



RTE-A

System Generation and Installation Manual

**Data Systems Division
1266 Kifer Road
Sunnyvale, CA 94086-5304**

Manual Part No. 92077-90034
E0887

Printed in U.S.A. August 1987
Fourth Edition

HP Computer Museum
www.hpmuseum.net

For research and education purposes only.

NOTICE

The information contained in this document is subject to change without notice.

HEWLETT-PACKARD MAKES NO WARRANTY OF ANY KIND WITH REGARD TO THE MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Hewlett-Packard shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance, or use of this material.

Hewlett-Packard assumes no responsibility for the use or reliability of its software on equipment that is not furnished by Hewlett-Packard.

This document contains proprietary information which is protected by copyright. All rights are reserved. No part of this document may be photocopied, reproduced, or translated to another language without the prior written consent of Hewlett-Packard Company.

Copyright © 1983,1985-1987 by HEWLETT-PACKARD COMPANY

Printing History

The Printing History below identifies the edition of this manual and any updates that are included. Periodically, update packages are distributed which contain replacement pages to be merged into the manual, including an updated copy of this printing history page. Also, the update may contain write-in instructions.

Each reprinting of this manual will incorporate all past updates; however, no new information will be added. Thus, the reprinted copy will be identical in content to prior printings of the same edition with its user-inserted update information. New editions of this manual will contain new information, as well as all updates.

To determine what manual edition and update is compatible with your current software revision code, refer to the Manual Numbering File or the Computer User's Documentation Index. (The Manual Numbering File is included with your software. It consists of an "M" followed by a five digit product number.)

First Edition	Jun 1983	
Update 1	Aug 1983	Replace Appendix G
Reprint	Aug 1983	Update 1 Incorporated
Update 2	Aug 1983	
Reprint	Dec 1983	Update 2 Incorporated
Update 3	Jun 1984	Add 15-Mb disc, MAC disc, 9144, 7974, 2566
Second Edition	Jan 1985	
Third Edition	Jan 1986	
Update 1	Oct 1986	New Drivers and Peripherals
Fourth Edition	Aug 1987	Rev. 5000 (Software Update 5.0)

Preface



This manual provides information for generating and installing a new RTE-A Operating System. The information consists of the following:

1. General descriptions of the complete process of creating a new RTE-A Operating System, with an overview of each of the two major steps – system generation and installation.
2. How to run the system generator program RTAGN.
3. How to prepare the generator answer file required for RTE-A generation.
4. Descriptions of generator command syntax and the effects of each of the commands.
5. How to install a new RTE-A Operating System, replacing an existing RTE-A Operating System.
6. How to prepare and install a memory-based RTE-A system.

The appendices provide examples of blank generation worksheets, generation and installation error messages, specific requirements for configuring various devices into the RTE-A Operating System, requirements for generating and installing localizable systems, and other reference data.

Who should read this manual?

Anyone who needs to perform any of the following:

Prepare a system generation answer file.

Generate a new RTE-A Operating System from an existing RTE-A Operating System.

Install a new RTE-A Operating System.

Boot up a new operating system.

Create a memory-based RTE-A system.

Refer to the System Design Manual for a conceptual discussion of the RTE-A Operating System. Each subsystem also has its system generation and installation requirements. Refer to the associated subsystem manual for details.

Table of Contents

Chapter 1 General Information

Introduction	1-1
Type 6 Program Files	1-8

Chapter 2 Running the Generator

General Information	2-1
The RTAGN Runstring	2-1
Command File	2-2
List File	2-2
System File	2-3
Snapshot File	2-3
Overlaying Files	2-4

Chapter 3 Initialization Phase

Conserving Space	3-2
Links	3-3

Chapter 4 System Relocation Phase

Relocating Modules	4-1
System Modules	4-1
Driver Modules	4-2
Libraries	4-2
System Relocation Phase Commands	4-3
RELOCATE Command	4-3
SEARCH Command	4-4
MSEARCH Command	4-4
LOCC Command	4-4
BLOCC Command	4-5
DISPLAY Command	4-5
LENTRIES Command	4-6
ALIGN Command	4-6
END Command	4-6
System Partitioning Concerns	4-7

TG Command	4-7
Partitioned Command	4-9

Chapter 5 OS Module/Driver Partition Phase

General Description	5-1
ALIGN Command	5-7
Partition Configurations	5-8

Chapter 6 Table Generation Phase

General Description	6-1
Interface and Device Tables	6-1
Node Lists	6-2
Interrupt Table	6-3
The System I/O Configuration Worksheet	6-4
The IFT/DVT Worksheet	6-5
Making Use of Defaults	6-8
IFT and DVT Parameter Defaults	6-8
Building the Answer File	6-10
Table Generation Phase Commands	6-10
IFT Command	6-11
IFT Example	6-13
DVT Command	6-14
END Command	6-19
NODE Command	6-20
END Command	6-21
INT Command	6-21

Chapter 7 Memory Allocation Phase

Allocating Class Numbers	7-1
Allocating Resource Numbers	7-2
Allocating Debug Table Entries	7-2
Allocating ID Segments	7-2
Allocating Memory Descriptors	7-3
Allocating System Available Memory (SAM)	7-4
Allocating Extended System Available Memory (XSAM)	7-4
Specifying Spool Buffer Limits	7-5
Specifying Background Priority and Timeslice Quantum	7-5
Specifying Shared Programs	7-5
Allocating an Extended Schedule Table	7-6
Specifying System Memory Block	7-6
Specifying Concurrent Users	7-6
Specifying LOGOF buffer limits	7-7
Setting Up System Common	7-7
Allocating System Message Block	7-8
Specifying Security Tables	7-8
Specifying System Default Libraries	7-9
Memory Allocation Phase Commands	7-10
Class Number Command CLAS	7-10
Resource Number Command RESN	7-11
ID Segment Command ID	7-11
Memory Descriptor Command RS	7-12
System Available Memory Command SAM	7-12
Spooling Limits Command SL	7-13
Background Priority Command BG	7-13
Quantum Timeslice Command QU	7-14
Shared Program Command SP	7-14
System Memory Block Command MB	7-14
Concurrent User Command US	7-15
LOGOF Buffer Limit Command LB	7-15
Labeled Common Commands	7-15
Unlabeled (Blank) Common Command COM	7-16
System Message Catalog Relocation	7-16
Library Command LIB	7-17

Chapter 8 System Installation Concepts and Considerations

Types of Systems	8-1
Host and Target Systems	8-2
Type 6 Program Files	8-2
The Boot Process	8-3
Boot From Disc	8-3
Boot From Cartridge Tape Drive	8-3
Boot From PROM	8-3
Boot From DS Link	8-4
Target System Installation	8-5

Chapter 9

Disc-Based System Installation Procedure

Prepare the Boot Command File	9-1
Echoing Commands	9-2
Mirrored Disc Configuration	9-2
Specifying the System	9-2
Specifying Memory Size	9-3
Specifying Bad Pages	9-3
Mounting Disc Volumes	9-4
Setting Up the Swap File	9-4
Specifying the Scratch LU	9-5
Turning Session Accounting On/Off	9-5
Setting System Parameters	9-6
Setting Master Security Code	9-6
Restoring Programs	9-6
Reserved Partition Definition	9-7
Terminating Input	9-9
Sample Boot Command File	9-9
Installing System, Snapshot, and Command Files	9-10
Installation Considerations	9-10
FPUT and BOOTEX for New Disc Configurations	9-11
Preparation for Boot	9-13
Establishing New CI Directories & Programs	9-15
SYSTEM Directory	9-16
HELP Directory	9-16
CATALOGS Directory	9-16
LIBRARIES Directory	9-16
PROGRAMS Directory	9-16
SCRATCH Directory	9-17
CMDFILES Directory	9-17
CI Commands Implemented in Separate Programs	9-17
Format System	9-18
Boot the Target System	9-19
Checking the Target System for Proper Operation	9-21
Setting Up the Startup Program & WELCOME File	9-22
CM as a Secondary Program	9-22
Backing Up the Target System	9-23
Setting Switches for Autoboot	9-24
Installing Optional Subsystems	9-24
Installing DS Transparency Software	9-25

Chapter 10 Installing a Memory-Based System

Assembling the Required Files	10-1
Run BUILD to Create the Merged System File	10-2
Operational Phases	10-3
Sample Command File	10-5
Installing Merged System File on a Bootable Medium	10-7
PROM	10-7
CTD	10-7
Magnetic Tape	10-7
Disc Volume	10-7
Directly from FMGR Cartridge	10-8
From Disc, Using BOOTEX	10-8
Booting the Target System	10-9
Testing the Target System	10-12
Backing Up the Target System	10-12
Setting Switches for Automatic Boot	10-13

Chapter 11 Multiuser and Spooling Setup

Group and User Management Program (GRUMP)	11-1
Establishing Multiuser Accounts	11-5
PROMT Program	11-6
LOGON Program	11-6
Spooling Initialization and Use	11-7
SP Program	11-10
SMP Program	11-10
SPGET Program	11-10
OUTPT Program	11-10
Shared CI	11-11
CDS Library	11-11

Appendix A Error Messages

RTAGN Errors	A-1
BUILD Program Error Messages	A-9
VCP Loader Errors	A-12

Appendix B Worksheets

Appendix C Configuring a Disc Subsystem

Disc Subsystem Categories	C-2
Disc Configuration Steps	C-3
Configuring Non-CS/80 Discs	C-4
Physical Disc Structure	C-4
Logical Disc Structure	C-4
Surface Mode	C-4
Cylinder Mode	C-4
File System Considerations	C-6
Non-CS/80 Disc Driver DD*30 Parameters	C-7
HP-IB Address (DP1)	C-7
Disc Drive Unit Number (DP2)	C-7
Starting Head for LU (DP3)	C-8
Starting Cylinder for LU (DP4)	C-8
Number of Spares for LU (DP5)	C-8
Number of Tracks for LU (DP6)	C-8
Number of Blocks per Track for Disc (DP7)	C-8
Number of Surfaces for LU (DP8)	C-9
Configuring CS/80 Discs	C-10
CS/80 Driver Parameters	C-10
HP-IB Address (DP1)	C-11
Disc Drive Unit Number (DP2)	C-12
Starting Block of Disc (DP3,DP4,DP5)	C-12
Number of Tracks for LU (DP6)	C-12
Number of Blocks per Track for Disc (DP7)	C-13
Reserved Parameter (DP8)	C-13
Starting Block Address of Disc Cache	C-13
Address of First Cache Block	C-13
Sample CS/80 Worksheets	C-14
Sample 7907 Worksheet, Fixed Disc (Unit 0)	C-14
Sample 7907 Worksheet, Removable Disc (Unit 1)	C-14
Sample 7908 Worksheet	C-14
Sample CTD Worksheet	C-15
Sample 7911 Worksheet	C-15
Sample 7912 Worksheet	C-15
Sample 7914 Worksheet	C-16
Sample 7933/35 Worksheet	C-16
Sample 7941/7942 Worksheet	C-17
Sample 7945/7946 Worksheet	C-17
Sample 9122D Worksheet	C-17
Sample 9133H Floppy Worksheet	C-18
Special Considerations for the 7902 Disc Drive	C-19
Sample 7902 Disc Worksheet	C-20
7902 Disc Worksheet Sample	C-20
Special Considerations for Model 6 Integrated Disc Drives	C-21
Sample Model 6 Integrated Disc Worksheet	C-21
Sample 9133/9134/9121 Disc Worksheet	C-22

Microsystems with Integrated Peripherals (248x)	C-23
Address and Unit Specification on Microsystems	C-24
Special Considerations for the 7906M Disc Drive	C-25
Using the Worksheet	C-26
7906M Disc Configuration Worksheet Cylinders:	C-26
7906 Disc Configuration Worksheet	C-27
Special Considerations for MAC Drives	C-28

Appendix D GEN Records

Parameter Default Records	D-1
Processing	D-2
Sample Default File	D-2
Format of the Default GEN Record	D-2
System Partitioning Records	D-4
Processing	D-4
Entry Records	D-4
Module Flag Records	D-4
Format of System Partitioning GEN Records	D-5
Entry Records	D-5
Module Flag Records	D-5

Appendix E Device and Interface Driver Tables

Appendix F Default DVT Entries by Driver Name

Appendix G Sample System Generation Answer File

Appendix H VCP Boot Information

VCP Errors	H-1
General Boot Procedure	H-1
Loading Your System from Disc	H-3
Loading Your System from CTD	H-9
Loading Your System from PROM	H-11
Loading Your System over a DS Link	H-11
Loading Your System from Magnetic Tape	H-13
Loading Your System from 248X Integrated Discs	H-15

Appendix I BUILD Program

Introduction	I-1
BUILD Program Operation	I-2
The BUILD Runstring	I-2
BUILD Examples	I-6
BUILD Command File	I-11

Appendix J Boot Extension BOOTEX

Boot Program Operation	J-1
ECHO Command	J-2
MI Command	J-2
SYSTEM/SNAP Commands	J-3
Memory Commands	J-3
Mounting Cartridges	J-3
Scratch LU	J-4
Bad Pages	J-4
Restore Program Commands RP, ST, SZ, PR	J-5
Reserved Partition Assignment	J-6
Reserved Partition Definition	J-6
SWAP File Definition	J-6
Timeslicing, Swapping, Security Code, Session Accounting	J-7
END Command	J-7
Sample Boot Command File	J-8
Skipping the Boot Program	J-8

Appendix K CS/80 Disc Configuration

Appendix L Installing Localizable Programs

Localized Programs	L-1
File Naming Conventions	L-2
Language ID Numbers	L-2
Hierarchical File System (CI) File Names	L-2
FMGR File Names	L-2
System Generation and Installation	L-3
System Design Considerations	L-3
Terminals	L-3
Memory Requirements	L-3
System Generation	L-3
Relocating the Appropriate Driver	L-4
Configuring Terminals for 8-Data Bit Support	L-4
Relocating the System Message Relocatables	L-4
System Installation	L-5
Placing Binary Message Catalogs into /CATALOGS	L-5
Installing the Message Monitor (If Required)	L-6
Placing Help Files xxxxx.Hnnn in the /SYSTEM Directory	L-6
Creating Directory /HELPxxx for Localized Help Files	L-6
Placing Localized Programs into /PROGRAMSxxx	L-6
Placing Unlocalized Programs in Directory /PROGRAMS	L-6
Setting Up NLTERMCAP.TXT	L-6
Using NLSID to Set Terminal Language ID	L-7
Preparing the Welcome File	L-8
Booting Up the New System	L-8
Localizable System and Subsystem Files	L-8
Error Messages	L-11

List of Illustrations

Figure 4-1.	Results of Using the LOCC Command	4-4
Figure 5-1.	Sample Memory Maps	5-6
Figure 5-2.	Sample Partitions	5-8
Figure 6-1.	Example Node List Configuration	6-3
Figure 10-1.	Memory Map of Sample Memory-Based System	10-6
Figure C-1.	Arrangement of Disc Cylinders	C-5
Figure C-2.	Track Access in Cylinder Mode	C-6
Figure C-3.	7902 Dual Flexible Disc Drive	C-19
Figure C-4.	7906M Disc Drive	C-25
Figure C-5.	Example MAC Disc Generator Answer File Statements	C-29

Tables

Table 5-1.	Partitionable Driver List	5-2
Table 5-2.	Partitionable System Module Sizes and Tag Area Requirements ..	5-4
Table 10-1.	Processor Card Switch Settings	10-13
Table E-1.	Device/Driver Correspondence	E-3
Table E-2.	Standard Generation File Entries	E-5
Table E-3.	Device and Interface Driver Tables	E-7
Table E-4.	Flexible Disc Default File Values	E-8
Table E-5.	9133A/9134A Disc (Four-Volume Format) Default File Values ..	E-9
Table E-6.	9133B/9134B Disc (Single-Volume Format) Default File Values ..	E-10
Table E-7.	9133D/H/L 9134D/H/L Default File Values	E-11
Table E-8.	9133D/H/L Dual-Sided Dbl. Density Microfloppy Def. File Values	E-12
Table E-9.	9122D Dual-Sided Double Density Microfloppy Def. File Values	E-13
Table E-10.	9121/9133A/B/XV Single-Sided Microfloppy File Default Values ..	E-14
Table E-11.	Mag Tape Default File Values	E-15
Table E-12.	7906 Default File Values	E-16
Table E-13.	7906M Default File Values	E-17
Table E-14.	7920 Default File Values	E-19
Table E-15.	7920M Default File Values	E-21
Table E-16.	7925 Default File Values	E-23
Table E-17.	7925M Default File Values	E-25
Table E-18.	Cartridge Tape Drive File Default Values	E-27
Table E-19.	7907A Removable Disc Default File Values	E-28
Table E-20.	7907/08/11/12/14/33/41/42/45/46 Default File Values	E-29
Table E-21.	248X Default File Values	E-31
Table E-22.	IFT Default File Values	E-33
Table E-23.	7936 Default File Values	E-34
Table E-24.	7937 Default File Values	E-36
Table E-25.	7957 Default File Values	E-38
Table E-26.	7958 Default File Values	E-40
Table H-1.	VCP Default Parameters for Boot	H-3
Table L-1.	Localizable System and Subsystem Files	L-9

General Information

Introduction

This manual provides the information required to generate and install a new RTE-A Operating System. The generation process produces the system and snapshot files required to define the system to be installed. Generation does not affect the operation of the existing RTE-A system. The system installation process makes the newly generated system operational.

The information is organized as follows:

- Chapter 1 provides an overview of the RTE-A system generation and system installation processes. It also describes the conventions used in this manual.
- Chapter 2 describes how to run the RTE-A generator program RTAGN and gives the RTAGN runstring syntax and describes the output files.
- Chapters 3 through 7 describe the operation of the system generator program (RTAGN) and the generator commands. These chapters correspond to the various phases in the generation process.
- Chapter 8 contains general information on the system installation process.
- Chapter 9 provides the installation procedure for a disc-based RTE-A system.
- Chapter 10 provides the installation procedure for a memory-based RTE-A system.
- Chapter 11 describes the multiuser and spooling features provided with the optional HP 92078A Virtual Code+ (VC+) Package.
- Appendix A contains the generation and installation error messages.
- Appendix B provides the generation worksheets used in preparing the generation answer file.
- Appendix C provides procedures for configuring a disc subsystem.
- Appendix D and Appendix E contain the generation requirements for drivers.
- Appendix F lists default DVT entries by driver name.

Appendix G contains a sample generation answer file.

Appendix H contains system boot information.

Appendix I describes the BUILD program used for creating a memory-based system.

Appendix J describes the BOOTEX program.

Appendix K contains procedures for reconfiguring CS/80 discs.

Appendix L describes the requirements for generating and installing localizable systems.

System Generation

System generation consists of preparing a system generation answer file and running the system generator program RTAGN. The system generation process is shown in Figure 1-1. The system configuration planning portion includes following the instructions given in Chapters 3 through 7 of this manual with the aid of the worksheets and sample answer files provided. After the appropriate worksheets have been filled out, an existing answer file can be edited according to the worksheets to create a new system generation answer file. Then the system generator program RTAGN is run using the new answer file to create a system, a snapshot, and a list file. The system and snapshot files are used to install the new operating system.

Before system generation can be started, you must have the RTE-A software available in the existing RTE operating system. If the RTE-A software is furnished on CS/80 cartridge tape, magnetic tape, or flexible discs, you must restore the software onto the existing RTE operating system. The media restoration procedures are given in the RTE-A Primary System Software Installation Manual (92077-90038) and the utilities used are described in the RTE-A Utilities Manual (92077-90004).

