 occurs. If halt 1601 occurs, proceed to step 1 of the loading phase. If halt 1602 occurs, proceed to step 5. If no halt occurs, proceed to step 9. (Halt numbers appear in register R1.)

LOADING PHASE FOR OFFLINE APPLICATION FROM PAPER TAPE

After you have completed one of the two alternative bootstrapping procedures described above, perform the following loading procedure:

1. If loader halt 1601 occurs, press Stop, placing the central processor in the stop state; perform steps 2 through 4.
2. Enter in register B3 the high memory address, unless you are using the default value.

3. If you are using a KSR-like device as an operator's console, enter in register R4 the device channel number, unless you are using the default value.

4. If you have performed steps 1 through 3, press Ready, then Execute, causing the Paper Tape Loader and Offline Trap Handler to be loaded into high memory.

5. If loader halt 1602 occurs, press Stop, unless the processor is already in the stop state (see step 1 above); perform steps 6 through 9.

6. Enter in register B2 the relocation factor to be added to all relocatable addresses in the program being loaded, unless you wish to use the default value of zero.

7. Enter the following bit mask of the loader indicator word in register R6:

   8000_{16} - Load and halt

   unless you are using the default value:

   0000_{16} - Load and go

   The load and halt option can be used to mount any other files required by the offline application (see step 11 below). (The loader indicator word is described in Appendix B of this manual.)

8. If you have performed steps 5 through 7, press Ready, then Execute.

9. The Paper Tape Loader loads the offline application program, which is the next load module on the paper tape reader.

10. If the load and go indicator is set, the loader turns control over to the offline application program which begins execution.

11. If the load and halt indicator is set, the loader halts (halt 1603 in register R1). Mount any other files required by the offline application program.

12. Press Ready, then Execute. The offline application program begins execution.

APPENDICES

APPENDIX A
DISK LOADER

The Disk Loader resides on the system load module disk in a file called PROGFILE; the member name of the Disk Loader in PROGFILE is DSKLDR[1]; this load module is used with either a diskette or a cartridge disk. The bootstrap routine reads the bootstrap record into memory; the bootstrap record reads the boot loader into memory, which in turn brings the Disk Loader in and transfers control to it.

The Disk Loader selects a load module from a disk, as specified by the Command Processor or the Configuration Load Manager, or the bootstrap record (in which a program name was stored when the record was created by Bootstrap Generator). The loader loads the module, and turns control over to it.

LOADER STRUCTURE AND FUNCTIONS

The Disk Loader consists of several subroutines that perform the various services required in the process of loading modules into memory. The three major subroutines are:

- Loader Initialization Routine
- Command Processor Calling Sequence
- Loader

Loader Initialization Routine

When the boot loader finishes processing, it turns control over to the Disk Loader at the entry point of this routine. The functions of this routine are:

- Save the bootstrap channel in the loader communication area (see Table A-1).
- Save the two halt addresses in the loader communication area.
- Set values for the Disk Loader and the processor type in the loader indicator word (see Table A-2).
- Load the Offline Trap Handler (see Appendix D).
- Transfer control to the Command Processor Calling Sequence.

[1]The Disk Loader (DSKLDR), Command Processor (CMDPRC) and Offline Trap Handler (TRPHND) modules are always loaded from the file PROGFILE on the channel used in bootstrapping.

Table A-1. Contents of the Loader Communication Area as Used With Disk Loader

Entry	Location (Displacement From High Memory Address)	Size (Words)	Description
1	HMA	1	Loader indicator word
2	HMA-2	2	Relocation factor
3	HMA-6	4	Member name (Program identification)
4	HMA-12	6	File name
5	HMA-13	1	Program loading channel number
6	HMA-14	1	Loader entry point
7	HMA-15	1	Reserved
8	HMA-16	1	Command Processor Calling Sequence entry point
9	HMA-17	1	Attach table entry point
10	HMA-19	2	Address of special item HALT
11	HMA-21	2	Address of file/member not found HALT
12	HMA-22	1	Bootstrap channel
13	HMA-23	1	Current system console channel number
14	HMA-25	2	Loader lower boundary
15	HMA-26	1	Initial system console channel number

Table A-2. Loader Indicator Word Contents as Used With Disk Loader

Bit Number	Values
0	0 - Execute after loading (default value)
	1 - Halt after loading
1	0 - Execute before returning to caller (default)
	1 - Return to caller after loading
2-4	Reserved
5	0 - System console present (default)
	1 - No system console
6-7	Reserved
8	0 - Not applicable when Disk Loader in use
	1 - Indicates Disk Loader in use
9-15	Must be zeros

Command Processor Calling Sequence

This routine supplies the information required to load the Command Processor by placing several pieces of information into the loader communication area. The functions of this subroutine are:

- Initialize the relocation factor to the value that was set in the bootstrap record when it was created by Bootstrap Generator.
- Set the program loading channel to the value preset in the bootstrap record.
- Enter the file and member names for the Command Processor (or for any other program whose file and member names were stored in the bootstrap record).
- Transfer control to the loader.

This routine has a special function in those situations where a KSR-like device is not being used as a command input device; namely, it retrieves any information previously entered through the control panel and stored by the bootstrap routine, and places this information in the loader communication area. That information can consist of the relocation factor, loader indicator word, and/or program loading device channel number.

Loader

This subroutine searches the specified file for the member to be loaded. When the member is located, the loader reads the member a sector at a time, analyzes the item type, and relocates and distributes the code. It resolves backpatch chains and relocates global addresses.

When the end-of-member designator is found, the loader either turns control over to the loaded module, or comes to a halt, depending on the value of bit 0 in the loader indicator word.

LOADING PROCESS

The loading process is the series of actions taken by the Disk Loader to bring system software and application programs into memory for execution.