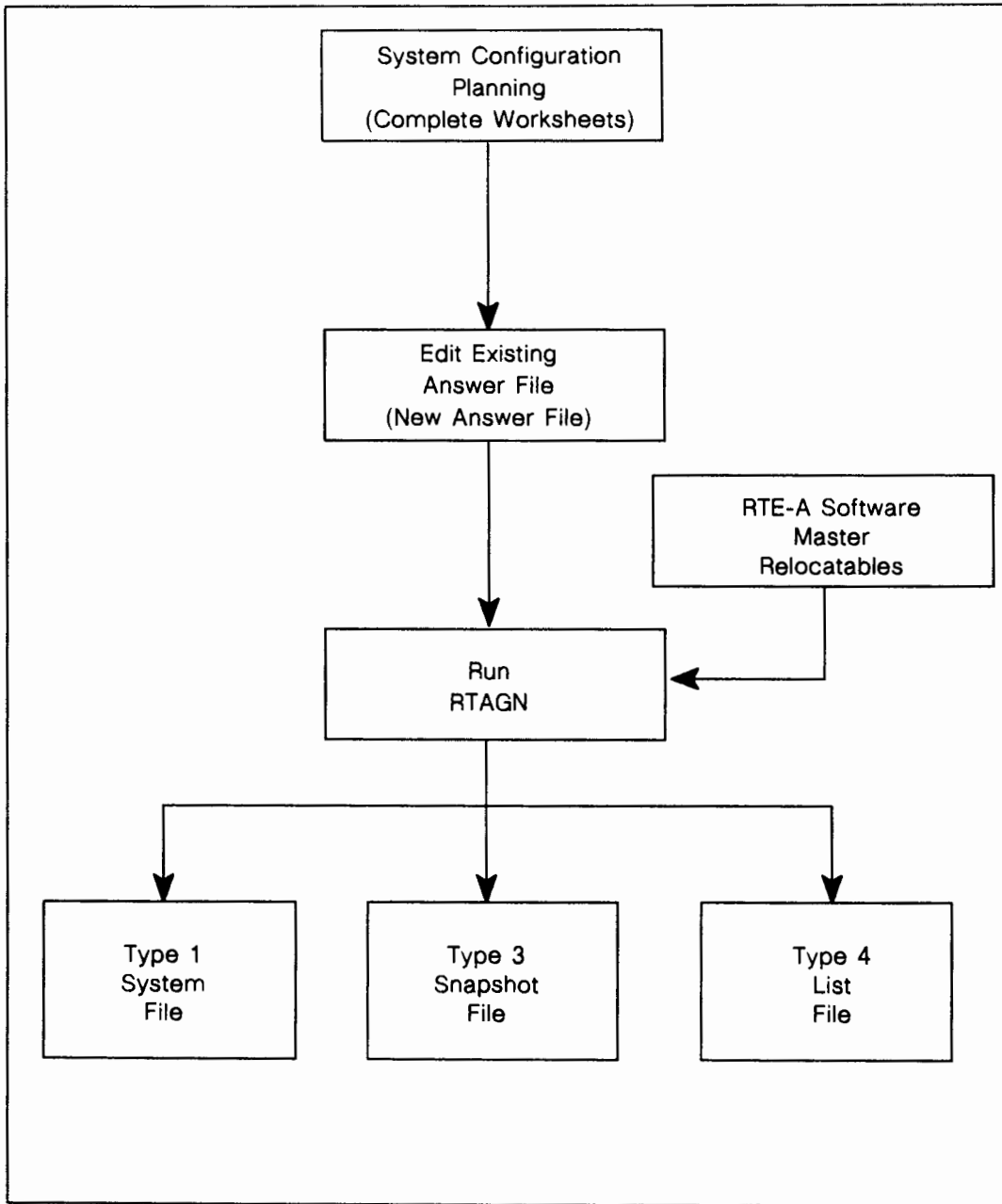


Figure 1-1. RTE-A System Generation Process

Running RTAGN

The system generator program RTAGN executes in five functional phases: initialization, system relocation, OS module/driver partition, table generation, and memory allocation. The generator accepts only those commands expected in each phase of execution.

1. *Initialization Phase.* In this phase, the generator initializes the tables, variables, and parameters used by the operating system.
2. *System Relocation Phase.* During this phase, the specified relocatable modules of the operating system are relocated.
3. *OS Modules/Driver Partition Phase.* In this phase, system modules and I/O drivers are relocated and grouped into physical partitions. Partitioning allows more system table space and the use of more system code, including large drivers.
4. *Table Generation Phase.* In this phase, space for various system tables is allocated and the system tables are configured.
5. *Memory Allocation Phase.* During this phase, space is allocated for ID segments, system available memory, system memory block, the maximum number of classes, resource numbers and memory descriptors, system common, and error message relocation. The number of shared programs, number of concurrent users, swapping limit, timeslice value, and default libraries to be searched are also defined.

Generation Answer File

A system generation answer file contains the commands needed to build an operating system using the system generator program RTAGN. Answer files are commonly constructed by editing an existing file, such as the primary system answer file listed in Appendix G of this manual. The primary system answer file is available on your primary system under the name #ANS.

The command entries in the answer file direct the generator operations. Comments also are included in the file to enable the reader to follow the generation process.

Commands

The generator requires commands in a specific order. In this manual, the commands are described in the order expected by the generator. If this order is not followed, the generator issues an error message.

In some cases, the generator accepts any one of a group of commands, as in the various relocation sections. These are terminated with the END command. An error message is issued if a command encountered by the generator is not one that is expected.

All generator commands can be entered with only two characters. The full command name may be used for clarity; for example, you may enter either LOCC or LO, BLOCC or BL.

Comments

Comments may be used anywhere in the answer file. These are particularly helpful when generating a new system since an existing answer file generally is edited to create the new answer file. A well-commented answer file simplifies the editing process. An asterisk (*) in the first column of an entry indicates that the entire line is a comment line. This line will be written to the list file.

Comments may also appear at the end of some command lines. However, this form of commenting may be used only with commands in which the number of optional parameters is fixed, such as the RESN command or the ID command. Comments are not allowed on the same lines as commands such as the DVT command where the number of parameters may vary. To place comments at the end of a line, precede the comment with extra commas. Commas must be used as place holders for all parameters. If a command is expecting three parameters, then two commas will make the comment the third parameter. (The safest technique is to use five commas preceding a comment, since no commands have more than five parameters.)

Some sample comment formats are shown below:

```
*      This is a comment line using an asterisk
*
*      Two examples of using comments after a command:
REL,%PROGA::-23,,      RELOCATION OF PROGA FROM LU 23
SEARCH,$SYSLB ,,      SEARCH THE SYSTEM LIBRARY
```

Command String Syntax

The commands used in the answer file are structured so that each command starts on a new line. A maximum of 80 characters can be entered in a command line. Each command may have several parameters, which must be separated by commas. The number of parameters may be fixed or variable, depending on the specific command.

Each parameter may also have subparameters, which must be separated by colons. The maximum number of subparameters allowed is six per parameter. Additional subparameters are ignored by the generator.

System Installation

System installation consists of generating a new system file, preparing the target system hardware and media for boot, booting the new system, setting up a primary program, establishing the account structure, spooling, and directories (if the optional HP 92078A Virtual Code+ Package is used), and backing up the new system. Figure 1-2 is a functional diagram of the system installation process.

The installation procedure differs for memory-based and disc-based systems. A memory-based system in this case refers to a system in which program swapping or segmentation is not allowed. A special program named BUILD is used to create a composite system file. The composite system file includes the system file produced in the system generation process and all type 6 program files required for the new operating system. This system can be run without a disc. A special case of a memory-based system is the terminal-less system that is built and then generated across the Distributed System Network. For details on generating a terminal-less system across DS, refer to Volume 1, Appendix A of the DS/1000-IV Network Manager's Manual (part no. 91750-90010).

A disc-based system refers to one that boots from disc and one that allows program swapping and segmentation. All the files required for the new system must be installed on a medium bootable on the new system.

The boot process brings the newly generated (target) system into operation. The new system can be generated from an existing (host) system which can be any RTE-A or RTE-6/VM system.

The backup process is one means of preserving the new operating system in case of emergencies. It eliminates the need for regenerating the system in case the new system is destroyed.

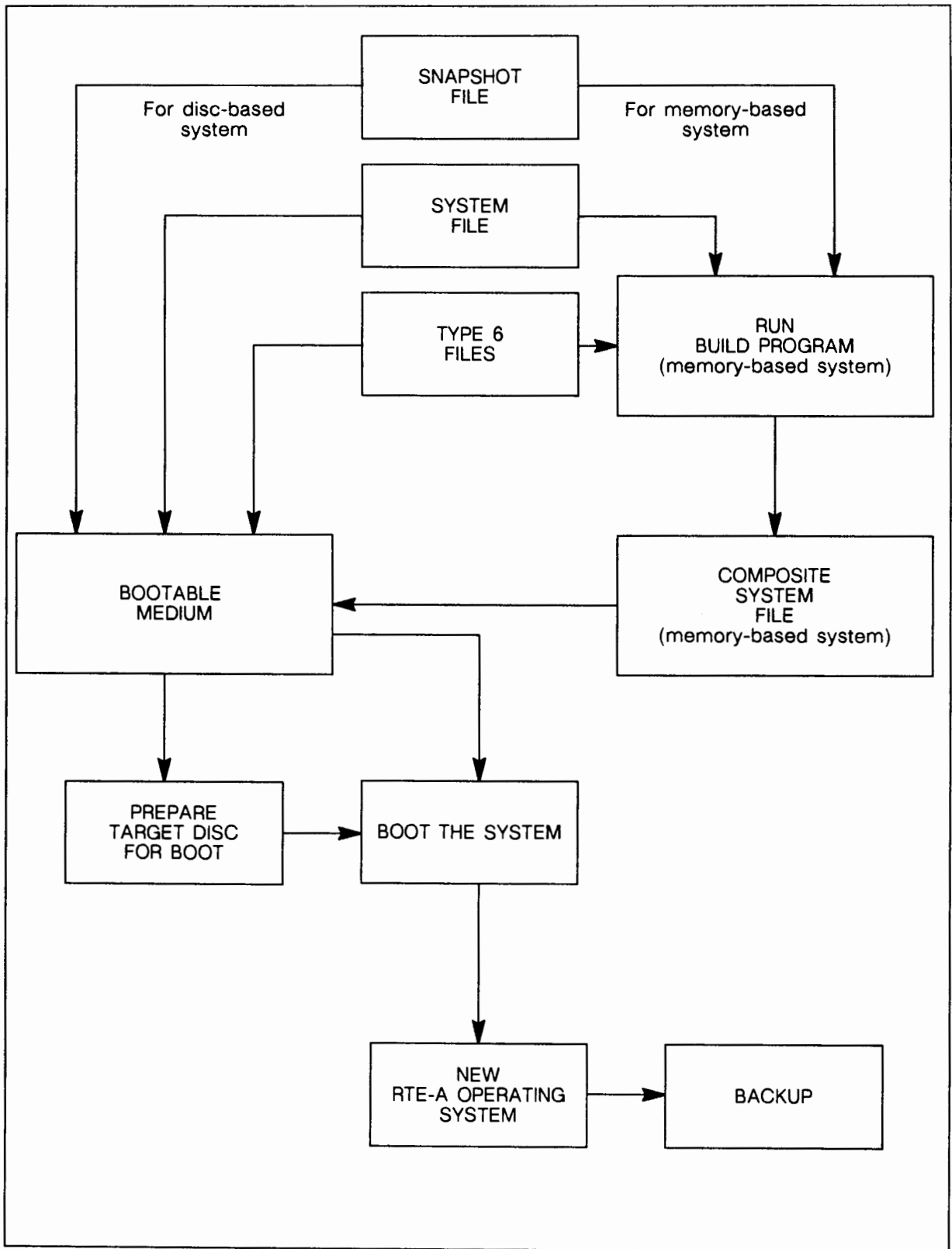


Figure 1-2. RTE-A System Installation Process

Type 6 Program Files

Type 6 program files are produced by the program LINK. These files contain program code in memory-image format, linked for a specific operating system. Such files are ready to be copied directly into a memory partition for execution.

The system file produced by the generator does not contain any programs because the generator does not relocate programs. It is necessary, however, that some programs be available to the system at boot time so that system operations can begin. These programs are described in Chapter 9.

The Boot Process

Booting is the process of loading an operating system into the computer and executing it. You can load a system from a disc, a distributed system (DS/1000) link, a cartridge tape drive (CTD), a magnetic tape, or a PROM module. The system can be booted manually by means of the virtual control panel (VCP) terminal, or automatically at power-on by setting the switches on the processor card of the A-Series computer.

The VCP terminal for manual boot can be a local terminal or a remote terminal connected to the local system through a direct DS/1000 link. A direct DS/1000 link is a distributed system connection with no intervening nodes (computers) on the DS network. Two computer systems so connected are called *neighbors* on the DS network.

Specific boot procedures are given in Chapters 9 and 10. Appendix H contains detailed information on the boot parameters and several boot examples.

Conventions Used in This Manual

Optional parameters are placed in square brackets. For example, in the command

DVT,... [,TO:to] ..., etc.

the “TO:to” parameter is optional. In this case, the TO: portion of the parameter (TimeOut) is required when this option is specified.

Variables that you must supply are shown in lower case letters. For example, the “to” parameter in the above command can be specified as:

DVT,... ,TO:3 ..., etc.

Commands or parameters that must be entered exactly as shown are in ALL CAPITAL LETTERS. For example, in the above example, the optional parameter must be specified as “TO:” followed by the variable “to”.

The slash is used to indicate an OR condition. For example, the following entry

LE[,ON/OFF]

is the same as:

LE[,ON] or LE[,OFF]

The file names specified in the command parameters follow the file system naming conventions. A file can be specified by its name followed by the necessary sub-parameters. Throughout this manual, this is indicated in the command parameters as a file descriptor. A file can sometimes be specified as an I/O device. In this case, the parameter is indicated as a file. Refer to the RTE-A User’s Manual for details of the file descriptor.

The convention HP uses to refer to driver names has been changed from DD.nn to DD*nn. The HP driver relocatable file names of the form %DD.nn have also been changed from %DD.nn to %DD*nn. HP driver names in the NAM statement (DD.nn) and driver entry points (DD.nn) remain the same.

DD*nn	Referenced driver name
%DD*nn	Driver relocatable file name
DD.nn	Driver name in NAM statement
DD.nn	Driver entry point

Running the Generator

General Information

The generator program RTAGN can run on either the RTE-A or RTE-6/VM operating system. This is possible because the RTAGN input and output files are normal RTE files that can be used in either environment. All of these files can be specified in the runstring for RTAGN and all but the command file can be defaulted.

When the generator is run, it reads commands from the specified command file (typically the answer file) and begins constructing a system based on the commands. Information describing each phase of RTAGN and any file errors encountered are displayed at the terminal from which the generator was scheduled. When an error is found during the generation, the appropriate message is written to the list file, the offending command is skipped, and the generation continues. This allows the generation to proceed and scan for any further errors.

The output written to the list file can also be displayed at the terminal by using the RTAGN EC (echo) command. The ER (echo error) command also can be used to display only the error messages. Either command can be given at any time; the output is displayed from that point through completion of the generation process. At the end of the generation, RTAGN displays a message indicating the number of errors found and then terminates.

The RTAGN Runstring

RTAGN is scheduled for execution using the runstring:

```
CI> RTAGN command[,list[,system[,snap]]]
```

where

- | | |
|---------|---|
| command | is the parameter that specifies the name of the file that contains the generator entries and commands needed for system generation. |
| list | is an optional parameter that specifies the name of the list file to be produced by the generator. The default file name is SYLIST. |

- `system` is an optional parameter that specifies the name of the system file to be produced by the generator. This file contains a memory image of the system. The default file name is `SYSTEM`.
- `snap` is an optional parameter that specifies the name of the snapshot file to be produced by the generator. The default file name is `SNAP`.

The RTAGN runstring parameters must be entered in the order shown. If any default is used, the parameter position must be held with commas, as:

```
RTAGN,CMDFIL,,,SNPS1
```

The above runstring specifies `CMDFIL` as the file from which generator commands are to be taken, defaults the list file to `SYLIST` and the system file to `SYSTEM`, and specifies `SNPS1` as the snapshot file. If you omit all parameters, RTAGN will display the runstring syntax showing the parameters in the proper sequence.

If your answer file has a type extension of `.ANS`, any of the list, system, or snap file parameters can be specified as a dash (`-`). The `-` specifies that the name of the answer file is to be used with a type extension of `.LST`, `.SYS`, or `.SNP` for the list, system, or snap file, respectively. These files will overlay any existing files with the same name. For example:

```
RU,RTAGN,SYSTEM1.ANS,-,-,-
```

will produce the files `SYSTEM1.LST`, `SYSTEM1.SYS`, and `SYSTEM1.SNP`.

Command File

A command file is a source file constructed by the user that contains commands needed by the generator to generate the operating system. This file must be specified in the runstring. If a command file is not specified, the generator aborts after displaying the following message:

```
RU RTAGN COMMAND LIST SYSTEM SNAP
```

List File

The first line of the list file output by the generator contains the runstring used to schedule RTAGN. The next line contains the system time at the beginning of the generation. Following this header information, the file contains a listing of the input commands and comments, the program names, bounds, and entry points, as well as any errors encountered during the generation.

If the list file is not specified, the default file `SYLIST` will be created by RTAGN. If `SYLIST` already exists, the generator will not overlay it and will abort with the "File already exists" error message.

If a file name starting with an apostrophe (') is specified for the list file, or if "-" is specified, the generator will overlay that file if it already exists.

Any error messages output by RTAGN are written to the list file, followed with the line

```
** error **
```

In this way, error messages can be easily identified.

System File

The system file is a type 1 file that contains the memory image of the operating system. This file is copied into memory and executed at system boot. If not specified, the default file SYSTEM will be created by RTAGN. If it already exists, it will not be overlaid. The generator will abort with the "File already exists" error message.

If a file name starting with an apostrophe (') is used, or if a "-" is specified, the generator will overlay the file if it exists. If, however, the existing file was specified with a security code, it will not be overlaid unless the new file name is specified with the same security code.

Snapshot File

The snapshot file is a type 3 file created by RTAGN to hold the generator system snapshot. It contains the values of system entry points, system library names, and other system information such as system checksums and system common checksum. If an entry point or module name is longer than sixteen characters, the additional characters are ignored by the generator.

The snapshot file is required by the loader to load programs on-line. If the default snapshot file name SNAP is specified and this file already exists, it will not be overlaid. The generator will abort with the "File already exists" error message.

If the snap file name is specified starting with an apostrophe (') or a caret (^), or if "-" is specified, the generator will overlay the file if it exists. If, however, the existing file was specified with a security code, it will not be overlaid unless the new file name is specified with the same security code.

Overlaying Files

If the first character in a list, system, or snapshot file name specified in the generator runstring is an apostrophe ('), or if the file is defaulted with a dash (-), an existing file of the same name will be overlaid each time RTAGN is executed with that file specified. An existing snapshot file with the same name will also be overlaid if the first character is a caret (^). If, however, the file was originally specified with a security code, it will not be overlaid unless the new file specification includes the same security code.

If you specify a file name without the leading ' or ^ character, and the file already exists, it will not be overlaid by the generator. In this case, the generator will terminate immediately, displaying a duplicate file name error.

For example, if the generator runstring is:

```
CI> rtagn,comnd,=list,'outpt,snap
```

the file 'OUTPT will be created by the generator if it does not exist. If it does exist prior to running the generator, the old information will be overlaid by the new output from the generator. The file =LIST will be created if it does not exist; if a file by that name does exist, it will not be overlaid and the generator will abort with the message:

```
File already exists =LIST:::3:155
```

The file SNAP will also be created by the generator if it does not exist. If it does exist prior to running the generator, the generator will not overlay it and will abort with the file already exists message.

In the above example, the file 'OUTPT would not be overlaid, despite the leading ' character, if the original file was specified with a security code. For example:

Existing File	New File Specified	Existing File Overlaid?
'OUTPT:BC:JT:4:155	'OUTPT	No
'OUTPT:BC:JT:4:155	'OUTPT:BC	Yes
'OUTPT::JT:4:155	'OUTPT	Yes
'OUTPT::JT:4:155	'OUTPT:BC	Yes

Initialization Phase

In this phase, the generator initializes internal tables and variables in preparation for system generation. This phase is very short. You may specify the linking method to be used by the generator and the amount of memory in the system. Use the worksheet in Appendix B to record your choices.

You can specify either current page linking (CPAGE or CP) or base page linking (BPAGE or BP). These types of linking are discussed in detail in the System Design manual. This command covers all links generated up to and including the System Message Relocation Phase. If you specify BPAGE, the links will be placed in the base page linkage area. If you specify CPAGE, the generator will place as many links as possible in the current page linkage areas, and the rest will be placed on the base page.

The linkage area is used for system links. If current page linking is specified, most required links will be placed on the page of memory currently being relocated. Base page will be used when it is not possible to put the link on the current page. Thus, while current page linking does use base page, it uses much less than in base page linking.

If base page linking is used for a given module, a few words of memory on the current page are saved, at the expense of words on base page. This leaves less room for links from other modules. Guidelines for conserving base page links and program space are given in the following section.

Conserving Space

For most applications, you can relocate all of the modules shown on the worksheet that make sense for your hardware configuration, as this will give you a full-capability system. If, however, your application does not require the full system capability, and if you need as much memory for table space as possible, you can omit the modules not required by crossing them out in the worksheet. Refer to the System Design Manual for the names of modules you can eliminate. The essential system modules are:

- %ABORT
- %EXEC
- %IORQ
- %MAPS
- %PROGS
- %IOMOD
- %UTIL
- %RTIOA
- %VCTR

and one driver;

The remaining modules are optional.

If you specify base page linking (BP) in the Initialization Phase, you can reduce the size of your operating system, and thereby increase the space left for programs. Because each link takes one word of memory, you will reduce the system size by one word for each current page link you can eliminate.

The most effective way to conserve base page links is to use current page linking. Also, you may reduce the number of base page links by using the LOCC command to reorder the system modules. Where possible, this causes each large module to be contained entirely within a page, and page boundaries are crossed by small modules. Each page of memory contains 1024 words (2000 octal).

Reordering system modules in this manner will waste a little space at the end of the page, but it will minimize the number of page crossings (and therefore the number of links). With a minimum of effort, you can arrange the modules in an order that will waste little memory and still save a significant number of base page links.

Reordering of system modules will not have any effect on system size if you are using base page linking. However, it may have an effect on the number of base page links generated. The size of the base page remains constant no matter how many base page links are generated.

Links

The linking option is specified in the initialization phase for the relocation of the entire operating system. This is done using the LINK command in the generation answer file:

```
LINK,[CP/BP]    (CP is the default)
```

CP specifies current page linking

BP specifies base page linking

With current page linking, the links required for relocating particular modules are put on the page currently being relocated whenever possible. This saves linkage space on the base page.

With base page linking, any links generated during the relocation processes are put on the first page of physical memory (base page). This option tends to conserve program space, but uses base page heavily.

The generator displays information about the base page links used for each module. For example,

```
RE,%A
A          <start_cp> <stop_cp>
BP LINKS  <start_bp> <stop_bp>
USED      BP   nn    CP   mm
```

This display shows that module A used nn base page links and mm current page memory locations. The <start_cp> and <stop_cp> numbers reflect the module relocation boundaries (not including current page links) and the <start_bp> and <stop_bp> numbers reflect the location of base page links used by the module (does not include base page relocation). Note that base page usage grows towards low memory whereas current page usage grows towards high memory.

If the LINK command is specified incorrectly, the generator defaults to current page linking and the message:

```
Link error, current page used
```

is written to the list file.



System Relocation Phase

During the system relocation phase, the operating system modules and the drivers needed in the system are relocated. Any libraries that must be searched to satisfy undefined external references generated during the relocation process also are specified in this phase.

Use the Initialization and System Relocation Worksheet in Appendix B to specify the modules, libraries and drivers for your system. Cross out those modules not needed, and enter the FMGR cartridge reference designator or CI directory where the file resides during relocation for those modules and libraries you are generating into your system.

Relocating Modules

Except for %VCTR, which must be the first module relocated, system modules and drivers not targeted for special OS module/driver partitions may be relocated in any order; you may even intermix them if that is useful. The system and driver modules are described in the following sections.

System Modules

System modules are relocated as shown in the System Relocation worksheet. For most applications you can relocate all of the modules shown on the worksheet, as this will give you a full-capability system. If your application does not require the full system capability, and if you need more memory for table space, omit the modules not required. Refer to the System Design Manual to determine which modules you can eliminate. The minimum system modules required are %ABORT, %EXEC, %IOMOD, %IORQ, %MAPS, %PROGS, %RTIOA, %UTIL, %VCTR, the proper %RPLxx file, and one driver. The remaining modules are optional.

Be careful not to omit any modules that are referenced by the modules that remain in your system. The function of each module and its dependencies on other modules are discussed in the System Design Manual.

Driver Modules

If your system has drivers that will be loaded into driver partitions, refer to the OS Module/Driver Partition Phase chapter of this manual.

Privileged and unmappable drivers cannot be partitioned and must therefore be loaded as described here.

Driver modules are relocated in the same way as system modules. They may be relocated in any order, and may even be intermixed with the system modules. You must relocate all drivers that are required by the interfaces and devices that are generated into your system. Appendix E lists the HP-supplied devices and interfaces that may be used in a system and the standard (default) generation answer file entries.

Only one copy of each driver is required, even if it will be used by more than one device.

If your system has the power-fail recovery option, you must relocate the power-fail driver, ID*43, in your system. When a power failure occurs, this driver notifies all other drivers and causes the background program AUTOR (auto restart) to be scheduled when power returns.

Libraries

After the system modules and drivers have been relocated, the \$SYSA and \$SYSLB libraries should be searched to satisfy any undefined external references made during the relocations. Library \$SYSA contains dummy modules that are used when certain system modules are omitted from the system. Refer to the System Design Manual for more information.

At this time you may also search user libraries (to satisfy references from any user-written driver, for example). Note that the generator will attempt to satisfy undefined external references with the first entry point of the required name that it can find; i.e., the first library file that contains that entry point. The merged library \$BIGLB cannot be used because of possible EMA errors due to allocate records, however modules within \$BIGLB may be searched if required.

System Relocation Phase Commands

This section describes the commands used to select the modules and to guide the relocation of modules included as part of the operating system. The modules relocated are the operating system modules, the I/O drivers, and labeled common.

These commands are described in the following paragraphs. A summary of the commands is listed below in the proper order of entry:

```
RELOCATE,file[,module name]
SEARCH,file[,module name]
(Any optional commands may be entered here)
END
```

Optional commands:

```
MSEARCH,file
LOCC[,address]
BLOCC[,address]
DISPLAY
LENTRIES[on/off]
ALIGN
TG,tags
PARTITION[,module[,module[,...]]]
```

Note that the generator commands may be abbreviated to two characters.

RELOCATE Command

Purpose: Relocates a file to be included as part of the operating system.

Syntax: RE,file[,module name]

file may be a system module, driver, or subroutine.

module name specifies that only the module named is to be relocated.

CAUTION

Partitionable system modules must not be specified in either an SE or an MS command. Refer to the section System Partitioning Concerns for details.

SEARCH Command

Purpose: Searches the specified file to satisfy undefined externals.

Syntax: SE,file[,module name]

file specifies the library file to be searched. The file is searched only once, thus backward references within the library will not be satisfied.

module name specifies that only the named module is to be examined for external reference resolution.

MSEARCH Command

Purpose: Searches the specified file multiple times to satisfy undefined externals. All backward references within the library file will be satisfied.

Syntax: MS,file

file specifies the name of a file to be searched.

LOCC Command

Purpose: Displays or sets the current module relocation base address. The relocation base address is the memory location where the module to be relocated will start in physical memory.

Syntax: LO[,addressB]

addressB sets the relocation base address of the module to be relocated. Address is specified in octal.

If omitted, the current relocation base address is displayed in the list file and the value of the LOCC pointer remains unchanged.

Description:

The address entered should be greater than the current relocation address. If not, the generator issues a warning message to the list file but continues with the generation. The message warns of the possibility of overlaying portions of the previously relocated system.

The LOCC command may be entered to increase the value of the relocation address at any time during generation. The increase may conserve base page links. It also has the effect of leaving space between the two modules in memory. For example, in Figure 4-1 file %EXEC was relocated, memory was filled to address 3770B, and the location counter was set to 4000B with the LOCC,4000B command. Then the %RTIOA file was relocated. This resulted in RTIOA starting on a new page boundary while leaving the space between the two modules unused.

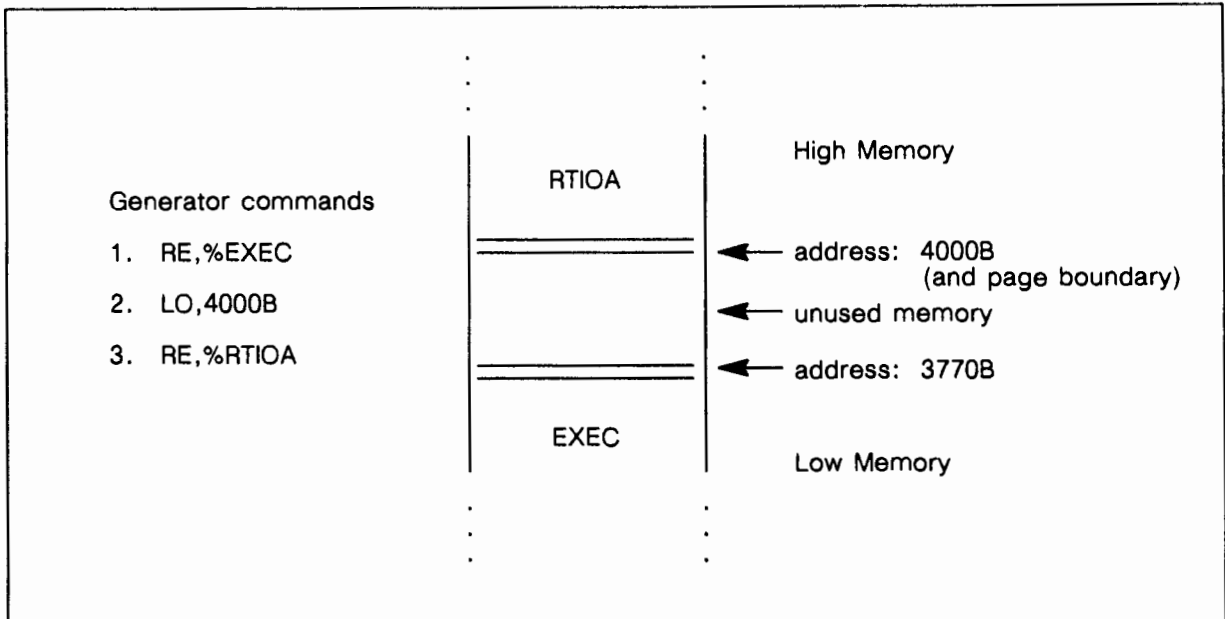


Figure 4-1. Results of Using the LOCC Command

BLOCC Command

Purpose: Sets or displays the current base page relocation address.

Syntax: BL[,addressB]

addressB the octal value to which the base page location counter is set.

Description:

The address is first checked to determine if it is within the base page bounds. If the address is not within the base page bounds, an error message is given and the base page relocation counter does not change. The value may be increased or decreased at any time during the relocation process. Care should be taken not to overlay base page links already used.

DISPLAY Command

Purpose: Issues a list of the current undefined externals to the list file.

Syntax: DI

LENTRIES Command

Purpose: The List Entries command turns on or off listing of module entry points to the list file.

Syntax: LE[,ON/OFF]

ON/OFF The default is ON, list the module entry points.

Description:

If the OFF option is specified and later in the generation the ON option is selected, all entry points from previously relocated modules will be listed.

ALIGN Command

Purpose: Causes the relocation address to be aligned to the next page boundary.

Syntax: AL

Description:

The ALIGN command causes the relocation of the next module to start on a page boundary. If the relocation address is already at the page boundary, the ALIGN command has no effect.

END Command

Purpose: Terminates system relocation phase.

Syntax: EN

Description:

This command signals the end of relocation and enables the generator to proceed to the next phase of generation.

System Partitioning Concerns

Two commands, TG and PA, must appear in the system relocation phase if operating system modules are relocated into partitions during the partition relocation phase. (This phase is described in Chapter 5.)

In addition, the RTE-A system mapping module MAPOS must be included in any generation in which system modules are relocated into partitions. Add "RE,%MAPOS" to the answer file.

TG Command

Purpose: Specify the size needed for tags. Tags are interface routines created by the generator for use by RTE-A to enter and exit partitioned modules.

Syntax: TG,n
 n tag area size in number of words

Description:

The TG command **MUST** be given before any partitionable system module is relocated. Only one TG command may be given. Unless modules are being reordered to conserve links, you should give the TG command immediately after %VCTR is relocated.

The number of tag area words required for each partitionable system module is listed below:

# Words of Tag Area (In partition)	# Words of Tag Area (In main system)	Module	Description
6	0	ABORT	program abort processing
13	2	ALARM	timer signals
29	2	CDSFH	load code segment from disc (VC+)
34	2	CHECK	program security checks
49	2	CLASS	for class I/O
8	2	DSQ	required for DS/1000
30	0	IORQ	I/O requests
84	2	LOAD	EXEC 8, load and swap programs
80	2	LOCK	resource numbers, LU locks
77	2	MEMRY	EXEC 28, manage memory partitions
18	2	PERR	memory parity error recovery
18	2	SECOS	os security checks
53	2	SIGNL	signal processing
44	2	SPOOL	spool processing
364	2	SPSLG	Performance monitoring
14	2	STAT	EXEC 26, PS and DS commands
85	2	TIME	time scheduling, time requests
34	2	VEMA	VMAIO requests
46	2	XCMND	AS,BR,CD,DN,DT,GO,PR,SS,SZ,LU, VS,WS commands

To determine the tag area size for your generation, add the number of tag area words for each system module placed in a partition, adding two this value for each module in the above list (except IORQ or ABORT) that you use in your generation but you do not include in a partition. Add one to the final value and use this value in the TG command. For example, if you place all of the partitionable system modules above in a partition specify the tag command as

```
TG,1086
```

RTAGN allocates the specified number of words for the tag area. The bounds of the tag area are printed:

```
TG,500
```

```
Tag area 2444 3427
```

If you do not allocate enough tag area, the message

```
Tag area overflow
**      error      **
```

will occur. It may occur many times. If you allocate a larger tag area than is required, the extra space is wasted. When the partitioning phase is complete, RTAGN will tell you how many words remain to the tag area and how the tag area was used:

```
*          34 unused words in tag area
*          56 jsb tags created
*          42 jmp tags created
*          3 trap cell tags created
```

If there are a lot of wasted words, you may want to reduce the tag area size and re-generate the system.

Note that even if no tag command is specified, a small tag area is created automatically by the generator. This area is reserved for partitionable module flags.

Partitioned Command

Purpose: Specify the system modules that will be relocated into partitions.

Syntax: PA,<module name>[,<module name>][,.....

Example: PA,STAT,CDSFH,VEMA

Description:

The PA command informs the generator of the modules that will be relocated during the partition phase. It must appear before \$SYSA is searched. Otherwise, the dummy versions of the partitioned modules would be relocated out of \$SYSA. (See the System Design Manual for information on optional and dummy system modules.)

A PA command cannot extend from one line to the next or be continued in any way. However, multiple PA commands may appear and all the modules specified will be expected in the partitioning phase. If a module name entered is not actually an RTE-A system module name, no error will result. The erroneous name is ignored.

For fastest generation time, the PA command should appear JUST before the "SE \$SYSA" command. For improved readability, all PA commands should appear together.

OS Module/Driver Partition Phase



General Description

The RTE-A system is limited to 32 pages of logical memory. Since a fully loaded system (one that includes all operating system modules, all drivers, and other software tools) would exceed this limit, OS Module/driver partitioning was designed to allow mapping of software modules into the system logical map. With partitioning, only the module required for a specific system operation in progress is mapped into the system. Other modules are ignored so that memory space will be available for other system uses.

In the following discussion, the term “module” is used to represent the more cumbersome “OS module/driver”. Module partitions are defined at generation, after the nonpartitionable system modules have been relocated. Because partitioned modules are mapped dynamically as they are needed in operation, the order in which they are generated into the system is not important. The size of the partition that becomes part of the logical system map is the size of the largest partition created; the maximum is eight pages. Up to 127 partitions may be specified.

More than one module may be relocated into a partition. The END command that separates partitions moves the relocation address to the next page boundary in the generation; thus, if a small module will fit in the remaining space between another module and the next page boundary, it may save user program space in upper memory by relocating the small module in the same partition.

In most cases, however, it is recommended that you relocate a limited number of modules per partition. This will keep the size of the partition in the logical system map at a minimum. The approximate size of drivers used in the RTE-A system is given in Table 5-1. Table 5-2 gives the approximate size of system modules. Note that the sizes given (in decimal words) do not include links allocated by the generator, and that the size of each driver is subject to change on any software revision. The information in Table 5-2 also contains the decimal number of tag area words required for each of the partitionable system modules; further information on the tag area is found in the System Relocation Phase chapter.

Table 5-1. Partitionable Driver List

Driver	Description	Size
ADV00	DS/1000 I/O Mapping	575
DD*00	Terminal	1056
DD*12	Line Printer Device (HP-IB)	702
DD*20	CTU Device	697
DD*23	Magnetic Tape Device	590
DD*24	Magnetic Tape Device (Streaming Mode)	1095
DD*30	Disc Device (ICD)	497
DD*33	Disc Device (CS/80)	1879
DD*60	X.25	720
DD*63	MRJE	115
DDC00	Full Function Serial Driver	977
DDC01	Full Function Serial Driver for Terminals with Attached Devices	1751
DDC12	2608S Line Printer Device (HP-IB)	1563
DDM30	Disc Device (MAC)	1023
DDX00	X.25	2225

Table 5-1. Partitionable Driver List, continued

Driver	Description	Size
ID*00	Async Serial	473
ID*01	Async Serial with Modem	1003
ID*27	Integrated Disc Driver	894
ID*36	PROM I/F	136
ID*37	HP-IB I/F	1476
ID*41	Writable Control Store (WCS, A700)	190
ID*50	Parallel I/F (GPIO)	487
ID*52	Parallel I/F	445
ID*66	HDLC NS/1000 & DS/1000-IV	1335
ID*67	IEEE 802.3 (LAN)	2000
ID100	ASIC Interface Driver for DDC00/DDC01	1044
ID101	ASIC I/F Driver for DDC00/DDC01 with Modems	1167
ID400	OBIO Interface Driver for DDC00/DDC01 (A400 only) with support for External Modems	1451
ID800	8-Channel MUX Interface Driver for DDC00/DDC01	999
ID801	8-Ch MUX I/F Driver for DDC00/DDC01 with support for External Modems	1096
IDM00	Async Mux	1381
IDM37	HP-IB Driver for MAC Discs	1322
IDS00	Programmable Serial Interface (PSI)	1261
IDS64	DS/DL (Data Link) Slave	1370
IDZ00	TELNET Pseudo Terminal Driver	871

Table 5-2. Partitionable System Module Sizes and Tag Area Requirements

Module	Description	Size	Tag Area
Abort	Abort Processing	519	6
Alarm	(VC+ only) Timer Signals	395	13
Cdsfh	(VC+ only) Load Code Segments From Disc	795	26
Check	(VC+ only) Security Checks for OS Calls	138	34
Class	Class I/O	1062	49
Dsq	Required for DS/1000	403	8
IORQ	I/O request initiation	1122	30
Load	EXEC 8, Load and Swap Programs	1278	84
Lock	Resource numbers, LU locking	748	80
Memry	EXEC 28, Manage Memory Partitions	1963	77
Perr	Memory Parity Error Recovery Routine	525	18
Secos	(VC+ only) Security Checks for USER calls	245	18
Signl	Signal Processing	1245	53
Spool	Spool Processing	390	44
Spslg	Performance Monitor	2001	364
Stat	EXEC 26, PS and DS Commands	515	14
TIME	Time requests	1025	85
Vema	VMAIO Requests	390	34
Xcmnd	AS, BR, CD, DN, DT, GO, PR, SS, SZ, LU, VS, and WS Commands	830	46

After completing the system relocation phase, the generator displays the following:

‘OS/Driver partition relocation’

OS/driver partition 1

At this time, all system modules and drivers to be included in partition 1 can be relocated. The END command moves RTAGN to the next partition. After terminating the last partition relocation, a second END command terminates the module partition phase.