The bootstrap routine loads the boot loader, and turns control over to it. The boot loader loads the default high memory address into register B3 and the default system or operator's console channel number into register R4. (Honeywell defaults for these values are $1FFF_{16}$ and 0500_{16} respectively.) The boot loader then loads the Disk Loader.

The Disk Loader operates in systems that use, as well as those that do not use, a system or operator's console. To perform its functions, the Disk Loader uses information supplied to it in a structure called the loader communication area (see Table A-1).

When a system or operator's console is present, the Disk Loader uses information supplied by the Command Processor to load the system and application programs needed for program development, to load the Configuration Load Manager (CLM), or to load an offline application. The Configuration Load Manager provides information to the Disk Loader for loading CLM and Executive modules and application programs required for the execution of an online application.

If a system console is not present, information normally supplied by the Command Processor is supplied by the bootstrap record and boot loader (in which parameter values were set when the records were created by Bootstrap Generator); some of these parameter values can be modified through the control panel during loading.

Once the information in the loader communication area is complete, the operation of the Disk Loader is essentially the same whether or not a system or operator's console is present.

Briefly, entering information through the control panel for the Disk Loader is done in the following way. The boot loader can halt with a value of 1601 in register R1, if the bootstrap halt was preset in the bootstrap record. (See Table A-3 for loader halts.) At this point, you can alter the high memory address; set the bits of the loader indicator word for either a "load and go," or a "load and halt" operation, with or without system console; set the relocation factor; and specify the loading channel for the program to be loaded, if it is different from the value preset in the bootstrap record. This information is stored in a temporary area within the boot loader.

If you have entered information through the control panel, press the Ready and Execute keys in that order. The boot loader loads the Disk Loader and transfers control to the entry point of the initialization subroutine of the Disk Loader. The initialization subroutine performs the functions listed earlier in this appendix, and in so doing, provides some of the information for the loader communication area. It then transfers control to the Command Processor Calling Sequence portion of the Disk Loader.

The Command Processor Calling Sequence completes the information needed in the loader communication area. In systems that operate without a system or operator's console, this routine retrieves any information that had been entered through the control panel and stored in a temporary area in the boot loader, and places it in the loader communication area.

For systems that operate with a system or operator's console, this routine performs the functions listed earlier, thereby providing for the loading of the Command Processor.

Table A-3. Disk Loader Halts

Halt Number	Significance	Operator Action
1601	Boot loader entry point halt	
	1. High memory address may be altered	Enter HMA into B3
	2. If no system console:	
	a. Loader indicator word can be set for	
	Load and Go	Enter 0400 into R6
	or	
	Load and Halt	Enter 8400 into R6
	b. Relocation factor can be altered	Enter relocation factor into B2
	c. Program loading channel number can be altered	Enter channel number into R2
		Press Execute
1603	Loader halt (load and halt option)	Press Execute
1604	Software bootstrap entry point halt — allows mounting of system disk on the bootstrap channel device.	Press Execute
1612	File/member not found. For systems using a system console, pressing the Execute key causes the Command Processor module to be reloaded from the file PROGFILE.	Mount the correct volume and press Execute
1615	1. An unresolved symbol was encountered in the load module, or	1. Press Execute to bypass item
	2. A definition to be passed to the CLM was encountered and the CLM is not in memory.	2. Load the CLM and reload the module
1616	Loader I/O error: the loader cannot complete physical transfer of the data from this module.	Press Execute to reissue I/O request or try loading from another similar device after rebooting.
1692	Disk Loader not found	Verify that correct disk is mounted and reboot the system.
1694	Invalid item type encountered in load module.	Relink the load module and retry the load operation.
1695	Load module size exceeds available memory or relocation factor too large.	Correct the problem and reboot the system.
1696	Boot loader I/O error.	Verify that correct disk is mounted and reboot the system.
16FF	Trap occurs during offline program execution (halt number is set in R1 by the Offline Trap Handler).	See Appendix D for operator action.

When the Command Processor Calling Sequence finishes processing, it branches to the loader entry point[1] (see Table A-1), turning control over to the loader routine. The loader uses the information in the loader communication area to search for and load the required program.

[1] During internal loading, a program in memory can access the Disk Loader at this entry point (HMA-14). The executing program first enters in the loader communication area (see Table A-1) the information required for the next program to be loaded (including the program name), and then branches to the loader entry point. The loader loads the program whose name is stored in the loader communication area.

APPENDIX B
PAPER TAPE LOADER

The Paper Tape Loader is read into memory from paper tape by the bootstrap record that is itself read from paper tape by the bootstrap routine.

Since paper tape is a sequential medium, all of the modules to be loaded from this medium must be on the tape in the order in which they will be loaded.

LOADER STRUCTURE AND FUNCTIONS

The Paper Tape Loader consists of two subroutines:

- Loader Initialization Routine
- Loader

Loader Initialization Routine

When the bootstrap record finishes bringing the Paper Tape Loader into memory, it turns control over to this subroutine. The functions of this subroutine are:

- Save the bootstrap channel in the loader communication area.
- Save the two halt addresses in the loader communication area (see Table B-1).
- Load the Offline Trap Handler (see Appendix D).
- Branch to entry point 1 of the loader.

Loader

The loader reads a block of load text from the tape, analyzes the item type, relocates addresses and distributes code, and resolves backpatch chains.

Table B-1. Contents of the Loader Communication Area as Used With Paper Tape Loader

Entry	Location (Displacement From High Memory Address)	Size (Words)	Description
1	HMA	1	Loader indicator word[a]
2	HMA-2	2	Relocation factor
3	HMA-6	4	File name (Program identification)
4	HMA-12	6	Not used for paper tape
5	HMA-13	1	Bootstrap channel number
6	HMA-14	1	Loader entry point 2
7	HMA-15	1	Loader entry point 1
8	HMA-16	1	Not used for paper tape
9	HMA-17	1	Not used for paper tape
10	HMA-19	2	Address of special item halt
11	HMA-21	2	Address of program not found halt
12	HMA-22	1	Not used for paper tape
13	HMA-23	1	Operator's console channel number

[a]See Table B-2.