Accessing a module or driver in a partition requires some overhead. For drivers, the overhead is encountered when mapping on an EXEC call or interrupt entry. This applies only to drivers in the partitions; drivers relocated in the nonpartitioned portion of the system are accessed without the added time. Some user drivers must be modified so that DMA access are not from the driver itself, and to have the correct generation records. Refer to the RTE-A Driver Design Manual for details.

In addition to the page-mapping overhead incurred by accessing partitioned drivers, accessing partitioned OS modules also incurs overhead in passing parameters. The number of parameters passed varies with the OS routines used. In general, the less frequently the routines are used within a module, the better candidate that module is for partitioning. For example, the STAT module contains routines that process EXEC 26 calls and the PS and DS commands. If very few of these calls are made, placing STAT in a partition will have little impact on system performance. However, in a system in which many CDS programs execute (VC+ only), placing the CDSFH module into a partition will cause extra overhead for code segment faults.

Sample logical and physical memory maps are shown in Figure 5-1.

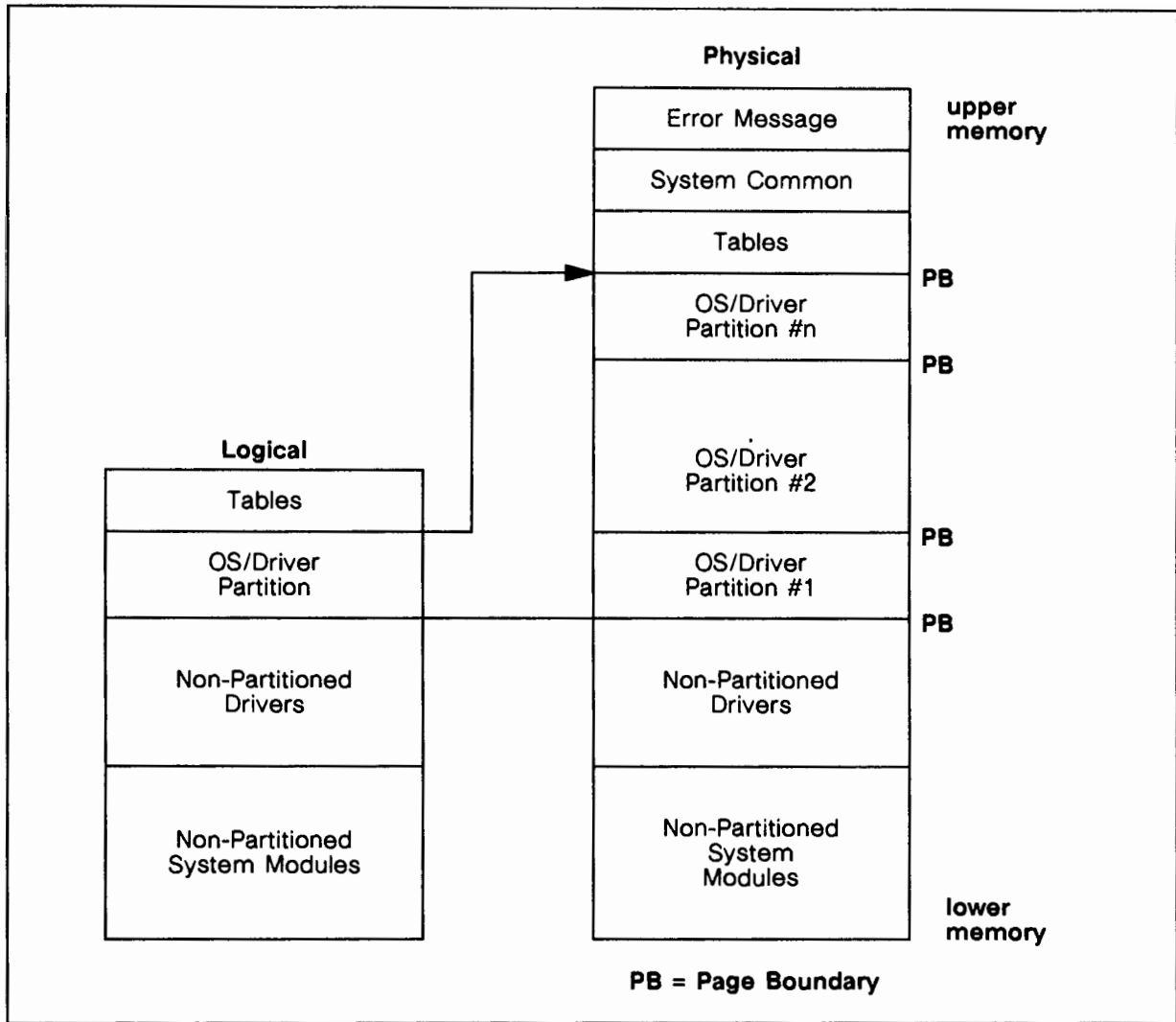


Figure 5-1. Sample Memory Maps

Non-partitioned drivers are loaded with the non-partitioned system modules. This must be done for privileged drivers and drivers that do not contain a GEN PA record in their relocatables (partitionable drivers must contain a GEN PA record). Refer to Appendix D for an explanation of GEN records.

All commands used in the system relocation phase (e.g., RE, MS, LOCC) are valid in the partition phase; however, system libraries (\$SYSLB, \$SYSA, etc.) should be searched prior to the partition phase. Note that if you partition the TIME module in an A600 system, you will need to relocate the .DMP and .DDI modules from the \$MATH library in the system relocation phase, as these modules are not partitionable. The ALIGN command, described in the following section, also can be used in this phase to align the page boundaries of the modules in a partition.

ALIGN Command

Purpose: Aligns the page boundaries of modules/drivers in a partition.

Syntax: AL or
ALIGN

Description:

The ALIGN command can be used for partition relocation, system relocation, system common relocation, and system message relocation.

This command aligns the current relocation base address to the next page if it is not already at the beginning of a page.

Aligning page boundaries reduces the number of base page links required. Since base page is not mapped, all links must fit. Because base pages links can quickly become unavailable, it is advisable to align modules and drivers on page boundaries within a partition.

Partition Configurations

Two sample configurations are shown in Figure 5-2. The configuration on the left (A), is better suited for current page linking, even though it fragments memory. The configuration on the right (B) would create many base page links.

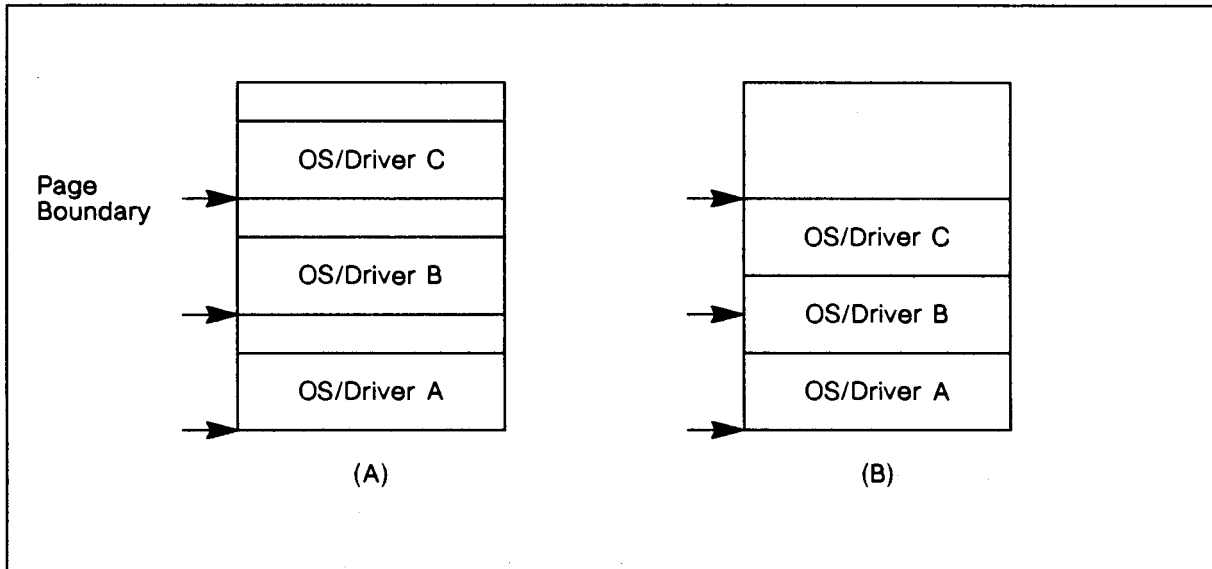


Figure 5-2. Sample Partitions

To align modules on page boundaries, the ALIGN command can be used between RE commands. As an example, the following sequence will map a driver partition and an OS module partition as shown in configuration (A) of Figure 5-2.

```
*
*Driver Partitioning
*
RE,%DD.45
ALIGN
RE,%DD.46
ALIGN
RE,%DD.47
END
*
*OS Module Partitioning
*
RE,%stat
ALIGN
REL,%dsq
.
.
.
END
END
```

Table Generation Phase

This chapter provides the information needed for planning and preparing the entries or the table generation phase of the system generation answer file. The information includes the following:

1. General descriptions of the various processes in the table generation phase.
2. Explanation of the I/O configuration worksheet.
3. Explanation of the IFT/DVT Worksheet.
4. Explanation of the use of defaults in a system generation. These defaults include default values for disc configuration.
5. Explanation of the use of node lists.
6. How to complete the table generation portion of the answer file.
7. Descriptions of the commands used in the table generation phase.

General Description

The table generation phase consists of several processes. First the interface and device tables are generated from the information specified by the interface and device table (IFT and DVT) commands. Next, the node lists are specified. Each node list contains the logical unit (LU) numbers on one device or controller. Then the interrupt tables, if required, are generated.

Interface and Device Tables

The interface and device tables are memory-resident tables that identify the interface cards and the devices to the system. The interface table (IFT) contains such information as the interface driver, the select code, and the interface type for a particular interface card. The IFT handles all the interface I/O requests made by a program to a device. The size of the IFT is nine words, plus any extensions.

The device table (DVT) contains device specific information: associated driver, device status, device time-out and buffer limits, and other device parameters. The size of the DVT is 25 words, plus any extensions.

Node Lists

A node list is a list of logical units that use the same I/O controller. In most cases, this means multiple logical units that are contained in the same physical device (such as CS/80 disc with several LUs and its CTD LU). In devices such as a line printer, digital voltmeter, or scanner, the controller is associated with only one device identified with a single logical unit. Typically, each device has a hardware controller that accepts and interprets commands and data transmitted to it.

In RTE-A, physical devices with multiple logical units using the same controller must be included in one node list. Inclusion in a node list ensures that these logical units are accessed by the computer one at a time. For example, if one LU is being serviced for a programmatic request (such as a read or write request), another request must wait until the first request is completed. The LU numbers in a particular list are serviced one at a time in a circular fashion, and each request from or to a new LU is completed in order of entry. If all LUs for a device are not in a node list, access will fail when two independent programs try to address different logical units on the same controller at the same time.

Figure 6-1 shows typical relationships between interface cards, devices, and LUs. Each box that has a bold outline represents a device. Note that a device may be made up of more than one LU. Examples of this are:

1. Disc drives with multiple LUs and/or integral tape drives.
2. Terminals with peripherals (for example, integral printers and cartridge tape units).

Also note that one interface card may have more than one device connected to it. The two most common interfaces like this are the HP-IB and the MUX interfaces.

Two general rules concerning node lists:

1. Devices that have multiple LUs must have all of the LUs that make up that device put into the same node list.
2. Interface cards that can have multiple devices should not have the LUs of separate devices put into a common node list.

For Figure 6-1, the following node commands should be put in the answer file:

```
NODE,21,22  
NODE,26,27
```

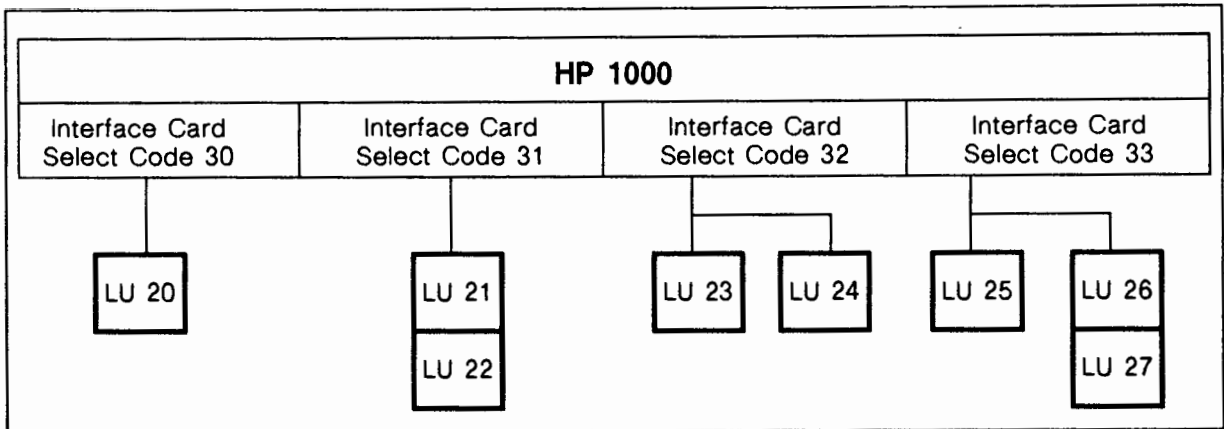


Figure 6-1. Example Node List Configuration

The IFT/DVT worksheet has space for node lists. For each case in which there is more than one LU for a device list all LU numbers associated with the device. The generation node list entry can be placed on multiple lines, using a dash following the comma separator to indicate the continuation. For example:

```
node,12,13,14,-
17,18,19,-
20,21,23
node, 1,3,5
```

Interrupt Table

The interrupt table contains information that specifies what the system is to do when an interrupt comes from any select code. It allows special handling of all interrupts from specified devices.

Typically, the interrupt table entry for a particular select code would reference the IFT for that select code. This link is generated automatically by RTAGN. This means that only in the special case when the interface driver is not to be called to service an interrupt, must an entry be specified for this table in the generator command file.

The System I/O Configuration Worksheet

A blank I/O configuration worksheet is given in Appendix B of this manual. Proper completion of the worksheet will insure that your LU numbers, select codes, and Hewlett-Packard Interface Bus (HP-IB) addresses have been assigned correctly. A sample I/O configuration worksheet is shown following this section.

In the first column, list all of the peripheral devices that are to be connected to your system. List the HP-IB itself, as well as the devices on the bus. In addition, make one entry for each disc logical unit that you will include in your system, even though several disc LUs may be located on the same physical disc drive.

On each line in the second column, enter the interface card that will be used by the device in the first column.

In the third column, enter the logical unit number of each device. The LU number is the reference number to the device. LU numbers may be in the range of 1 to 255 (decimal) and each device must have a unique LU number. This includes the HP-IB itself, as well as the devices on the bus. Even if you have no need to access the bus interface card directly, some system utilities may try to access the card, and will fail if an LU number has not been assigned.

In the fourth column, enter the select code for each interface card. The select code is the address of the interface card in the same way that the LU number is the address of a device. Select codes must be in the range of 20B to 77B. Each interface card must have a unique select code, which must match the value set by switches on the card.

If you have an interface that uses a privileged driver, there are special limitations on the select code values that may be used. Refer to the Sample System Generation Answer File appendix for the standard select code assignments.

The last column is reserved for the addresses of devices that are connected to an HP-IB. Each device connected to an HP-IB must have an address that is unique to that HP-IB. Valid addresses are in the range of 0 to 36B, with 36B reserved for the HP-IB interface itself.

In the sample worksheet shown on the next page, there are two HP-IB cards with select codes of 27B and 25B respectively. Note that each HP-IB card has its own LU number. In both cases, HP-IB address 36B sends control requests to the interface card. The remainder of the HP-IB addresses serve to access individual devices connected to the bus.

Referring to the Worksheet example, LUs 16 through 20 and 24 all have the same HP-IB address (0) because they are all on the same device. LUs 25 through 28, on the other hand, refer to separate devices; therefore, each has its own LU number.

The completed system I/O configuration worksheet gives a quick summary of the I/O structure of your system. It will be the basis for filling out the IFT/DVT worksheet.

The IFT/DVT Worksheet

The IFT/DVT worksheet is given in Appendix B of this manual. One worksheet must be prepared for each interface card in your system. (If you have more than five logical units attached to one interface, use additional worksheets for continuations.)

The information needed to fill out the IFT/DVT worksheet is found in the I/O configuration worksheet you have just completed, and in Appendix E, Device and Interface Driver Tables. Table E-2 summarizes the standard entries for the generation answer file. The sample IFT/DVT worksheet shown at the end of this section has been completed for the HP-IB#1 interface and the 7911 disc drives configured on the sample I/O configuration worksheet. The origin of the required information for each entry is defined below.

INTERFACE NAME	- I/O Configuration Worksheet
I/O Slot #	- Physical card cage slot number
IFT	- Table E-1, Device/Driver Correspondence: Interface Cards Category, 12009 HP-IB Interface entry
SC	- I/O Configuration Worksheet This code also is physically marked on the interface card. If this is a pseudo driver, the select code will be 0.
E	- Table E-22, IFT Default File Values: 12009A Column
QU	- Table E-22
TX	- Table E-22: Table Extension entry blank, use default value
IT	- Table E-22: DVT/IFT Device Type entry
AL	- LOkked. System disc and system console should have AL:LO; all others should leave this blank.
PS	- Blank: Interface has an assigned select code, therefore it is not a pseudo driver. (A pseudo driver may not have a select code.)
Device Name	- I/O Configuration Worksheet
Device Driver	- Table E-1: CS/80 Devices Category, 7911 entry

- Default File - This entry is generally the same as the device driver, or is a user-prepared file
- Model Number - Table E-20, 7907/08/11/12/14/33/41/42/45/46 Default File Values:
Determined by disc subdivision (Each model number must be entered in full)
- Logical Unit - I/O Configuration Worksheet
- Remaining DP Entries - Table E-20:
Determined by disc subdivision
- Queuing: - FIFO:
(First In, First Out). This is the default.
- NODE: - Disc LU numbers
(Logical Unit entered above)

```

*****
*
*           IFT/DVT Worksheet
*
*****

```

INTERFACE NAME : HP-IB#1
I/O SLOT #: 2

IFT, %ID*37 ,SC: 27B ,E ID.37 ,QU: PR ,TX: ,

Interface Driver Namr (Default file)	Select Code	Entry Point	Queuing	Table Extension
--	----------------	-------------	---------	--------------------

IT: 37B ,AL: LO ,PS:

Interface Type	Port Map Allocation	Pseudo Driver
-------------------	------------------------	------------------

(IFT/DVT Worksheet, continued)

Device Name: Device Driver:	7911 DD*33				
Defaults File: Model Number: Logical Unit:	DD*33 M7911_CF:0 LU: 16	M...:1 LU: 17	M...:2 LU: 18	M...:3 LU: 19	M...:4 LU: 20
Device Type: Device Priority: Time Out: Buffer Limits: Table Extension: Driver Extent: Driver Prams:	DT:33B PR:63 TO:1000 BL: : : TX: 72 DX: 8 -----	DT:33B PR:63 TO:1000 BL: : : TX: 72 DX: 8 -----	DT:33B PR:63 TO:1000 BL: : : TX: 72 DX: 8 -----	DT:33B PR:63 TO:1000 BL: : : TX: 72 DX: 8 -----	DT:33B PR:63 TO:1000 BL: : : TX: 72 DX: 8 -----
start # 1 2 3 4 5	DP :1 : 0 : 0 : 0 : 0 : 0	DP :1 : 0 : 0 : 0 : 0 :19200	DP :1 : 0 : 0 : 0 : 0 :64750	DP :1 : 0 : 0 : 0 : 0 :14592	DP :1 : 0 : 0 : 0 : 0 :27392
start # 6 7 8 9 10	DP :6 : 400 : 48 : 0 : :	DP :6 : 943 : 48 : 0 : :	DP :6 : 320 : 48 : 0 : :	DP :6 : 266 : 48 : 0 : :	DP :6 : 352 : 48 : 0 : :
start # 11 12 13 14 15 Queuing: Node 1:16,17,18 Node 2:19,20 Node 3: : :	DP :11 : : : : QU: FI	DP :11 : : : : QU: FI	DP :11 : : : : QU: FI	DP :11 : : : : QU: FI	DP :11 : : : : QU: FI

Making Use of Defaults

For most of the parameters that pertain to each device or interface, the values can be obtained from a default file instead of entering the parameters in the answer file. The default values are contained in Appendix E. You can construct your own default file (refer to Appendix D for instructions) or you can get default values from the driver file. The relocatable code for each driver contains default values for that driver.

For most interfaces you need enter only:

1. The default file name (usually the driver file name), and
2. The select code.

For most devices, you can use:

1. The default file name,
2. The model number, and
3. The LU of the device.

IFT and DVT Parameter Defaults

Default values for the IFT and DVT parameters are included in the relocatable code for each of the HP-supplied interface and device drivers. This information is included through the use of the GEN pseudo instruction in the driver Assembly language code. The use of this pseudo instruction for putting default information in files is discussed below.

During the table generation phase, when the generator reads an IFT or DVT command from the command file, the default values for the parameters are read from the specified file. The generator then sets the specified table parameters to these values. After this is done, the parameters in the command string are read. Any parameters found in the string will replace those defaults set previously. If contradictory parameters occur in the command string, the last parameter encountered (left to right parsing) will be used.

The default files are described in Appendix D and the default values are given in Appendix E. When overriding these values, exercise caution because the operating system may not operate correctly if the appropriate values for the table extension or parameter area are not used. Some drivers may report an error if this occurs. Refer to the appropriate driver manual for details.

For example, the relocatable file %ID*99 contains the default values for the IFT parameters for a hypothetical interface card. Defaults in the source file for that driver may be as follows:

```
NAM ID.99
GEN 6,EID.99,TX:15
.
.
.
END
```



If the IFT command used all these defaults it would look like:

```
IFT, %ID*99, SC:20B
```

The values used to build this IFT would be:

select code	20B	(from command string)
driver entry point	ID.99	(from default file)
IFT extension	15 words	(from default file)
queuing option	FI	(the queuing default)

To override any default value, specify the parameters desired in the IFT command string. For example, using the same default file as above, the command string:

```
IFT, %ID*99, SC:20B, TX:10, QU:PR
```

would cause the following parameters to be used:

select code	20B	(from command string)
driver entry point	ID.99	(from default file)
IFT extension	10 words	(from command string)
queuing option	PR	(from command string)

Defaults such as these are found in the relocatable code for both the interface drivers and the device drivers. All defaults from the driver file are output to the list file. This maintains a record of the defaults not overridden in a command string. If all the parameters for a given interface or device are omitted from the command file, the defaults from the default file will be used in the generation and displayed in the list file.

Building the Answer File

The IFT/DVT worksheet has been designed for direct transfer of information to the answer file you are building. You need transfer only the information that you have entered on the worksheet; the defaulted parameters will be supplied from the default file. Transfer information in the format shown on the table generation worksheet. You will need to enter as many IFT commands as you have IFT/DVT worksheets. For each IFT command you will enter as many DVT commands as you have devices listed on the corresponding IFT/DVT worksheet. Refer to the sample answer file given in Appendix G.

The entry of a new IFT command signals the end of the previous DVT section and the associated IFT section. After the last DVT command in the last IFT section, you must enter two END commands, one to signal the end of the DVT command and one to signal the end of the IFT command.

After entering all the IFT and DVT commands, enter the node lists from each of the IFT/DVT worksheets, in the format shown on the table generation worksheet. After the last node list, enter an END command.

Table Generation Phase Commands

During this phase, the various system tables are generated. Some of these tables require configuration information from the user (such as the interface table and the device table). Other tables are allocated by the generator and they require only that the number of entries in them be specified (such as the class table and resource number table).

These system tables are created and/or configured using the commands described in the following section. These commands must occur in the order given below. For clarity, the commands are entered with more than the required two characters.

```
IFT, ...      Sets up IFT for one interface (IF) card.
DVT, ...      Sets up DVT for a device connected to above IF card.
DVT, ...      (Note that all DVTs connected to that IF card must be
DVT, ...      entered here before entering another IFT command.)
IFT, ...
DVT, ...
END
END
*
NODE          All device node lists must be specified here.
:
END
INT
:
END          This command is needed even if INT is not required.
```

IFT Command

Purpose: The interface table (IFT) command sets up the interface table for a given interface. One IFT command is needed for each interface card used in the system. The appropriate information is supplied in the various parameters in the command string.

Syntax: IFT,file[,Eentry point],SC:sc[,QU:qu][,TX:tx][,IT:it][,AL:al][,PS]

file is the name of the file that contains the default values for the IFT entries for the interface card, for example, %ID*00.

It must be supplied in each IFT command. This parameter is usually the name of the file that contains the relocatable driver for this interface, and also contains the IFT defaults. It may be a user created file that contains the IFT defaults for this particular interface.

Specifying the file here does not cause it to be relocated as part of the operating system. The driver required for the interface must be relocated during the system or driver partition relocation phase of generation.

Eentry point

specifies the entry point of the driver called to service requests either from this particular card. This parameter must be specified in the default file or in the IFT command string. It may occur in any position following the file parameter. Refer to Appendix E for the entry points for various supported devices.

For example, if the serial interface card using driver ID*00 with entry point ID.00 is used, the entry point is specified in the IFT command as follows:

IFT,%ID*00,EID.00,SC:20B
↑ ↑ ↑
file entry point select code

If the entry point is not given, the NAM record from the file containing the default values will be used as the entry point.

SC:sc specifies the octal select code of the interface card. This parameter may be included in the relocatable defaults file but since it must be defined specifically for a given set of hardware, it is suggested that it always be specified in each IFT command in the command file.

The select code for a particular interface is defined by setting appropriate switches on the interface card itself. (Refer to the appropriate hardware interface reference manual for information on how to set these switches). The select code may be any octal number between 20B and 77B.

The select code is subject to the following constraint: if a device using a select code is privileged, all the devices using the select codes in the same group of four must also be privileged. These groups of four are:

20B-23B	50B-53B
24B-27B	54B-57B
30B-33B	60B-63B
34B-37B	64B-67B
40B-43B	70B-73B
44B-47B	74B-77B

QU:qu specifies the service order of requests to the IFT. Either FI or PR may be entered, for example, QU:FI or QU:PR. FI indicates first in first out (FIFO); PR indicates priority queuing. In priority queuing, the request from the highest priority device is queued for service first regardless of the sequence in which the requests were made. Device priority is determined in the DVT command for each device.

The defaults for the supported drivers are contained in Appendix E of this manual.

If this parameter is omitted from both the default file and the IFT command string in the command file, then queuing defaults to FIFO.

TX:tx specifies the table extension required by the particular driver for this IFT. It specifies the number of extra words to add to the IFT being generated. It may be any number in the range 0 to 511. Refer to Appendix E for the particular requirements of each driver.

IT:it specifies the interface type. The default interface type is zero. This parameter may be useful if a particular driver needs to differentiate between several interfaces. Information on the interface type is contained in Appendix E of this manual.

AL:al specifies the port map allocation algorithm. Either DY or LO may be entered, as AL:DY or AL:LO. DY indicates that the port map set used for I/O at this interface will be dynamically allocated and deallocated as required by the generating system. LO indicates that a map set is permanently locked to this interface.

The default for the AL command is determined by the specified Select Code:

SC: 20B-47B - Locked
50B-77B - Dynamic

PS specifies that this IFT is actually a dummy, that the device driver(s) associated with device(s) on the interface are pseudo-drivers. A select code may not be specified for a pseudo-driver (the select code in the IFT will be 0). If both a non-zero SC and PS are specified, an error is issued.

Description:

Each interface card in the system requires an IFT in memory. This table contains information which specifies the select code of the card itself, the driver associated with the card, and various parameters the driver may require. The first IFT command must follow the END command for the system relocation phase. After each IFT command, all the device tables (DVTs) that are to be created for all devices to be connected to that particular interface must be built before another IFT command may be entered.

IFT Example

This section provides a sample IFT entry and descriptions of the IFT parameters.

```
IFT, %ID*00, EID.00, SC:20B, QU:PR
```

This command entry indicates the following:

1. The file %ID*00 (the terminal interface driver) is to be searched for the default parameters.
2. The entry point that the driver will use is ID.00.
3. The select code of the interface will be set to 20 (octal).
4. The interface is set up for priority queuing (which means that the requests will be handled in order of their device priority).
5. No table extension is specified. This means that the default in the file (%ID*00) for the table extension will be used. Information on the number of extension words required may be found in Appendix E of this manual.
6. No port map allocation is specified. Because the select code is specified as 20B, the default is to lock a port map set to this interface.

DVT Command

Purpose: The device table (DVT) command sets up a table containing information specific to a particular device, including the logical unit number.

One DVT command is required for each device connected to the same interface card. These DVT commands must be entered immediately following the IFT entry.

Syntax: DVT,[file],[Mmodel],LU:lu[,Eentry point][,TO:to]
[,BL:bb:ll:ul][,DT:dt][,TX:tx][,DX:dx]
[,DP:start #:a:b:c:d:e][,PR:pr][,QU:qu]

To enhance readability, it is suggested that continuation lines (usable in the DVT command and NODE command) be used. Continuation lines are indicated by a dash (-) immediately following one of the commas in the normal DVT command string. Note that both the file and the model number parameter, with a comma place holder, must occur in the line containing the DVT command before the '-' indicating a continuation line. The command is continued on the next non-comment line in the command file. Note that an extra comma is not required to start the new line. The DVT command may be broken only where commas normally occur, never at colons. If continuation lines are used, the DVT command using all the options would be as follows:

```
DVT,[file],[Mmodel #],-  
LU:lu,-  
[Eentry point],-  
[TO:to],-  
[BL:bb:ll:ul],-  
[DT:dt],-  
[TX:tx],-  
[DX:dx],-  
[DP:start #:a:b:c:d:e],-  
[PR:pr],-  
[QU:qu]
```

file specifies the file that contains the DVT parameter default values for a specific device. It may be omitted if the default values are not needed. (This may be the case if no device driver is to be used with a particular device. In that case the interface driver will have to handle any I/O to the device.) If omitted, its place must be held by a comma.

This parameter is usually the name of the file that contains the relocatable driver code for this device, but it may be any user-defined file that contains only the DVT parameter defaults.

Specifying the default file name here will not relocate the file as part of the operating system. The driver required for the device operation must be relocated during the system relocation phase of generation.

Mmodel specifies the model number of the specific device described by this table. For example, the entry for a 7908 CS/80 disc can be specified as M7908_LF or M7908_CF. A model number specified with LF indicates long format and provides for longer disc LUs than the plain model number. A model number specified with CF indicates continuous format and is the standard one used. Continuous format allows you to change your CS/80 disc to a larger model and retain the LUs from the smaller model. Any of these formats can be used (there is only one model of a disc, the differences are in format only) but the recommended format is CF.

If a default file is specified, this parameter is usually required by the generator to select the parameter defaults for the specific device being configured. For example, in the case of the disc driver, DD*33, several different types of discs are supported. The correct track configuration for the given model of drive is selected using this model number parameter.

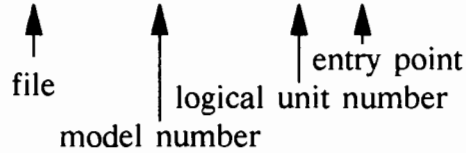
The model number may have subparameters to further specify particulars about a device. For example, the disc may be divided into multiple cartridges, each with its own DVT and set of driver parameters. (For example, M7906:0, M7906:1, M7906:2, M7906:3 are the model numbers for the four 7906 disc surfaces.) The appropriate model numbers for devices are given in Appendix E of this manual.

LU:lu specifies the unique logical unit number (LU) to be assigned to the device. Each LU is a decimal number between 1 and 255 (except disc LUs, which must be between 1 and 63). It is a number used by the operating system and users to access a particular device. It may also be used to access interface cards (such as the HP-IB card) if a DVT entry for the card itself is included in the generation.

Eentry point specifies the entry point used when the driver is called to service requests from this particular device. This parameter must be specified either in the default file or in the DVT command string if a device driver is to be used with this device. This parameter may occur in any position following the file and model parameters.

For example, if the terminal device driver DD*00 with entry point DD.00 is used, the command may be as follows:

DVT,%DD*00,M26XX,LU:20,EDD.00, ... etc.



If no device driver is to be used with the device, an entry point need not be specified. In this case any I/O request will call the interface driver associated with the IFT that this device is connected to.

TO:to specifies the decimal time-out value for the device. The value is a number which is a multiple of 10 milliseconds. This is the time a device is allowed to complete a request before the request is terminated. For example, TO:500 indicates a time-out of 5.00 seconds.

If no time-out is specified in either the default file or the DVT command string, its value will default to 0, to indicate that the device does not time-out.

BL:bb:ll:ul specifies the device buffering information. It specifies whether a device is buffered or not, the size of output buffer used for a buffered device and for class I/O buffers.

bb is set to BU to indicate a buffered device, or UN an unbuffered device. If *bb* is omitted, its place must be held with a colon and an unbuffered device is assumed.

The *ll* and *ul* parameters are the lower and upper buffer limits, respectively. The limits are not optional but may be set to 0. If these limits are not specified in either the default file or in the generator command file, they will default to 100 words as the lower limit and 400 words as the upper limit. If sixteen is used as both the upper and lower limit, only one buffered/class request may be pending on that particular device at any given time. If zero is used as both the upper and lower limits, unlimited requests will be allowed. It is recommended that you do not allow unlimited requests because System Available Memory may become used up quickly with requests for this device.

DT:dt specifies an octal number that defines the device type. It is typically the last two digits in the driver name. This parameter is required for drivers that need to make a decision based on the type of device. The device type range is 0 to 77B. Default is 70B. The device types are given in Appendix E of this manual.

TX:tx specifies the number of table extension words required by the driver. This number is given in Appendix E of this manual. It may be any number from 0 to 511; the default is 0.

DX:dx specifies the number of device driver parameters required for the DVT entries. The entries must be specified with the "DP:dp" command. This number may be in the range of 0 to 127.

If *dx* is larger than the number of driver parameters specified in the DP list of parameters, the extra driver parameters allocated are set to zero. For example, if the DVT command is:

DVT ... ,DX:4,DP:1:3:6 ...

the driver parameters in the driver parameter (DP) area will be:

DP 1 = 3 (set by DP:1:3:6)
 DP 2 = 6 (set by DP:1:3:6)
 DP 3 = 0 (set to zero by dx parameter)
 DP 4 = 0 (set to zero by dx parameter)

If *dx* is smaller than the number of driver parameters specified in the DP parameter lists, the list of driver parameters will be truncated to length *dx* and an error message will be issued. For example, if the DVT command is:

DVT ... , DX:2, DP:2:3:6:7, ...

the driver parameters would be:

DP 1 = 0 (set to zero by dx parameter)
 DP 2 = 3 (set by DP:2:3:6:7)

the rest of the driver parameter list specified with DP:2:3:6:7 (the starting *dp* parameter is 2) is truncated because space for only two driver parameters was specified with the DX:2 parameter.

DP:start #:a:b:c:d:e

specifies the starting driver parameter and a list of driver parameter values up to a maximum of five. Each parameter in the list is separated by a colon. The start # parameter indicates the first driver parameter being specified, and the following parameters are the actual driver parameter values. If more than five parameters are required (e.g., with a disc) another “DP:start #:f:g:h:i:j” entry is required for each group of additional *dp* values. For example, if the default file has specified the following parameters:

... DP:1:A:B:C:D:E, DP:6:F:G:H:I:J ...

assuming that *dx* is 10, the values put into the driver parameter locations will be:

DP 1 = A (A,B,C,D, and E specified by
 DP 2 = B DP:1:A:B:C:D:E)
 DP 3 = C
 DP 4 = D
 DP 5 = E
 DP 6 = F
 DP 7 = G
 DP 8 = H (Specified by the 2nd DP entry
 DP 9 = I DP:6:F:G:H:I:J)
 DP 10 = J

If certain parameters must be replaced (i.e., values other than these defaults are desired), the DP parameter in the DVT command can be used to effect the change. For example, if the following was entered instead of using the defaults mentioned earlier:

DVT,%DD*30,M7906:0,...,DP:1:Q:R:S;,DP:8:X:Y,

After the generator processes this command, the driver parameters would be altered from above to:

DP 1 = Q	←	specified by
DP 2 = R	←	DP:1:Q:R:S
DP 3 = S	←	
DP 4 = D		
DP 5 = E		
DP 6 = F		
DP 7 = G		
DP 8 = X	←	specified by
DP 9 = Y	←	DP:8:X:Y
DP 10 = J		

The use of these parameters depends on the device. Refer to the Driver Reference Manual for details. In the case of disc drives, these parameters are used to define the track map for each drive. Refer to Appendix C on disc configuration in this manual for more information on how to configure the discs associated with the system.

PR:pr specifies the priority (a decimal number from 0 to 63) that will be assigned to each I/O request from this device. This priority determines the order of queuing of requests for I/O only if priority queuing (QU:PR) was specified in the IFT command string immediately above this DVT section. Default is a priority of 63, the lowest priority.

QU:qu specifies the way in which I/O requests from programs will be serviced from the DVT list. Parameter *qu* may be either FI or PR to indicate first-in-first-out or priority queuing respectively. If neither queuing option is specified in the command file or the default file, the queuing defaults to priority queuing.

If FI is specified, the first request made to the device will be serviced first. If PR is specified, the requests made to a device will be serviced in order of program priority regardless of the sequence in which the requests were made.