Table B-2. Loader Indicator Word Contents as Used With Paper Tape Loader

Bit Number	Values
0	0 - Execute after loading (default)
	1 - Halt after loading
1	0 - Execute before returning to caller
	1 - Return to caller after loading
2-4	Reserved
5	0 - Operator's console present
	1 - No operator's console
6-7	Reserved
8-9	Must be zeros
10	0 - Not applicable when Paper Tape Loader in use
	1 - Indicates that the Paper Tape Loader in use
11-15	Must be zeros

LOADING PROCESS

The loading process using the Paper Tape Loader requires that all modules to be loaded be on the paper tape in the order in which they will be loaded.

The sequence of modules on the tape for setting up either an offline or an online application is as follows:

Offline	Online
Bootstrap Record	Bootstrap Record
Paper Tape Loader	Paper Tape Loader
Offline Trap Handler	Offline Trap Handler (applicable only during initial loading of CLM)
User-written program	Configuration Load Manager CLM modules needed Executive modules needed User-written programs

The information required for the loader communication area is entered through the control panel during the loading process as described below and in the procedures earlier in this manual.

When the bootstrap routine turns control over to the bootstrap record, the default high memory address is loaded into register B3 and the default operator's console channel number is loaded into register R4. (Honeywell defaults for these values are $1FFF_{16}$ and 0500_{16}, respectively). A halt can occur with the value 1601 in R1 (see Table B-3) if halt 1601 was specified in the bootstrap record when it was created. At this point, the high memory address and the operator's console channel number (if any) can be altered by entering new hexadecimal values in B3 and R4, respectively. When the Execute key is pressed, the bootstrap record reads the Paper Tape Loader into memory. If no halt 1601 occurs, the bootstrap record automatically loads the Paper Tape Loader. Control is then given to the initialization routine.

The initialization routine saves the following information in the loader communication area: the bootstrap channel number, and the addresses for special item type and program mismatch conditions (see Table B-1). It then loads the Offline Trap Handler (see Appendix D), and turns control over to the loader at entry point 1.

At entry point 1, a halt with the value 1602 in R1 can occur if so indicated in the bootstrap record when it was created by Bootstrap Generator. At this halt, a relocation factor other than the default value that was preset in the bootstrap record may be entered into B2, and the loader indicator word may be set for either a "load and go," or a "load and halt" condition. Pressing the Execute key causes this information to be placed into the loader communication area.

Table B-3. Paper Tape Loader Halts

Halt Number	Significance	Operator Action
1601	Bootstrap halt:	
	1. High memory address can be altered.	Enter HMA in B3.
	2. Operator's console channel number can be altered.	Enter console channel number in R4.
1602	Loader entry point 1 halt:	
	1. Relocation factor can be changed.	Enter relocation factor in B2.
	2. Loader indicator word can be set for:	
	a. Load and Go	Enter 0000 in R6.
	or	
	b. Load and Halt	Enter 8000 in R6.
1603	Loader halt (load and halt option).	Press Execute.
1612	Result of program name mismatch.	Mount correct paper tape on reader and press Execute.
1615	1. An unresolved symbol was encountered in the load module, or	1. Press Execute to bypass item.
	2. A definition to be passed to the CLM was encountered and the CLM is not in memory.	2. Load the CLM and reload the module.
1616	Loader I/O error: the loader cannot complete physical transfer of the data from this module.	Press Execute to reissue I/O request or try loading from another similar device.
1693	Checksum error.	Reboot and retry paper tape strip.
1694	Invalid item type encountered in load module.	Relink the load module and retry the load operation.
1695	Load module size exceeds available memory or relocation factor too large.	Correct the problem and reboot the system.
1696	Bootstrap record I/O error.	Verify that correct paper tape is mounted and reboot the system.
1697	Type 01 item not found.	Reload paper tape and reboot the system.
16A4	Invalid block size detected in load module.	Reboot and retry. If problem persists, repunch paper tape.
16FF	Trap occurs during offline program execution (halt number is set in R1 by Offline Trap Handler).	See Appendix D for operator action.

The program identification field is initialized to spaces and a branch is made to entry point 2 of the loader. At entry point 2, the loader checks the program identification field of the loader communication area. If the field has spaces in it, the loader loads the next module on the paper tape.

If an internal load[1] is to be performed, when the executing program branches to loader entry point 2, the loader compares the program name on the type 01 item on the tape with the name in the loader communication area, and if they do not match, causes a halt to occur with the value 1612 in R1. To clear this halt, the correct paper tape roll must be mounted on the reader. The Execute button is pressed, the identification of the next program is checked, and, if valid, the program is loaded.

[1]During internal loading, the Paper Tape Loader can be accessed by a program in memory. The executing program enters in the loader communication area (see Table B-1) the information required for the next program to be loaded (including the program name), and then branches to entry point 2 of the loader. The loader loads the program whose name is stored in the loader communication area.

APPENDIX C

CARD LOADER

The Card Loader is used to load into memory an image text module punched on 80-column cards in 8-bit ASCII notation. The Card Loader then turns control over to the module loaded.

The Card Loader is read into memory from cards by the bootstrap card (the first card of the Card Loader) that is itself read from the card reader by the firmware bootstrap routine.

The Card Loader which includes the Offline Trap Handler and the program to be loaded, must be stored in sequence in the card reader hopper, since they must be loaded in sequence.