Description:

Continuation lines may be used with DVT commands, as described under the Syntax heading. Unlimited continuation lines are allowed, and comment lines are allowed between continuation lines. For example, the DVT command

```
DVT,%DD*30,M7906:0,EDD.30,LU:33 ... etc.
```

may be split into two lines as follows:

```
DVT,%DD*30,M7906:0,EDD.30,-  
*                               With a comment here if desired.  
LU:33 ... etc.  
*  
END
```

Comments are not allowed on the same line as one of the DVT parameter list lines. Comments are only allowed on the same line as a command if all of the parameters for that command have been used or their places held by an extra comma, which is never the case for the DVT command as the parameters may be specified more than once.

END Command

Purpose: Terminates one process in the table generation phase.

Syntax: END

Description:

Instead of using only two characters, this command is spelled out for clarity. Two END commands are needed to terminate the interface and device table generation. The first END command terminates the last DVT generation. The second END command terminates the IFT generation and allows RTAGN to process the NODE command.

NODE Command

Purpose: Specifies the logical units in a node list. One NODE command is required for each node list.

Syntax: NO,lu 1,...,lu n

lu is a logical unit (LU) number. The LU numbers are usually LUs that are part of the same physical device.

Description:

The command entry can be continued on the next line by using a dash (-) as the line continuator. The dash must be used after a parameter delimiter (the comma).

The following is an example showing how node lists are established. The example assumes that the device tables have been specified for a 7906 disc drive, a 7908 disc drive with CTD, a line printer, and a digital voltmeter assigned the following LU numbers:

Device	LU Number
7906 disc LU	30
7906 disc LU	33
7906 disc LU	43
7906 disc LU	45
7908 CS/80 disc LU	10
7908 CS/80 disc LU	16
7908 CS/80 disc LU	17
7908 CS/80 disc LU	19
7908 CS/80 CTD	24
Line printer	6
Digital voltmeter	55

The following node lists are required:

```
NODE,30,33,43,45
NODE,10,16,17,19,24
END
```

Neither the line printer or the digital voltmeter require a node list because they each have dedicated controllers.

END Command

Purpose: Terminates node list input.

Syntax: END

Description:

This command allows RTAGN to process the INT command to build the interrupt table.

INT Command

Purpose: Sets up the entry in the interrupt table for a given select code.

Syntax: INT,sc,entry point

sc specifies the select code for which the interrupt table entry is being defined. The select code number must be specified in octal.

entry point specifies the name of the entry point to transfer to when an interrupt occurs at this select code. For example,

INT,25B,\$GPIO

would transfer to routine \$GPIO when an interrupt occurs on select code 25B.

Description:

The interrupt table entry for a particular select code references the IFT for that select code. This link is generated automatically by RTAGN. The INT command is needed only when relocating a privileged driver, which must be performed during the system relocation phase. See the RTE-A Driver Design Manual for a description of privileged drivers.

The interrupt table input is terminated with another END command. This command allows the generator to complete the construction of the interrupt table and proceed to the memory allocation phase and the ID segment allocation. The END command must be included in the generator answer file even if no IFT entries were included.



Memory Allocation Phase

In the memory allocation phase, memory is allocated for class numbers, resource numbers, ID segments, shared programs, system memory block, memory descriptors, System Available Memory (SAM), system common, and system message block. During this phase of the system generation, the program swapping priority and quantum time values are set. The system default libraries that are searched automatically during all program loading of disc subsystems also are specified in this phase.

The immediately following sections describe the operation of the memory allocation phase; the generator commands are then described. As with the other generation phases, the memory allocation phase commands must be entered in the order presented in this chapter.

Allocating Class Numbers

Class numbers are used by any operation that performs a class I/O. Class I/O is used for I/O without wait, swappable I/O, and program-to-program communication. Class I/O is accomplished by calls to EXEC routines 17 through 21, and CLRQ. Each outstanding class-get call requires one class number.

As a general rule, you should allocate 10 class numbers, plus one class number for each active program in a program development system. If your system is an NS/1000 or DS/1000-IV node, add 10 additional class numbers. Each subsystem also has its own requirements for class numbers; refer to the associated subsystem manual for details.

If there are not enough class numbers, some of the requests that require them must wait for class numbers to become available and could create a bottleneck. If there are too many class numbers, the unused ones waste memory (two words per unused class number).

If no class numbers are required, you must still include the CLAS command, specifying zero class numbers.

Allocating Resource Numbers

Resource numbers provide the ability to share a resource, either data or program code, among two or more programs. There should be enough resource numbers in your system to cover all of the programs that will try to use them simultaneously.

As a general rule, you should allocate one resource number per every two ID segments in a program development system. Each subsystem has its own resource number requirements; refer to the associated subsystem manual for details.

Not enough resource numbers could cause a bottleneck that slows down your programs, and too many resource numbers will waste memory (one word per unused resource number).

If no resource numbers are required, you must still enter the RESN command, specifying zero resource numbers.

Allocating Debug Table Entries

Debug table entries are used by the debugger to communicate with the program being debugged. There must be one debug table entry per program being debugged.

In a program development system, there should be at least one debug table entry per user.

One resource number is needed for each debug table entry, so the number of debug table entries cannot exceed the number of resource numbers specified.

The default, if the debug table entries are not specified, is the minimum of five, or the number of resource numbers specified. If no debug table entries are desired, you must enter 0.

Allocating ID Segments

The ID segments are tables in memory that contain information identifying programs to the system. The information (name, size, current status, and where the program resides) is compiled by the relocating loader program during on-line program loading.

A maximum of 254 ID segments, each requiring 48 words of memory, can be allocated. One ID segment is needed for each program that will be active concurrently with other programs.

The number allocated is the maximum number that will be available in the system; new ID segments cannot be created online. You can, however, deallocate and reallocate existing ID segments with the OF, RP, and RU operator commands.

In a typical four-terminal program development system with five real-time memory-resident programs, you might allocate ID segments as follows:

1	for D.RTR, the directory manager
1	for FMGR
5	for the real-time programs
4	for CI (4 copies)
4	for CI auxiliary programs CIX, WH, DL and LI
3	for PROMT, LOGON, and CM (1 each)
4	for EDIT (4 copies)
4	for compilers and linkers
8	for extra flexibility
<hr/>	
34	ID segments total

In this example, as long as you execute from a CI prompt, the ID segment used by the editor will automatically be deallocated when the editor terminates (unless you restored it with the RP command). It will be available for use by the loader and then the loaded program. In this way, one ID segment can be used by several different programs and there is no need for additional ID segments.

Allocating Memory Descriptors

Memory descriptors are used to define both reserved partitions and dynamically allocated partitions. A memory descriptor is also needed for each bad memory page (or group of bad pages) in the system.

At generation time, the generator computes the number of memory descriptors needed for dynamic partition allocation, using the following formula

$$\begin{array}{ll} [4 * (\text{number of ID segments})] & \text{for systems with CDS programs} \\ \text{or} & \\ [2 * (\text{number of ID segments})] & \text{for systems without CDS programs} \end{array}$$

The generator determines from the RPL value whether or not the system uses CDS programs and selects the proper formula for calculating the memory descriptors needed. Seven words of memory are allocated for each memory descriptor.

The number of memory descriptors allocated may be adjusted up or down by using the RS command. The minimum number of memory description allowed is 1 * (# of ID segments).

Most real-time programs should run in reserved partitions, as should heavily used programs (e.g., D.RTR). It is also useful to place frequently used NS/1000 or DS/1000-IV programs in a reserved partition. Knowing the type of programs to be run on the system will help you determine how many reserved partitions you should specify at boot time.

After space is allocated for memory descriptors, space for swap file area descriptors is automatically allocated. No user command is required because the number of words needed is determined from the number of ID segments previously specified. The generator calculates the number of swap-area descriptors necessary, using the formula $3 * (\# \text{ ID segments} + 1)$.

Allocating System Available Memory (SAM)

System Available Memory (SAM) is required for class I/O, buffered output, and string passing. The maximum size of SAM is 32K words. Refer to the System Design Manual for a detailed discussion of the SAM requirements. The size of SAM varies with the application; however, an allocation of approximately 4000 words would be appropriate for a stand-alone program development system with no extra subsystems.

The RTE-A Operating System provides SAM from physical memory, not system logical memory. System space is not affected by using SAM although the physical memory available may limit the amount of SAM that can be allocated.

Allocation of SAM is in number of words. Note that the generator rounds off the number to the beginning of the next page of memory and places the starting physical page in the system area.

Allocating Extended System Available Memory (XSAM)

Extended System Available Memory (XSAM) is used for User-Defined Search Path Directories (UDSPs), signals, and security support. XSAM is optional, and if it is not specified, it defaults to using the same memory space as SAM. The maximum size of XSAM is 32K words. Refer to the System Design Manual for a detailed discussion of XSAM.

The size of XSAM varies with the application, however an allocation of approximately 4000 words is sufficient for most applications.

Specifying Spool Buffer Limits

Buffer limits for the spool system are specified with the SL command. The maximum upper spool buffer limit is 6112 words, and the maximum difference between the upper and lower limits is 2032 words. If the upper spool limit does not leave at least 1000 words of SAM remaining, the generator will issue a warning. The command

```
SL,1000,2000
```

can be used to set the lower and upper limits for spooling. With this command, the spooling system will pause when the SAM limit of 2000 words is reached, allowing the buffers to be deallocated. Spooling will restart when 1000 words of SAM remain. You may enter SL without any limits and the generator will default the lower limit to 1/16 of SAM and the upper limit to 1/8 of SAM. If you do not have the VC+ option (HP 92078A), or if spooling is not required in your system, enter:

```
SL,0,0
```

Specifying Background Priority and Timeslice Quantum

Background program swapping priority is specified following SAM allocation. The default background priority limit is 30, but any number between 0 and 32767 can be used. (The highest program priority in RTE-A is 1.) Refer to the System Design Manual for further information.

Next, the quantum timeslice value is specified. This value is specified in milliseconds in the range of 0 and 32767. Timeslicing begins at a particular priority limit specified in the quantum timeslice command string. The default timeslice quantum is 1000 milliseconds and the default timeslice priority is 50.

Specifying Shared Programs

Following the timeslicing specification, the number of shared programs is specified. The generator creates a table of five words for each shared program. The command

```
SP,2
```

allows space for up to two shared programs in the system. This allows up to two different shared programs in the system. Each different shared program requires an entry in the shared program table. The number of copies of each shared program is limited by the number of ID segments in the system.

If you do not have the VC+ option, enter:

```
SP,0
```


Allocating an Extended Schedule Table

The extended schedule table is used by the serial drivers to contain the names of programs to be scheduled by the drivers. Space for 15 programs is provided by the drivers, but if more program names are needed, their space is reserved by this command. Up to 112 programs can be placed in the Extended Schedule Table. Each entry takes five bytes, which is rounded up to the nearest word. This parameter is optional, and if not specified, defaults to 0.

Specifying System Memory Block

After declaring the number of shared programs, a number of words is reserved for the system memory block. The block of memory is used by the NS/1000 or DS/1000-IV system for tables; refer to the NS/1000 or DS/1000-IV manuals for details. The command

```
MB,500
```

reserves 500 words for a medium size network. If you do not use NS/DS, enter:

```
MB,0
```

Specifying Concurrent Users

After a number has been entered to reserve a system memory block, the number of concurrent users on the system is specified. This is used only for the multi-user environment provided by the VC+ (HP 92078A) option. Otherwise, the number zero is entered.

For each user, the generator creates a table of 22 words. This US value entered sets the maximum number of users that can be on the system at any time in the multi-user environment. You may use one user table per terminal, plus two for NS/DS transparency, and several extra for programmatic log-on or background sessions. For example,

```
US,12
```

allows twelve concurrent sessions, suitable for an eight-terminal system. For a single-user system, enter:

```
US,0
```

Specifying LOGOF buffer limits

The LB command, LOGOF buffer limit specification, follows the concurrent users specification. LOGON is the special system program that executes when a session user logs off (HP 92078A VC+ option only). (Note that the process of logging on and logging off is handled by LOGON.) When a user logs off, RTE creates a session information buffer used by the LOGON program. The buffer requires 25 words and is allocated from System Available Memory (SAM). If many users log off at once, the LOGON program may not run quickly enough to deallocate all the information buffers, causing SAM to become fragmented or clogged.

The LB command lets you limit the amount of SAM that can be used for LOGOF information buffers. You specify the number of information buffers that may occupy SAM; if the buffer limit has been reached and another user logs off, no session information is saved, thus no CPU usage or connect time information will be recorded for the session just ended. When LOGOF has executed and cleared out all the buffers, more buffers may be allocated.

If you do not use the HP 92078A option, or if you wish to let the system use the default buffer limit, simply enter:

LB

The default buffer limit allows 25% of all concurrent users (specified by the US command) to log off simultaneously.

Setting Up System Common

There are two types of system common: labeled and blank. Labeled system common is initialized at boot with data or code relocated into the labeled system common area during system generation. Blank system common is initialized by the programs that access the blank system common area. Labeled and blank system common are accessible to a program only if the loader command LCOM or SCOM was issued at the appropriate time in the program relocation.

Modules for labeled system common are produced by the assembler or compiler. They are relocated in the system common relocation phase of the system generation procedure.

A number of words is allocated for unlabeled (blank) system common; relocatables are not needed. If no blank common is required, enter zero with the COM command.

Allocating System Message Block

Immediately after declaring system common, the system message module %MSGTB must be relocated. This module contains all of the system message pointers and tokens. After this, you must relocate the module %\$M000 to specify the English native language for the messages. (Native language modules for messages in other languages are supported in that country.)

The following sequence will relocate the system message block. An END statement is required after each module, and a second END statement is required to signal completion of the system message block relocation

```
RE, %MSGTB      (system message module)
END
RE, %$M000     (English native language)
END
END
```

The order in which the message blocks are relocated determines the definition of the System Language Numbers (SLN) 0 to 7. In the above, %\$M000 defines SLN 0. The definition of SLNs 1 to 7 will default to English native language (all non-specified SLNs default to the definition of SLN 0). The system message module and system message blocks are stored in physical memory (not logical memory) and retrieved for printing as required. Note that the total size of all system message blocks plus the %MSGTB module must be less than 32k words.

Specifying Security Tables

The security tables are added after the system message block. The default set of tables supplied with the HP 92078A VC+ System Enhancement Package are in a file called SECURITY.REL. For example:

```
RE, %MSGTB
END
RE, %$M000
END
RE, SECURITY.REL
END
END
```

Security tables should always be relocated **after** the system message block.

Specifying System Default Libraries

Following system message block allocation, the system library files are specified. These are system library files that the relocating loader LINK searches during the relocation of all real-time and background programs. Files specified here are included in the snapshot file. Libraries that are typically included are shown below. Note that the order shown must be followed to obtain the fastest loading speed.

1. \$FNDLB for systems not using NS/DS, or \$FDSL B for systems using DS.
2. PASCAL.LIB and SHSLB.LIB, the Pascal libraries, if the Pascal compiler is being used.
3. \$BIGLB, the system library. \$BIGLB contains \$FMP, %DECAR, \$HPIB, \$SYSLB, \$CMDLB, \$MATH, \$FLIB, and \$FNEWF. All these libraries are merged together and then indexed. Using \$BIGLB produces the fastest possible loading time.

A second set of libraries can be specified for VC+ systems. This set, the system CDS libraries, is used by LINK in loading CDS programs. Typical VC+ system CDS libraries are \$BGCDS and \$BIGLB.

You can also include other libraries to be searched during relocation of real-time and background programs. For example, you may include merged and indexed libraries for NS/1000, DS/1000-IV, IMAGE, Graphics, etc. The libraries required are described in the related subsystem manuals.

If you use the Vector Instruction Set (VIS) in programs that will run on hardware without the VIS firmware, the software equivalents library \$VLB6B can be included in the system library (\$VLB6B also can be searched on-line using LINK). The VIS RPL file is included in the A700 RPL files %RPL71 and %RPL73. For systems with the VIS firmware, the firmware interface library \$VLBA1 is used instead of \$VLB6B.

Memory Allocation Phase Commands

The generator commands and their required sequence of entry in the memory allocation phase are:

CLAS	allocates class numbers
RESN	allocates resource numbers, debug table entries
ID	allocates ID segments
RS	allocates reserved partition
SAM	allocates System Available Memory, Extended SAM
SL	specifies spool buffer limits
BG	specifies background program priority limits
QU	specifies quantum timeslice value
SP	declares number of shared programs, extended schedule tables
MB	specifies system memory block
US	specifies maximum number of concurrent users
LB	specifies LOGOF buffer limits
RE,	relocates system common modules
RE,	relocates security tables
END	terminates relocation of labeled common modules
COM	specifies unlabeled (blank) common
RE,	relocates system message module
RE,	relocates system message language module
RE,	relocates security tables
END	terminates relocation of system message modules
END	terminates relocation
LIB	specifies all non-CDS library files to be searched when loading a program on-line
END	terminates library specification
LIB	specifies CDS library files to be searched
END	terminates library specification

Class Number Command CLAS

Purpose: Allocates class numbers to be used in class I/O calls.

Syntax: CLAS,n
n is a number between 0 and 255.

Description:

If this command is omitted, the class error message will be issued. Based on the number entered, the generator builds a class table using $(2*n)+1$ words of memory.

If insufficient class numbers are generated into the system, programs using class numbers will be competing for this resource, resulting in degrading system performance.

Resource Number Command RESN

Purpose: Allocates the maximum resource numbers for use in the system, and allocates the maximum number of programs that can be debugged simultaneously.

Syntax: RESN,n [,m]

n is a number between 0 to 255 that specifies the resource numbers.

m is a number between 0 to n that specifies the number of programs to be debugged simultaneously.

Description:

This command may not be omitted and must follow the class number command. If this command is omitted, the resource number error message will be issued.

The generator uses n+1 words of memory for the resource number table. If insufficient resource numbers are allocated, program processing time may be increased because programs requesting a resource number may have to wait until a number becomes available.

The generator uses m+1 words of memory for the debug table. The parameter m is optional, and if omitted defaults to the minimum of 5 or n.

ID Segment Command ID

Purpose: Allocates the total ID segments for use in the system.

Syntax: ID,n

n is the number of ID segments to be allocated, ranging from 1 to 254. A number of 10 may be used for small systems; large systems with NS/DS may need 60 or more.

Description:

It is important to note that no program can run without an ID segment. ID segments may be reallocated to different programs on-line with the CI OF, RP, and RU commands. However, additional ID segments cannot be created on-line.

Memory Descriptor Command RS

Purpose: Changes the default allocation of resource numbers.

Syntax: RS,n

n is the number of memory descriptors (MDs) to add to or subtract from the standard value used for the system.

n = 0 indicates that the value is as calculated by using the formula:

$4 * (\text{number of ID segments})$ (CDS)

or

$2 * (\text{number of ID segments})$ (non-CDS)

n = positive value – specifies the number of MDs to be added to the value calculated above.

n = negative value – specifies the number of MDs to be subtracted from the value calculated above.

The minimum number of memory descriptors allocated is equal to the number of ID segments in the system.

Description:

The generator can determine from the RPL value whether the system is a CDS or non-CDS system. This information is used to calculate the default number of memory descriptors for the system, according to the formulae given above. This number is added to the number specified (n). This number (n) should also include the number of blocks of bad pages of memory. The sizes of reserved partitions are defined, and bad memory pages are specified at boot time. Refer to Appendix H for further boot information.

System Available Memory Command SAM

Purpose: Specifies the number of words of System Available Memory (SAM) and Extended System Available Memory (XSAM).

Syntax: SAM,n[,m]

n is the minimum number of words of memory to be allocated for SAM.

m is the minimum number of words of memory to be allocated for XSAM.

Description:

The actual number of words allocated will be increased, if required, to force SAM and XSAM to end on a page boundary. If the amount of SAM specified is too large, the extra portion is wasted memory.