LOADER STRUCTURE AND FUNCTIONS

The Card Loader consists of two subroutines:

- Loader Initialization Routine
- Loader

Loader Initialization Routine

When the bootstrap card finishes bringing the Card Loader into memory, it turns control over to this subroutine. The functions of this subroutine are:

- Save the bootstrap channel in the loader communication area
- Save the two halt addresses in the loader communication area (see Table C-1)
- Load the Offline Trap Handler (see Appendix D)
- Branch to entry point 1 of the loader

Table C-1. Contents of the Loader Communication Area as Used With Card Loader

Entry	Location (Displacement from High Memory Address)	Size (Words)	Description
1	HMA	1	Loader indicator word[a]
2	HMA-2	2	Relocation factor
3	HMA-6	4	File name (program identification)
4	HMA-12	6	Not used for cards
5	HMA-13	1	Bootstrap channel number
6	HMA-14	1	Loader entry point 2
7	HMA-15	1	Loader entry point 1
8	HMA-16	1	Not used for cards
9	HMA-17	1	Not used for cards
10	HMA-19	2	Address of special item halt
11	HMA-21	2	Address of program not found halt
12	HMA-22	1	Not used for cards
13	HMA-23	1	Operator's console channel number

[a]See Table C-2.

Table C-2. Loader Indicator Word Contents as Used With Card Loader

Bit Number	Values
0	0 - Execute after loading (default)
	1 - Halt after loading
1	0 - Execute before returning to caller
	1 - Return to caller after loading
2-4	Reserved
5	0 - Operator's console present
	1 - No operator's console
6-7	Reserved
8	Must be zero
9	0 - Not applicable when Card Loader in use
	1 - Indicates that the Card Loader is in use
10-15	Must be zeros

Loader

The loader reads into memory one card of image text at a time; it analyzes the item type, distributes the code, relocates addresses, and resolves back-patch chains.

BOOTSTRAPPING AND LOADING FROM CARDS

The loading process using the Card Loader requires that the following modules be on cards in the card reader hopper in the order in which they will be loaded:

- Card Loader (which includes Offline Trap Handler)
- Program or component to be loaded

Bootstrapping and loading a program or component from cards, summarized in Figure C-1, consists of these phases (as described in the following subsections):

- Performing one of two alternative bootstrapping procedures
- Performing loading of the program

Bootstrapping Phase for Program on Cards

Control panel operation causes the bootstrap card to be read into memory from the card reader on which it is stored. The LOAD indicator lamp goes on when the Load key is pressed, and will remain on until the bootstrapping routine jumps to the first location of the bootstrap record.

Perform one of the following two bootstrapping procedures as described in the next two subsections:

- Default bootstrapping: Use this procedure if you wish to perform the Quality Logic Test (QLT) and use the default bootstrap device channel number (0400_{16}). (The QLT clears memory; therefore, do not perform the QLT if you need to preserve the contents of memory.)
- Bootstrapping when channel number default and/or QLT not used: Perform this procedure if you wish to change the bootstrap device channel number and/or omit the QLT.

DEFAULT BOOTSTRAPPING

The default bootstrapping procedure is as follows:

1. Perform master clear, automatically setting the P-register to 0000_{16}.
2. Press Load.
3. Press Execute. The QLT will be run, setting register R1 to the default bootstrap device channel number and the P-register to 0002_{16}.
4. Press Execute, causing the bootstrap record to be loaded into memory. The bootstrap record places the default high memory address in register B3, sets registers B2, R4 and R6 to zero, and then halts (halt number 1601 in register R1). Proceed to loading phase.

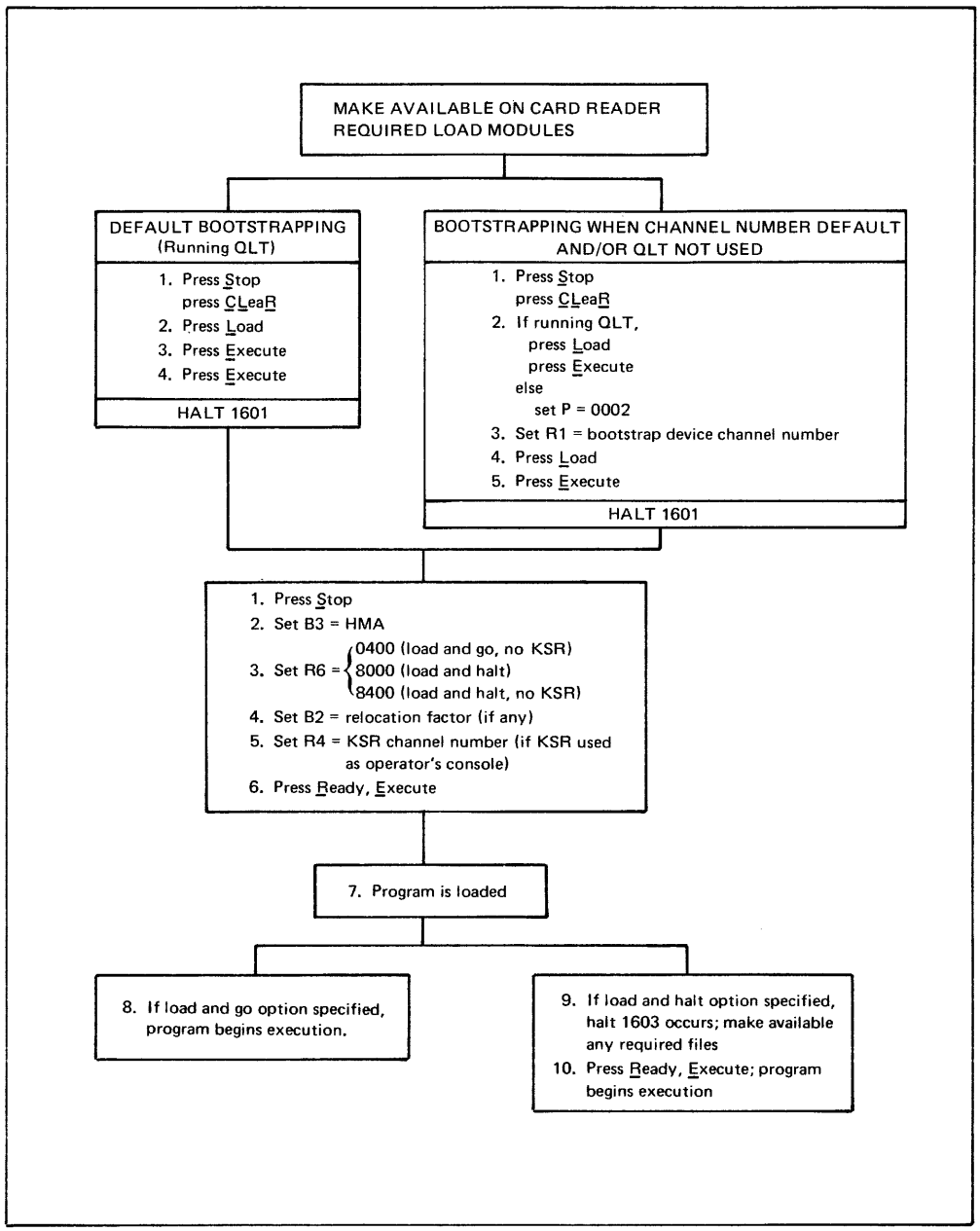


Figure C-1. Procedure for Bootstrapping and Loading a Program From Cards Using Control Panel

BOOTSTRAPPING WHEN CHANNEL NUMBER DEFAULT AND/OR QLT NOT USED

The bootstrapping procedure described below permits you to use a different bootstrap device channel number and/or omit the QLT.

1. Perform master clear.
2. If you are running the QLT, press Load and then Execute (the QLT sets the P-register to 0002_{16}); otherwise, set the P-register to 0002_{16}.
3. Set the bootstrap record device channel number in register R1, as follows:
   a. If you are using a channel number other than the default channel number, place the desired channel number in register R1.
   b. If you are using the default channel number but are not running the QLT, place the default channel number 0400_{16} in register R1. (If you are using the default channel number and are running the QLT, the QLT places the default channel number in register R1.)
4. Press Load.
5. Press Execute, causing the bootstrap record to be loaded into memory. The bootstrap record places the default high memory address in register B3, sets registers B2, R4 and R6 to zero, and then halts (halt number 1601 in register R1). Proceed to loading phase.

Loading Phase for Program on Cards

After you have completed one of the two alternative bootstrapping procedures described above, perform the following loading procedure:

1. At loader halt 1601, press Stop, placing the central processor in the stop state.
2. Enter the high memory address in register B3, unless you are using the default value.
3. Enter one of the following bit masks for the loader indicator word in register R6, unless you are using the default value:

   0400 - Load and go, no KSR

   8000 - Load and halt

   8400 - Load and halt, no KSR

   The load and halt option causes loader halt 1603 to occur, permitting you to make required files available (see step 9 below).
4. Enter in register B2 the relocation factor to be added to all relocatable addresses in the program, unless you are using the default value.
5. If you are using a KSR-like device as an operator's console, enter in register R4 the KSR channel number, unless you are using the default value (0500_{16}).
6. Press Ready, then Execute. The bootstrap card loads the Card Loader, which in turn loads the Offline Trap Handler into memory.

7. The Card Loader loads the program or component which is the next load module in the card reader hopper.
8. If the load and go indicator is set, the loader turns control over to the program or component which begins execution.
9. If the load and halt indicator is set, the loader halts (halt 1603 in register R1). Make available any required files.
10. Press Ready, then Execute. The loader turns control over to the program or component which begins execution.

INVOKING CARD LOADER FROM EXECUTING PROGRAM

A loaded program or component can enter the Card Loader at either of the two loader entry points (as listed in Table C-1). If entered at entry point 1, the loader halts with 1602 in register R1 (see Table C-3 for loader halts). The operator can alter the high memory address stored in register B2 and/or the loader indicator word stored in register R6. Pressing the Execute key causes this information to be placed in the loader communication area and the program identification field to be initialized to spaces. The loader then branches to entry point 2.

Table C-3. Card Loader Halts

Halt Number	Significance	Operator Action
1601	Bootstrap card halt:	
	1. High memory address can be altered.	Enter HMA in B3.
	2. Operator's console channel number can be altered.	Enter KSR channel number in R4.
	3. Relocation factor can be altered.	Enter relocation factor in B2.
	4. Loader indicator word can be set for:	
	a. Load and go, no KSR.	Enter 0400 in R6.
	b. Load and halt.	Enter 8000 in R6.
	c. Load and halt, no KSR.	Enter 8400 in R6.
1602	Loader entry point 1 halt:	
	1. Relocation factor can be changed.	Enter relocation factor in B2.
	2. Loader indicator word can be set for:	
	a. Load and halt.	Enter 8000 in R6.
1603	Loader halt (load and halt option).	Press Execute.

Table C-3 (cont). Card Loader Halts

Halt Number	Significance	Operator Action
1611	Type 01 item not found.	Press Execute to read next card.
1612	Result of program name mismatch.	Press Execute to read cards until next type 01 item card encountered.
1613	End of card deck reached.	Load another card deck containing an image text file.
1614	End-of-file card missing.	Press Execute (ignore the warning).
1615	An unresolved symbol was encountered in the load module.	Press Execute to bypass item.
1616	Loader I/O error: the loader cannot complete physical transfer of the data from this module.	Put the last card read back into the card reader hopper.
1691	Card deck out of sequence.	Correct card deck and reload.
1692	Sequence number field overflow.	Reboot and retry.
1693	Checksum error.	Reboot and retry.
1694	Invalid item type encountered in load module.	Relink the load module and retry the load operation.
1695	Load module size exceeds available memory or relocation factor too large.	Correct the problem and reboot the system.
1696	Bootstrap I/O error.	Verify that the correct card deck is in the card reader hopper and reboot the system.
16FF	Trap occurs during offline program execution (halt number is set in R1 by the Offline Trap Handler).	See Appendix D for operator action.