If SAM is too small, programs using class I/O, passing strings or involving buffered I/O will be in contention with one another for the available SAM. This will result in slower operation as the competing programs will have to wait for available SAM.

If the optional Distributed System and the NS/DS transparency feature is in the system, more SAM is required. Refer to the Network Manager's Manual (part no. 91750-90010/11) for a description of SAM requirements with NS/DS.

Spooling Limits Command SL



Purpose: Specifies spooling buffer limits.

Syntax: SL[,m[,n]]

m is the lower buffer limit in number of words. Defaults to 1/16 of SAM.

n is the upper buffer limit in number of words. Defaults to 1/8 of SAM.

Description:

This command sets the limits for the spooling system. When the upper limit is reached, the spooling system is halted to allow buffers to be deallocated. The spooling system will restart when the lower limit is reached. The maximum value for the upper limit is 6112 words. The difference between the upper and lower limits must not be greater than 2032 words. Note also that the default limits will produce an error if 32 pages of SAM are specified. If the maximum limit leaves less than 1000 words of SAM available, the generator will issue a warning, but will continue.

Background Priority Command BG

Purpose: Specifies background program priority for swapping purposes.

Syntax: BG,priority

priority is the background priority limit for swapping, ranging from 0 to 32767. If this parameter is incorrectly specified or is omitted, the generator will use the default priority of 30.

Quantum Timeslice Command QU

Purpose: Specifies a quantum timeslice value for the maximum time allocated to each program. The value is specified in number of milliseconds.

Syntax: QU,quantum,priority limit

quantum is the timeslice value between 0 and 32767 milliseconds. The default value is 1000 milliseconds if this parameter is incorrectly specified or omitted.

priority limit is the priority level where timeslicing begins. All programs of equal or lower priority (higher priority number) will be timesliced. The default priority is 50 if priority is incorrectly specified or omitted.

Shared Program Command SP

Purpose: Specifies the number of shared programs and the number of extended schedule table programs in the system. These numbers indicate the number of different programs, not the total number of copies. Note that extended schedule table programs do not need to be shared programs, although it is permitted.

Syntax: SP,n[,m]

n is the number of shared programs.

m is the number of extended schedule table programs.

System Memory Block Command MB

Purpose: Specifies the size of the system memory block.

Syntax: MB,n

n is the number of words reserved for the system memory block.

Description:

The system memory blocks are used by NS/DS. Reserve 500 words for a system with 10 nodes. Refer to the DS manual set for more information.

Concurrent User Command US

Purpose: Specifies the number of concurrent users in the system.

Syntax: US,n

n is the number of concurrent users allowed at one time in the system.

LOGOF Buffer Limit Command LB

Purpose: Specifies the number of LOGOF information buffers which may occupy SAM concurrently.

Syntax: LB[,n]

n is the number of buffers. Default is 25% of number of concurrent users.

Description:

When a user logs off, a session information buffer is created by RTE which the LOGON program reads. If many users log off at once, the LOGON program may not run quickly enough to deallocate all the information buffers, causing SAM to become fragmented or clogged. You specify the number of information buffers that may occupy SAM; if the buffer limit has been reached and another user logs off, no session information is saved.

Labeled Common Commands

The commands used to set up labeled system common are the same commands used in the system relocation phase to relocate modules. These commands can be entered after the following generator prompt:

* System Common Relocation

Modules to be placed in labeled common are relocated using the RE command. Labeled common relocation is terminated with the END command. The generator will then begin the process of allocating unlabeled common.

The library \$BIGLB need not be searched as it does not contain any modules appropriate for system common; the library \$SYSLB can be searched.

NOTE

Only non-CDS modules may be relocated into labeled system common.

If you generated a modem into your system, remember to relocate the hpmdm_table.rel module in labeled common.

Unlabeled (Blank) Common Command COM

Purpose: Specifies the minimum number of words of memory for unlabeled common.

Syntax: COM,n

n is the minimum number of words to be allocated for unlabeled system common. This number can be either given as octal (indicated by appending the letter B to the number) or decimal.

Description:

Unlabeled common is a common storage area for all the programs in the system. At system boot, it is blank and must be initialized by the first program that uses it.

The actual number of words allocated will be increased, if required, to force system common to end on a page boundary.

System Message Catalog Relocation

Purpose: Specifies system error messages in the preferred language, and includes the security configuration (92078 only).

Syntax:

re,%MSGTB	message table
end	
[re,%\$M000]	English language catalog
end	
[re,other language catalog]	if appropriate
end	
[re,security.rel]	92078 only
end	
end	

Description:

The system message catalog relocation allows you to customize your system to output messages in your preferred language. The message table and one language catalog must be relocated in order to get system error messages. Up to eight catalogs may be specified in one system (seven if the security table is relocated).

The security table is relocated after the message catalogs have been relocated. The security table specifies what capability levels will be required for system and user application commands. Refer to the RTE-A System Manager's Manual (part no. 92077-90056) for further information on the security system.

Library Command LIB

Purpose: Specifies a library file to be searched by the relocating loader LINK when a program is loaded on line. This command is repeated for each library file to be searched.

Syntax: LIB,file

file is the name of a library file to be searched at the end of program loading. It can also be a directory, in which case all files in that directory will be searched. If the directory is specified in the file parameter, that directory must be available when using the relocating loader.

Description:

The system libraries are specified following system common relocation. Use an END statement to terminate the set. For VC+ systems, a second set of libraries (used for linking CDS programs) may be specified following the non-CDS library specification. Use an END statement to terminate the CDS library specification sequence.

The files specified will be searched following the search of the snapshot when a loader EN command is encountered by the relocating loader. Any number of library files may be specified, with one LIB command for each file.

Only the names of these files will be placed in the system snapshot.

SEC1000.LIB must be the first library in the non-CDS library declaration. To minimize search time, the following order of search is recommended. Delete any libraries not used extensively in your system. The deleted libraries can be individually searched as required with the relocating loader LI command.

```
LIB,SEC1000.LIB
LIB,$FNDLB          (or $FDSL B for systems using NS/DS)
LIB,PASCAL.LIB     if PASCAL is used extensively
LIB,$FNEWF
LIB,$FLIB
LIB,$FMP           if File Management or Pascal is used extensively
LIB,%DECAR        if Pascal or BASIC overlays are used extensively
LIB,$SYSLB        System Library calls
LIB,$MATH
END
```

To optimize load speed, it is best to use the library \$BIGLB, a single indexed library containing \$FMP, \$MATH, \$FNEWF, \$FLIB, \$SYSLB, \$HPIB and %DECAR. The above library search sequence would then read:

```
LIB,SEC1000.LIB
LIB,$FNDLB
LIB,PASCAL.LIB
LIB,$BIGLB
```

For VC+ systems, the suggested library sequence is

* System libraries

```
LIB,SEC1000.LIB  
LIB,$FNDLB  
LIB,PASCAL.LIB  
LIB,$BIGLB  
END
```

* System CDS Libraries

```
LIB,SEC1000CDS.LIB  
LIB,$BGCDS  
LIB,SEC1000.LIB  
LIB,$BIGLB  
END
```

SEC1000.CDS.LIB must be the first library in the CDS library declarations. Note that the list of non-CDS libraries must have SEC1000.LIB at the head of the list.

Load speed is further optimized if the user libraries are merged and indexed as well. Refer to the LINK User's Manual for more information.

System Installation Concepts and Considerations

System installation is performed after system generation to bring up a new operating system. Upon completion of the system generation, a system file and a snapshot file are created for the new operating system. These files together with certain type 6 program files produce an operating system that can be loaded into memory and made operational.

The installation procedure is different depending upon the type of system being installed. The information on system types and other overall installation considerations is given in this chapter. Installation procedures are given in Chapters 9 and 10 of this manual.

Types of Systems

Operating systems are generally classified as either memory-based systems or disc-based systems.

A disc-based system as considered in this manual is a system that boots from disc and allows program swapping and program segmentation. It provides all the features of a computer system. Such a system is useful for general purposes, including system and program development and large scale data acquisition and manipulation.

In this manual, a memory-based system is one that boots from a PROM module or a DS/1000 link. Program swapping or program segmentation is not possible in memory-based systems. Typical memory-based systems are self-contained and generally dedicated to a specific application, often in a measurement and control environment to provide fast real-time response to external events.

Other systems that vary from these strict definitions are not addressed here. However, the principles discussed are still applicable for installing other systems not covered in this manual.

Host and Target Systems

Throughout this chapter and in the installation procedures, the terms host and target systems are often used. These terms are defined as follows:

Target System = the new system. This is the newly generated system which is ready to be installed.

Host system = an existing system. This probably is the system where system generation was performed and one that will be used to install the target system.

The host system can be almost any RTE-A or RTE-6/VM operating system. This allows many different ways of installing the target system. The flexibility is discussed in the Target System Installation section in this chapter.

Type 6 Program Files

The type 6 program files required depend on the type of system being installed and the system applications. A disc-based system must include, at a minimum, the directory manager program, D.RTR, and at least one program capable of restoring other programs into the system. This latter program might be CI (it has an RP function) or some user-written program that uses a subroutine such as FmpRunProgram. The boot extension program, BOOTEX, is used to place such programs in the system at boot time.

At boot time, a memory-based system without DS must include ALL programs to be run on the system; the system is not capable of loading programs. A memory-based system with DS is capable of loading programs over DS if the APLDR program is included in the memory-based system at boot time. The BUILD program, run in the host system, is used to merge the type 6 program files with the system file to form a complete target system file. Before running BUILD to create a memory-based system or booting a disc-based system, ensure that the necessary type 6 program files are available. If the type 6 files are not available, you can create them using LINK.

The Boot Process

A system can be booted from a disc, a distributed systems (DS/1000) link, a Cartridge Tape Drive (CTD), or a PROM module. Boot can be done manually, using the virtual control panel (VCP) terminal, or you can set the switches on the processor card of the A-Series computer for automatic boot whenever the power to the computer is turned on.

When manual boot is done from the VCP terminal, enter a boot string consisting of parameters that specify the type of device from which you are booting, the physical location of the system file, and (when booting from disc) the name of the system file. These parameters are described in Appendix H. Continuation of the boot procedure depends on the media, as described in the following subsections.

Boot From Disc

When boot is from a disc, the VCP/loader ROM loads into memory and executes the "file" specified by number in the boot string. This file contains the boot extension program, BOOTEX, which is used to load the operating system into main memory.

The operation of BOOTEX depends on the type of file specified in the boot string parameters. If the file is a type 1 file, BOOTEX assumes that it is a system file that is ready for boot. BOOTEX tells the VCP loader to copy such a file into memory and execute it. If the file is a type 4 file, BOOTEX assumes that it is a boot command file and follows its instructions for boot. If a file is not specified, BOOTEX searches the boot cartridge for a file named BOOT.CMD::SYSTEM, or SYSTEM::0 if the LU specifies an FMGR cartridge. If unable to find the file specified, BOOTEX switches to interactive mode and prompts for boot instructions.

Boot From Cartridge Tape Drive

When boot is from the Cartridge Tape Drive (CTD) unit of a 7908, 7911, 7912, 7914, 7942 or 7946 disc drive, the VCP/loader ROM moves the tape forward to the specified file. It then loads the system file from the tape and executes it.

Boot From PROM

When boot is from a PROM module using file number zero, the VCP/loader ROM code starts at the beginning of the PROM module, loads in the file it finds there, and executes it.

Boot From DS Link

When boot is over a DS link, the VCP loader ROM code at the destination system interacts with program PROGL, which runs at the DS neighbor of the destination system. Program PROGL is described in the DS/1000-IV Network Manager's Manual.

The operating system being booted (loading into memory in the destination computer) need not be on disc at the neighbor node. It can be at any node on the DS network, and the neighbor node can use the store-and-forward version of PROGL to boot the operating system into the destination computer.

Also, the destination computer need not be a node on the DS/1000 network, however, it must be connected by a DS/1000-IV link to the neighbor that is running PROGL. The VCP loader in the destination system can respond to the PROGL queries over the DS/1000 communications line.

When a file number and the destination computer select code of the DS/1000-IV communication line are specified at the keyboard of the destination computer VCP terminal, the VCP loader sends the file number to the program PROGL at the other end of the specified line. Whether or not the VCP terminal itself is remote over a DS link is an independent option.

Program PROGL translates the file number it receives into a file name and sends the contents of this file to the VCP loader on the destination system. The file must be either a type-1 system file acceptable for disc boot (created by BOOTEX or BUILD) or an absolute binary file created from such a system file by binary to absolute system utility MI2AB.

Target System Installation

There is a great deal of flexibility in setting up a working system. You might install a bootable system directly on the medium from which it will be booted, or you might make an intermediate copy of a bootable system and transport it for use in the final destination. Following are a few examples (not an exhaustive list) of variations in the software installation process.

If you have a CS/80 disc with CTD (7908/11/12/14/42/46) you can install your system from the CTD tape to disc. This involves placing the tape cartridge in the CTD and performing the off-line restore procedure. The RTE-A Primary System Installation Manual supplied with your A-Series system describes the procedure.

On an RTE-6/VM host system, you can install a memory-based RTE-A target system on a CS/80 cartridge tape using the CSYS utility. You can then transport the tape to an A-Series computer and load the target system directly from the tape.

If you have a CS/80 disc without cartridge tape (7933/35/41/45), you can install your system from magnetic tape to disc. This involves booting into memory from magnetic tape a memory-based RTE-A system containing the ARSTR utility and using this utility to restore your system from another magnetic tape. The procedure is described in the RTE-A Primary System Installation manual and the RTE-A Utilities Manual.

For a special purpose memory-based target system, you can use the host system to program the target system PROMs. Then install the PROMs on a PROM module on the target computer, and boot the target system.

If you need to keep several systems available on your A-Series computer, and the systems all use the same disc configuration, you can keep them all on the same bootable disc LU. You can boot the system you want just by specifying the name of the appropriate system file or boot command file when you enter the boot string.

If you have several memory-based systems located around a factory floor (in a process control application, for example) you can use a central program development system to generate new operating systems for these peripheral computers, and then place those new operating systems on floppy discs. You can then place a floppy disc drive on a cartridge and boot each system from floppy disc.

Multiple memory-based systems become even easier to install with DS. Each memory-based destination system can be set to power-on auto-boot from a single source system, using a different system file, if desired, for each memory-based system.

If the destination systems are directly connected to the source system by DS link, the VCP terminals of the destination systems can be a terminal on the source system, and booting can even be initiated from the source system.

Disc-based destination systems can also be booted over DS. For example, if you have several disc-based systems that are to be run by identical software, you can boot from a master system file on the source system. Thus, updates to the master system file will automatically apply to all destination systems booted from that file.

Disc-Based System Installation Procedure

This chapter provides the installation procedure for a disc-based system. It is assumed that your system is to be booted from a disc drive, although it is possible to have a disc-based system that does not boot from a local disc (from a PROM module or DS).

For a disc-based system, the required items are:

1. System file for target system.
2. Snapshot file for target system.
3. Type 6 program files.
4. Optionally, a boot command file.
5. A valid BOOTEX initialized at cylinder 0.



A summary of the steps in a disc-based system installation is given below:

1. Prepare the boot command file.
2. Install the system, snap, BOOTEX and boot command files on a bootable medium.
3. Boot the target system; confirm that it is operational.
4. Check the target system for proper operation.
5. Set up startup program or a WELCOME file.
6. Back up the target system.
7. Optionally, set switches for automatic boot.

Prepare the Boot Command File

The boot process for a disc-based system uses the boot extension program BOOTEX. BOOTEX has several functions. In addition to booting the system into memory, it is also used for mounting disc LUs, defining partitions, setting up the swap area, and relinking and restoring any programs to be included in the system at boot time. Complete details on the operation of BOOTEX, including error messages, can be found in Appendix J.

Use the text editor to create a boot command file that contains the commands you want to enter. The following paragraphs describe the commands that are available to you.

Echoing Commands

You may enter the

EC

command to echo all commands to the VCP terminal. This command must not be entered in the middle of an RP sequence (see below). If the VCP device is a neighboring DS node, commands will not be echoed.

Mirrored Disc Configuration

The MI command applies to Datapair/1000 configurations only. Use this command to specify a mirrored configuration information file. The syntax is:

[MI,filedescriptor]

where the filedescriptor is the mirrored configuration information file name. For more information, refer to the System Generation and Boot chapter of the Datapair/1000 Reference Manual (part no. 92050-90001).

Specifying the System

The following four commands specify the system file that will be loaded into memory and its associated snapshot file. The commands are:

- [S2,file] specifies the new copy of the system file.
- [MI,file] specifies information file for DATAPAIR system.
- SY,file specifies the system file.
- SN,file specifies the snapshot file.

The S2 command is optional and, if specified, must be entered before the SY and SN commands. S2 specifies a copy of the system that will overwrite the SY file before booting. This will prevent your system from being corrupted if power should fail during the system boot-up.

These commands must be entered before any of the commands listed below. The files you specify must be on the file system volume or the FMGR disc cartridge from which you boot. These commands must be entered to BOOTEX before any others.

Specifying Memory Size

The MS and SA commands are options that specify memory available to the system. The MS command specifies the main memory size. If omitted, all memory is used. The SA command specifies the number of words of SAM and XSAM available in the boot command file. This value may be either less than or greater than the value specified in the generation. If omitted, the value specified in the generation is used. Note that BOOTEX rounds the value specified up to the next page boundary.

MS,number of pages

SA,[number of words of SAM],[number of words of XSAM]

Possible SA commands;

SA,n1,n2	set SAM to n1 words, XSAM to n2 words.
SA,n1	set SAM to n1 words, leave XSAM as generated.
SA,,n2	set XSAM to n2 words, leave SAM as generated.
SA,n1,-	set SAM to n1 words, XSAM and SAM in same map.
SA,,-	leave SAM same as generated, XSAM and SAM in same map.

Specifying Bad Pages

The BP command allows you to specify any bad pages of memory:

BP,number	identifies the page specified by number as a bad page.
BP,num1,num2	identifies a block of bad pages from num1 through num2.

(The page number of a bad page is printed any time a parity error occurs. You should keep a list of bad pages at your system console.) You must specify all bad pages in memory before you set up the user partitions. Bad pages must be specified in increasing order.

Mounting Disc Volumes

BOOTEX will mount any disc volumes you want to appear in the cartridge list of the target system when you boot it up. Use the MC command to mount a volume:

MC,lu mounts the disc LU specified by the lu parameter. This LU is defined in the target system. The lu parameter may be either positive or negative; it makes no difference. Note that you are specifying an LU, not a FMGR CRN.

BOOTEX mounts the LU containing the system. You must mount any other volumes that contain the type 6 files for any programs that will be restored by BOOTEX at boot time. (You must mount a disc volume before you set up a swap area or restore a type 6 file on that cartridge.)

Setting Up the Swap File

The swap file command allows you to specify a file to which programs will be swapped. You can specify the size of the file, in 128-word disc blocks, in the filedescrptor portion of the command string:

SW,filedescrptor

BOOTEX searches for the named file on the indicated directory (or on the first volume mounted if a directory is not specified). If BOOTEX does not find this file, it creates a swap file by that name. If a FMGR LU is specified, the file is created with security code SW. Note that if BOOTEX finds the swap file, it will use the existing file, which may not be the same size as was specified in the filedescrptor string.

If this command is omitted, BOOTEX creates a swap file named SWAP of the default size on directory /SYSTEM. BOOTEX calculates the default size as

$32k * \text{number of system ID segments}$

This default swap file size is often much larger than you would use in practice. You can usually specify a smaller swap file to save space, and yet not affect system performance. BOOTEX will issue a warning, but will create the swap file, if you specify a file size that is smaller than the default size. The minimum swap file size is 3 blocks.