If an executing program enters the loader at loader entry point 2, it must first store in the loader communication area information required to load the next program (the program name, relocation factor and loader indicator word).

At entry point 2, the loader checks the program identification field of the loader communication area. If the field has spaces in it, the loader loads the next image text file in the card reader hopper. If the field contains a name, it compares that name with the program name in the type 01 item card, and, if they do not match, causes a halt to occur (halt 1612 in register R1). When the Execute key is pressed, the Card Loader bypasses all cards until the next type 01 item card is read. When a program name match is made, the Card Loader loads the specified program.

APPENDIX D
USE OF THE OFFLINE TRAP HANDLER

The Offline Trap Handler, TRPHND, is loaded into memory by the initialization routines of the Disk, Paper Tape, and Card Loaders.

The functions of the trap handler are:

- Set up a register save area (see Table D-1).
- Set up a trap save area (see Table D-2).
- Set up trap vectors 1 through 16.
- Put the halt value 16FF in R1 if a trap occurs during offline program execution.
- Put the trap vector number in R3 if a trap occurs during offline program execution.
- Save registers in register save area if a trap occurs during offline program execution.
- Save information in the trap save area if a trap occurs during offline program execution.
- Cause a halt if a trap occurs during offline program execution.

When the 16FF halt occurs because of a condition that has caused a trap, the easiest way to display memory is to load the offline Debugger and print out locations 0 through 81_{16} either to a printer, or to the system or operator's console.

To load the Debugger, enter the relocation factor in register B2; enter the channel number of the KSR-like device to be used by the Debugger in register R1 (the default is the system console channel number). Enter the channel number of a line or serial printer into register R2. Then press Execute.

Tables D-1 and D-2 show the contents of the register and trap save areas.

Table D-1. Contents of the Register Save Area

Location (Hexadecimal)	Length (Words)	Contents
2A	1	B7
2B	1	B6
2C	1	B5
2D	1	B4
2E	1	B2
2F	1	B1
30	1	I
31	1	R7
32	1	R6
33	1	R5
34	1	R4
35	1	R2
36	1	R1
37	1	M

Table D-2. Contents of the Trap Save Area

Location (Hexadecimal)	Length (Words)	Contents
08	1	Next trap save area pointer
09	1	I-register
0A	1	R3
0B	1	Instruction last executed
0C	1	Z-word (see Executive and Input/Output manual)
0D	1	A-word (see Executive and Input/Output manual)
0E	1	Program counter (P-register)
0F	1	B3

After the Debugger has been used, the system must be rebooted. The debugging procedure can be used even if a trap has not occurred, by causing a halt in the following way:

- Put machine in step mode; i.e., single instruction mode.
- Press CLeaR.
- Change location E0 (the P-register) to 003B.
- Press Ready.
- Press Execute.

This sequence brings the system to a 16FF halt and saves the registers, after which the Debugger can be loaded as described above.

APPENDIX E
PROCEDURE FOR DUMPING MEMORY USING MDUMP BOOTSTRAP RECORD

If a problem occurs during program execution, you can dump memory to a special diskette file using the bootstrap record named MDUMP. Later, you can print this memory dump from the diskette file, using the Dump Edit utility program.

Before loading a program for which a memory dump may be required, prepare the diskette on which the memory dump is to be stored:

1. Initialize the diskette volume DMPVOL and allocate space on it for the relative file DUMPFILE that is to contain the memory dump.[1]
2. Use Bootstrap Generator to create the MDUMP bootstrap record and place the record on the diskette DMPVOL on which the file DUMPFILE resides.

These steps are described in detail in the Bootstrap Generator section of the Utility Programs manual. Bootstrap Generator operating procedures appear in Part III of the Operator's Guide.

Once an executing program encounters a problem or a halt occurs, you can obtain a memory dump by taking these actions (as described in the following subsections):

1. Save the contents of registers and store in memory the size of the trap save area.
2. Bootstrap MDUMP, which then performs the memory dump to the diskette file DUMPFILE.
3. Use Dump Edit to print all or a portion of the memory dump from DUMPFILE.

PROCEDURE FOR SAVING THE CONTENTS OF REGISTERS BEFORE DUMPING MEMORY

When the executing program halts or aborts, follow this procedure for saving the contents of registers and the size of the trap save area. (This information is printed out in edited format by the Dump Edit utility program.)

[1] The size of DUMPFILE (in sectors) must be equal to 64 times the number of 4K modules of memory to be dumped.

1. Display the contents of the P-register (E0) and S-register (E7) and record them manually.
2. Press Stop.
3. Change the contents of the memory address register (A0) to 0000_{16}.
4. Press Write, then Plus 1, preparatory to writing in memory locations 0001_{16} through 0005_{16}.
5. Change the contents of the memory data register (B0) to $002A_{16}$. Press Execute, causing the value $002A_{16}$ to be stored in memory location 0001_{16}. (The address $002A_{16}$ is the starting location of the memory area into which the SAVE instruction loads the contents of the registers.)
6. Change the contents of B0 to $FFFF_{16}$ and press Execute, causing the mask (FFFF) for the SAVE instruction to be stored in location 0002_{16}.

   NOTE: To save in memory the contents of the P- and S-registers and the size of the trap save area, perform steps 7 through 9.

7. Change the contents of B0 to the value of the contents of the P-register, as recorded in step 1 above. Press Execute, causing the contents of the P-register to be stored in location 0003_{16}.
8. Change the contents of B0 to the value of the contents of the S-register, as recorded in step 1 above. Press Execute, causing the contents of the S-register to be stored in location 0004_{16}.
9. Change the contents of B0 to the size of the trap save area. Press Execute, causing the TSA size to be stored in location 0005_{16}.
10. Press Stop, CLeaR, and Execute, resetting the write mode, preparatory to performing a SAVE instruction.
11. Change the instruction register (D0) to $8F00_{16}$ (representing the SAVE instruction).
12. Press Execute. The SAVE instruction is executed: the contents of all machine registers are saved in memory, starting at location $002A_{16}$.

PROCEDURE FOR BOOTSTRAPPING MDUMP AND PERFORMING MEMORY DUMP

Perform the following procedure to bootstrap the MDUMP bootstrap record into memory; MDUMP then transfers the amount of memory image specified to the diskette file DUMPFILE.

1. Mount the diskette containing MDUMP and DMPVOL on the channel to be used in bootstrapping.
2. Press CLeaR to clear the instruction register (D0).
3. Set the P-register to 0004_{16}.
4. Enter in register B1 the initial address of the 64-word memory area into which MDUMP is to be read.
5. Enter in register R1 the channel number of the bootstrap device (see step 1 above).

6. Press Load, then Execute. The bootstrap record MDUMP is read into memory and a halt occurs. At this halt, register R1 contains the channel number of the diskette named DMPVOL; this diskette is now used as the output device.

7. Press Stop.

8. Enter in register R2 the value $000n_{16}$, where n represents one of the following, depending on the size of the area to be dumped:

n	Memory Area to be Dumped (Words)
0	4K
1	8K
2	12K
3	16K
:	:
7	32K
:	:
F	64K

NOTE: It is the operator's responsibility to ensure that enough space has been allocated to DUMPFILE to contain the amount of memory to be dumped.

9. Press Ready, then Execute. The MDUMP record performs a memory dump, causing the amount of memory image specified in step 8 above to be written on DUMPFILE. The dump is complete when an end-of-job halt occurs (see the table of halts later in this appendix).

MDUMP HALTS

No messages are issued during execution of MDUMP. If a halt occurs during execution, the contents of the P-register and R1 register must be displayed to determine the significance of the halt, as indicated in Table E-1.

Table E-1. MDUMP Halts

Register Contents		Condition	Operator Action
P-register	R1 register		
0036_{16}[a]	$= 0000_{16}$	End of job	No operator action required. For information only.
0036_{16}[a]	$\neq 0000_{16}$	Disk write error has occurred	Perform bootstrapping of MDUMP again.

[a] Address relative to the initial address of MDUMP as stored in memory.

PRINTING MEMORY DUMP USING DUMP EDIT

If an installation that includes a printer is available, you can use the Dump Edit utility program to print the memory dump. See the Utility Programs manual for a description of Dump Edit and Dump Edit commands.

Operating procedures for Dump Edit and messages issued by Dump Edit are presented in Part III of the Operator's Guide. Loading and execution of Dump Edit is summarized below.

1. Mount the disk volume(s) containing the Command Processor and Dump Edit, if necessary.
2. Mount the diskette volume named DMPVOL; the memory image obtained during the memory dump must be stored in the file DUMPFILE on DMPVOL.
3. Load the Command Processor.
4. Submit the following commands to the Command Processor:

   ```
   AT 01,DSKcccc,DUMPFILE
   AT 02,LPTcccc
   DPEDIT
   ```

5. Dump Edit is loaded and issues this message on the console:

   ```
   DPEDIT OF nnK DUMP, READY FOR "GO"
   ```

6. Submit GO command (including optional operands) to Dump Edit. (For the format of the GO command, see the Utility Programs manual.)
7. Dump Edit causes the printing of the memory dump from DUMPFILE. When printing is complete, Dump Edit issues these messages:

   ```
   DUMP COMPLETE
   DPEDIT OF nnK DUMP, READY FOR "GO"
   ```

8. Enter QT command; Dump Edit returns control to the Command Processor.

APPENDIX F
LOADING PROGRAMS USING BASIC CONTROL PANEL

The basic control panel (described in Part III) can be used to load a program from disk or paper tape. No operator intervention is possible during bootstrapping and loading. This loading procedure can be performed only under these circumstances:

- No halts can occur during bootstrapping and loading.
- No operator entries through the control panel are required. All bootstrapping and loading parameters required to load the program must have been preset in the applicable bootstrap record when it was created by Bootstrap Generator.

(The procedure for creating a bootstrap record using the Bootstrap Generator utility program is described in the Utility Programs manual.)

Once the operator has mounted the required load modules, unlocked the control panel, and powered up the system, he need press only the initialize switch to cause the program to be bootstrapped and loaded and automatically to begin execution.

The specific steps in bootstrapping and loading a program using the basic control panel are given below.

1. Before loading a program from disk or paper tape, check whether the bootstrap record on that disk or paper tape contains all of the parameters necessary to delineate that load environment. If not, create an appropriate bootstrap record using the Bootstrap Generator utility program. Submit to Boostrap Generator parameter values that specify the load environment in which program loading is to occur.

   Specify values for the bootstrap halt (BTHLT) and load and halt (LDHLT) indicators so that halts 1601 and 1603 will not occur during bootstrapping and loading. If the bootstrap record is to be placed on paper tape, specify a value (either default or non-default) for the relocation factor (REL) and specify the value N for the load and halt (LDHLT) indicator to prevent an automatic halt 1602 from occurring during loading.

   The parameters to be submitted to Bootstrap Generator when the Configuration Load Manager is to be loaded without operator intervention are listed in the Planning and Building an Online Application manual.

2. Make available on the appropriate device(s) the load modules required for loading the program. Table F-1 lists, for disk and paper tape, the load modules required when the Configuration Load Manager, the Command Processor, and offline application programs are to be loaded using the basic control panel. When you are loading from paper tape, it is recommended, but not necessary, that all load modules be on the same paper tape strip (see "Paper Tape Loading" in Section 3 of Part II). The device from which the bootstrap record is to be read into memory must be connected to channel number 0400_{16}.

3. Unlock the basic control panel and power up the system, as follows:

   a. Place panel security switch in unlocked position.

   b. Place power switch in power on position. After dc power is attained, DC ON indicator illuminates.

4. Press initialize switch, causing:

   a. System initialization

   b. QLT execution

   c. Bootstrapping of the bootstrap record from the device on channel number 0400_{16}.

   The bootstrap record causes bootstrapping of the appropriate loader, which loads the Offline Trap Handler and then loads the desired program into memory. If the Disk Loader is loaded, it automatically loads the program whose name has been preset in the bootstrap record (see step 1 above). If the Paper Tape Loader is loaded, it automatically loads the next program mounted on the paper tape reader.

   In either case, the loader loads the program into memory, relocating it using the value of the relocation factor preset in the bootstrap record (the actual memory location used for loading is determined by a combination of the Assembler ORG value, linker BASE value, and loader relocation factor). The loader then turns control over to the program at the program start address. If the program is an offline application, it begins execution. If it is the CLM, it configures and loads the online application, which then begins execution.

Table F-1. Load Modules Required When Program Is Loaded Using Basic Control Panel

Program To Be Loaded	Required Load Modules	
	Disk	Paper Tape[a]
Configuration Load Manager (CLM)	Volume S20002 that includes: 1. Bootstrap record and boot loader 2. PROGFILE containing: a. Disk Loader b. Offline Trap Handler c. CLM and CLM modules[c]	Bootstrap record Paper Tape Loader Offline Trap Handler CLM CLM modules[b]
Command Processor	Volume that includes: 1. Bootstrap record and boot loader 2. Disk Loader 3. Offline Trap Handler 4. Command Processor[d]	
Offline Application	Volume that includes: 1. Bootstrap record and boot loader 2. PROGFILE containing: a. Disk Loader b. Offline Trap Handler Volume that contains offline application, if it is a separate volume	Bootstrap record Paper Tape Loader Offline Trap Handler Offline application program

[a] Load modules must be mounted on paper tape reader in order specified.

[b] For the order in which CLM modules must be mounted on paper tape, see "Order of Loading CLM Modules" in Section 3 of Part IV. For the load modules that must be mounted on the paper tape reader once CLM is loaded, see "Making Load Modules Required for Online Application Available on Paper Tape" in Section 3 of Part IV.

[c] For the load modules required on disk once CLM is loaded, see "Making Appropriate Load Modules Available for Online Application" in Section 2 of Part IV.

[d] For the procedure for loading a system program using the Command Processor, see Section 1 in Part III.

APPENDIX G
EXPLANATION OF ERROR MESSAGE CODES

The code used in coded error messages issued to the operator has the following format (in hexadecimal notation):

(xxyyzz)

xx - Component number

yy - Error category

zz - Specific error code

The component number xx can have the following values:

Component Number	Component
10	Assembler
11	Linker
12	Utility sets
13	Configuration Load Manager
14	FORTRAN Compiler
15	FORTRAN object time routines
16	Loaders
17	Command Processor
18	Cross-Reference Program
19	Editor
20	Operator Interface Manager
21	Program Patch
22	File Manager
23	Macro Preprocessor
25	Dump Edit
26	COBOL Compiler
27	COBOL object time routines
30	Bootstrap Generator
80	Communications
81-FF	User components

The error category code yy can have the following values:

Category Code	Cause	Comments
00	Logical I/O (IOS) error	Code can be followed by message and logical file number
01	Physical I/O error detected by driver	Code is followed by channel number and software status word
02	File Manager error	
0F	Component-specific error	Text optional

Loaders and the Configuration Load Manager may display error conditions to the operator in register R1. The four digits displayed in this case are xxzz (where xx can be 13 for the CLM or 16 for a loader) and zz is specifically defined under the relevant component description.

HONEYWELL INFORMATION SYSTEMS

Technical Publications Remarks Form

TITLE	SERIES 60 (LEVEL 6) GCOS/BES2 OPERATOR'S GUIDE	ORDER NO.	AU46, REV. 0
		DATED	JULY 1976

ERRORS IN PUBLICATION

SUGGESTIONS FOR IMPROVEMENT TO PUBLICATION

Your comments will be promptly investigated by appropriate technical personnel and action will be taken as required. If you require a written reply, check here and furnish complete mailing address below. ☐

FROM: NAME _____ DATE _____

TITLE _____

COMPANY _____

ADDRESS _____

_____

FIRST CLASS
PERMIT NO. 39531
WALTHAM, MA
02154

Business Reply Mail
Postage Stamp Not Necessary if Mailed in the United States

Postage Will Be Paid By:

HONEYWELL INFORMATION SYSTEMS
200 SMITH STREET
WALTHAM, MA 02154

ATTENTION: PUBLICATIONS, MS 486

Honeywell

Honeywell

Honeywell Information Systems
In the U.S.A.: 200 Smith Street, MS 486, Waltham, Massachusetts 02154
In Canada: 2025 Sheppard Avenue East, Willowdale, Ontario M2J 1W5
In Mexico: Avenida Nuevo Leon 250, Mexico 11, D.F.

16260, 2876, Printed in U.S.A.

AU46. Rev. 0