Swap file size should be chosen with care. The swap file should be large enough to handle all system needs; system functions will be affected when the swap file fills up. For example, if a space in memory for a high-priority program cannot be created by swapping out lower-priority programs, the high-priority program cannot be run. Conversely, a large but mostly empty swap file wastes disc space and in no way improves performance.

To calculate an approximate swap file size for your system, first estimate the number of programs that may be active simultaneously, and the total number of memory pages they require. Then subtract the number of physical memory pages you have available to run programs. The difference is the number of pages needed for program swapping. (Allow some extra space for fragmentation of the swap file.) Multiply this figure by 8 to obtain the number of 128-word blocks you should specify as the swap file size:

$$(\text{total required pages} - \text{available pages}) * 8 = \text{swap file size}$$

Specifying the Scratch LU

You may specify the VMA backing store scratch files with the SC command:

```
SC[,lu]
```

Only a FMGR LU can be specified in the SC command. That disc LU will be reserved for use by the operating system to hold the backing store scratch files. The LU may have been previously mounted by an MC command (but need not have been).

Turning Session Accounting On/Off

For systems with multiple users (HP 92078A VC+ option), information about the CPU time and time spent logged on is maintained by the system. Gathering this information is called "session accounting". Session accounting incurs a small amount of system overhead. If no accounting information is desired, accounting can be turned off so the overhead is eliminated and system performance is increased.

The AC command can turn session accounting on or off:

```
AC[,ON/OFF]
```

If neither ON or OFF is specified, session accounting defaults to ON. If the AC command is not specified at all, the session accounting state carries over from the last time the system was booted.

The system utility program METER uses the accounting information to display CPU use statistics. If session accounting is turned off, attempting to run METER will result in an error message.

Setting System Parameters

The QU and BG commands let you change the timeslice quantum and the priority swapping boundary. These commands may be entered at any point in the boot process, except in the middle of an RP sequence.

QU,timeslice,priority	sets the timeslice quantum (in units of milliseconds) and specifies the priority at which timeslicing is to begin. If programs of equal priority are competing for CPU time, and those programs have a priority value equal to or greater than that of the priority parameter, then each such program will execute for the specified number of milliseconds and then suspend to allow the next program of the same priority to execute for the same time period. Programs with priority values less than the specified priority are not affected by timeslicing.
BG,priority	sets the boundary between real-time and background programs. Programs with priority values less than that specified here are real-time programs; the remainder are background programs. Real-time programs have some advantages over background programs in swapping. (See the System Design Manual for details.)

Setting Master Security Code

The SS command establishes the master system security code, as:

SS,cd

Specify the master security code as a one- or two-character set of letters or as a number within the range of 1-32767. The letters can be entered in either uppercase or lowercase; BOOTEX converts all letters to uppercase.

Restoring Programs

You can use BOOTEX commands to restore programs. As a minimum, you must restore the directory handling program (D.RTR) and at least one program capable of bringing other programs into the system (usually CI). The commands are:

RP,filename[,newname]	creates an ID segment for the program in the system. If the newname parameter is entered, the program is given that name; otherwise, the first five characters of the name of the type 6 program file are used.
-----------------------	---

Along with an RP command, you can specify several subcommands that modify how the program will run. The subcommands must immediately follow the associated RP command. The subcommands are:

SZ,nn	resets the size of the program to nn pages.
PR,n	resets the priority of the program to n.
ST[,p1. . .[,p5]]]]	identifies this program as the startup program that is executed at boot and, optionally, passes up to five RMPAR parameters to it.
EN	terminates this RP sequence. This sequence will also be terminated by another RP command.

Reserved Partition Definition

Two kinds of partitions are used in RTE-A: reserved and dynamic. Reserved partitions are of a fixed size and may have programs assigned to run in them. Dynamic partitions are of variable sizes, are allocated automatically by the operating system as they are needed, and are then de-allocated automatically. Only reserved partitions are explicitly defined.

One way to define a reserved partition is by assigning a program to it; each time the program is executed it will then run in the specified reserved partition. To assign a program to a reserved partition:

```
AS progname[ progname][ progname] . . . .
```

The size of the specified program (progname) determines the number of memory pages allocated for the reserved partition number. Up to 20 programs may be assigned to one partition. Each AS command creates one partition. All programs specified will be assigned the partition. Usually, there will only be one per AS command. The partition will be big enough to hold the biggest program named. If progname is a CDS program, the command

```
AS,progname[:D]
```

will assign its data segment to a reserved partition. The :D entry is optional, since the default is to assign the CDS program data segment to the reserved partition. Use the command

```
AS,progname:C
```

to assign the code segment to a reserved partition. In this case, the :C entry is required.

CAUTION

If you do not have enough dynamic memory to accommodate the primary program (usually CI), you must be certain that the program is assigned to a reserved partition.

You may use the RV command to define reserved partitions without assigning programs to them:

RV,k

defines a partition with “k” pages. You may give enough AS and RV commands to define all of memory to contain only reserved partitions or, if no AS or RV commands are given, only dynamic partitions will be used.

CAUTION

Real-time programs should be assigned to reserved partitions to eliminate competition with other programs for dynamic partitions.

If an RP'd program uses shareable EMA and specifies that the shareable EMA area reside in a particular reserved partition, the size of the EMA area will be taken into account when the system is creating the reserved partition during the AS/RV phase. If the partition is sized larger to account for the shareable EMA area, the following message will appear:

partition will be used for shareable EMA – nn pages in partition

where nn is the new partition size.

Terminating Input

The command

EN

is used to indicate the end of command input to BOOTEX. An end-of-file or a CRLF (carriage return and line feed) is also interpreted as an EN command.

Sample Boot Command File

EC,	command echo
S2,SYSTEM2.SYS	system copy file name
MI,file	information file name for DATAPAIR system
SY,SYSTEM.SYS,	system file name
SN,SNAP.SNP,	snapshot file
RP,DRTR,D.RTR	restore DRTR with name D.RTR
RP,DERR,D.ERR	restore DERR with name D.ERR
RP,LOGON	prog PROMT automatically loads this; if system has minimum ID segments, should RP this program here.
RP,CI,CM	restore CI
RP,CI	assign CI as startup program, using welcome file 1
ST,,1	end of RP phase
EN	
SW,SWAP::::1024	set up swap file
AS,D.RTR	define reserved partition 1, assign it to D.RTR
RV,28	define reserved partition 2 to be 28 pages
EN,	terminate command input

Installing System, Snapshot, and Command Files

The next step is to install the system, snapshot, and boot command files on a bootable medium. For a disc-based system, this medium is a disc volume.

Before performing any of the disc LU initialization and installation steps, be sure that you are familiar with the considerations discussed below.

Installation Considerations

The LU you boot from must be located at cylinder 0, sector 0 of one of the disc surfaces. For CS/80 discs, there will be only one such LU; 7906 removable platter discs have several. The correct LU will probably be the one with the lowest number on the drive.

The most common boot procedure is to put BOOTEX at the beginning of the LU. There are different procedures for putting BOOTEX at the start of the LU, depending on whether the LU is a FMGR cartridge or a CI disc volume. For FMGR cartridges, an empty BOOTEX file is created by the FMGR IN command, described in the RTE-A Utilities Manual.

For other disc volumes, the CI IN command is used to reserve space for BOOTEX, which must be 768 blocks. The following sequence can be used to initialize an LU, which is currently empty and mounted to FMGR, as a CI disc volume:

```
FMGR: CI
CI> in 13
Re-initialize valid directory [N]? y
Initializing disc
CI> _
```

FMGR can no longer access LU 13 as a result of the above steps.

CI may access both FMGR and CI volumes but FMGR cannot access CI volumes. In order to convert a disc LU from FMGR to CI format (and take advantage of the hierarchical file system), it must be mounted and then initialized from CI.

Be aware that the CI initialization destroys all data on the LU, so any important files must be saved before the initialization. To ensure that all data is recoverable, perform a pushbutton backup of the CS/80 disc. This allows recovery in the event of even the most catastrophic errors. A TF backup may also be sufficient for these purposes as long as the boot extension file is preserved.

On the primary system, LU 16 is the first LU on the CS/80 discs. It touches cylinder 0, which means it is possible to boot from this disc LU. To prepare the disc for boot, it must be initialized so that the boot extension (BOOTEX) can be placed beginning at cylinder 0 of the disc. BOOTEX requires 768 blocks, so this area must be made available at the beginning of the disc LU. This is done with the CI IN command illustrated in the procedure given in the next section, "FPUT and BOOTEX for New Disc Configuration".

After the area is made available with the CI IN command, BOOTEX can be placed in this 768-block area. Note that the 768-block area cannot be accessed by CI, and the BOOTEX file does not appear in any directory.

Program INSTL is used to create a valid copy of BOOTEX on a bootable disc LU. Refer to the RTE-A Utilities Manual for more information on INSTL. The system, snapshot, and an existing RTE-A BOOTEX file are required to run INSTL. An RTE-A BOOTEX previously installed in the reserved space on a CI disc volume will not work since it cannot be accessed as a file. The destination is also not a "file", but rather the 768 blocks set aside by the IN command. For INSTL to properly place BOOTEX in this area, a displacement of 0 is substituted for the destination file name.

FPUT and BOOTEX for New Disc Configurations

Often a new disc LU configuration (also called a track map) is created and generated into a target system that is to be installed to replace the present system. Preparing a disc LU that will boot the new target system is more involved than just installing the current system on another disc LU.

In a hypothetical example, the primary system disc configuration for the 7908 has been changed from LUs 16 – 20 to LUs 16 and 17. LU 16 describes the same disc area in both configurations, and LU 17 of the new configuration describes the area formerly described by LUs 17 – 20. LU 16 is the only one that begins at cylinder 0, and thus is the only one that can be booted from. In this example, the new configuration is to be installed as a CI-based system on LU 16 so that it may be booted from there.

It is important that, before initializing any discs, the current (host) system should be adjusted so that its "system" disc is not LU 16. This is because LU 16 will need to be initialized, and this is not possible if active type 6 programs and the SWAP file are located on it. Therefore, all files from LU 16 should be saved elsewhere, and the files necessary to sustain a functional system should be placed on another LU, LU 17 in the following example. The boot extension will have to remain on LU 16, as well as copies of the system and snapshot files. There should also be a copy of a boot command file on LU 16 that directs the mounting of LU 17 and restoring programs (D.RTR, etc.) from LU 17. The SWAP file should also be specified to exist on LU 17. (If the system is already a CI-based system on a CS/80 disc, then it already has been initialized with the boot extension in the 768-block area of LU 16.)

Assuming that the new disc configuration has been generated into a target system, it is now time to install that CI-based configuration on LU 16. It is recommended that you do a pushbutton backup of a CS/80 disc and a TF or FST backup of the entire disc before performing any of the initialization steps.

Note that current LU 16 is the same as target LU 16. It is essential to have at least one LU on both the host and target system describing the same physical disc area. The current LU 16 should be backed up so that it may be initialized without loss of data. If those steps have been taken, proceed with this example. Run the CI program and initialize LU 16 with a 768-block space for the boot extension:

```
CI> IN 16 768
Re-initialize valid directory [N]? yes
Initializing Disc
CI> _
```

Next, make the target snap, system and boot extension files available on a disc on the system. In the example below, these files are located on the directories NEWSYS (target) and HOSTSYS (host). These files are for use by INSTL and FPUT, which should also be available on the system. A BOOTEX file is necessary, not the installed one used to boot the current host system. One is provided in the RTE-A software.

Now run the INSTL program. In many cases, INSTL will have no way of knowing about the new LU on the target system, since that LU is not a disc LU on the host system. In this case, LU 16 describes the same area on both systems, so there is no problem. Assuming that there was a discrepancy in the physical disc areas described by LU 16 on the two different systems, run INSTL and create a BOOTEX file on directory NEWSYS:

```
CI> wd /newsys
CI> INSTL
Enter snap file, system file, destination file, lu, and source file
snap.snp,prmsys.sys,bootex,16,bootex::hostsys
INSTL end. BOOTEX::NEWSYS:1:768 is your boot extension file.
warning: boot file must be at cylinder 0 sector 0
CI> _
```

At this point, a file exists called BOOTEX::NEWSYS that is usable, but is in the wrong place. The FPUT program allows you to move it into the 768-block area on LU 16 (host LU 16), using the sequence:

```
CI> fput
Usage: RU,FPUT,filename,lu,[offset]
CI> fput bootex::newsys 16 0
CI> _
```

In reality, since LU 16 on both systems describes the same disc area, FPUT is not necessary. INSTL alone may be run with the following runstring, which will place the boot extension in the 768-block area of LU 16:

```
CI> wd /newsys
CI> INSTL
Enter snap file, system file, destination file, lu, and source file
snap.snp,prmsys.sys,0,16,bootex:-32767:hostsys
INSTL end. Your boot extension has been installed at boot block 0, on lu 16.
CI> _
```

The remaining steps for establishing LU 16 as a bootable CI-based system disc are contained in the following section. For more information about the FPUT program, refer to the RTE-A Utilities Manual.

Preparation for Boot

Once a valid copy of the boot extension is installed on LU 16, the system, snapshot, and boot command files must be placed on the LU, as well as some necessary programs such as D.RTR and CI. To accomplish this, the directories HELP, SYSTEM, PROGRAMS, and LIBRARIES must be established on the LU using the crdir command as follows:

```
CI> crdir /system 16
CI> crdir /programs 16
CI> crdir /libraries 16
CI> crdir /help 16
CI> _
```



Now that you have the four required directories on the disc LU, copy the system and snapshot files to directory SYSTEM, first setting the working directory to /SYSTEM. The system file should have a file type extension of SYS, and the snapshot file a file type extension of SNP. If the S2 command is used in the boot file, you will need two copies of the system file.

```
CI> wd /system
CI> co newsys::-17 system.sys
Copying NEWSYS::-17 to SYSTEM.SYS ... [ok]
CI> co newsys::-17 system2.sys
Copying NEWSYS::-17 to SYSTEM2.SYS ... [ok]
CI> co newsnap::-17 snap.snp
Copying NEWSNAP::-17 to SNAP.SNP ... [ok]
CI> _
```

Next, you need to create a boot command file for the new system. Usually modification of a currently existing one is all that is necessary. The default file name BOOT.CMD must be used for automatic boot, and the file must be located on directory SYSTEM.

Once the system, snap and boot command files are in place, the required programs should be copied onto the directory PROGRAMS. These usually do not need to be reloaded, unless there are RPL changes in the new system. If you already have a directory called PROGRAMS, you should create a new directory with a unique name and then rename both directories when the setup is complete.

```
CI> crdir /targetprograms 16
```

The directory TARGETPROGRAMS will eventually be renamed PROGRAMS when it is ready.

```
CI> wd /targetprograms
CI> co ci.run::programs ci.run
Copying CI.RUN::PROGRAMS to CI.RUN ... [ok]
CI> co drtr.run::programs drtr.run
Copying DRTR.RUN::PROGRAMS to DRTR.RUN ... [ok]
CI> co proga.run::newlyloaded proga.run
Copying PROGA.RUN::NEWLYLOADED to PROGA.RUN ... [ok]
CI> _
```


If your system uses VC+, the boot command file above also expects the program PROMT, so it should be copied as well. PROMT is described in Chapter 11.

Note that the program D.RTR is named DRTR.RUN on directory PROGRAMS. This is because the type extension suffix .RUN is used with all program names and two periods are illegal in a file name. In the boot command file, RP DRTR.RUN and re-name it as D.RTR to match its ID segment in memory.

When CI is designated as the startup program, it searches for a file called WELCOME_{xx}.CMD::SYSTEM, where _{xx} is the second parameter of the ST command in the boot command file. In the sample boot command file the command is ST,,1 thus CI will transfer to file WELCOME1.CMD::SYSTEM. When this transfer file is completed, CI will terminate and its ID segment will be released. Some provision must be made in the welcome file to run a permanent CI program or other program that can schedule other programs. In the example given below, PROMT is enabled as the primary program. It will schedule LOGON, which will run CI for qualified users. In a non VC+ environment, an alternate name should be used for the startup version of CI. This version will then be OF'd when it completes.

The transfer file should contain valid CI commands that will be executed at startup. The following is an example of what this file in the VC+ Multiuser environment might contain:

```
RP,PROMT.RUN
CN 1 20B PROMT
CN 3 20B PROMT
CO message.txt::system 1
EX
```

The enabling of program PROMT at terminal LUs 1 and 3 is part of the user accounts system discussed in Chapter 11. For systems that do not have VC+ and the user account system, PROMT should be replaced with CI as the primary program enabled at each terminal. In such a system, the startup program should be a second version of CI with an alternate name (such as CI.A). When this program completes execution of the welcome file, its ID segment will be released. If the name is CI, and the primary program is CI, there will be not be any RP'd version of the program available to schedule. The following is an example of the commands that should be in the boot command file for this system:

```
** RP program CI
RP,CI.RUN::PROGRAMS
** RP the startup version of CI
RP,CI.RUN::PROGRAMS,CI.A
** Make it the startup program, transfer to WELCOME1.CMD::SYSTEM
ST,,1
```

When TARGETPROGRAMS is ready to be used (just before booting the target system), the directory PROGRAMS may be renamed to OLDPROGRAMS, and the directory TARGETPROGRAMS renamed to PROGRAMS:

```
CI> rn /programs /oldprograms
Renaming /programs to /oldprograms ... [ok]
CI> rn /targetprograms /programs
Renaming /targetprograms to /programs ... [ok]
CI> _
```

The following section describes the remaining directories and programs needed to complete your system before booting.

Establishing New CI Directories & Programs

When a CI-based file system is used, the directories SYSTEM, PROGRAMS, LIBRARIES, and HELP are required. A SCRATCH directory also is recommended for programs (such as EDIT, Macro, LINK, etc.) that require work areas on disc. If they do not already exist, they may now be created (on LU 16 for this example) using the CRDIR command:

```
CI> crdir /system 16
CI> crdir /programs 16
CI> crdir /libraries 16
CI> crdir /help 16
CI> _
```

If these directories already exist elsewhere on a CI disc volume, they may be used by the target system as well when it is booted. However, if the target system is ultimately to replace the current host system, it is necessary to establish at least a new SYSTEM directory on the boot volume.

In order to avoid the duplicate directory name error, create the directory with a different unique name (e.g., SYSTM1) on the proper LU:

```
CI> crdir /systm1 16
```

Before actually booting the system from this volume, rename the existing SYSTEM (to OLDSYS, for example) and rename the SYSTM1 directory as SYSTEM using the following sequence:

```
CI> rn /system /oldsys
Renaming /system to /oldsys ... [ok]
CI> rn /systm1 /system
Renaming /systm1 to /system ... [ok]
```

The system may now be booted using the directory SYSTEM. If two disc volumes are mounted and a duplicate directory name exists on the second volume, it is ignored and the one on the first volume mounted is used.

CI automatically looks to directory PROGRAMS for type 6 program files, and the LINK program looks to directory LIBRARIES for library files when loading programs (unless otherwise specified). The SYSTEM directory is used during boot, and the HELP directory contains help files for use by the “?” command of CI. If the directories do not exist or are improperly named, the CI program will be crippled since it expects them to be available for normal operations.

SYSTEM Directory

The SYSTEM directory should contain the system, snapshot, boot command, and welcome files for the system. This is the directory sought by the boot extension at boot, so it must exist on the boot disc LU. It must also contain the EDIT HELP file and the Macro error files.

HELP Directory

The HELP directory is designed to contain text files (type 3 or type 4 files) created as a quick operator reference. For example, for help on the WH command, enter ? wh from CI. This causes CI to look in directory HELP for a file named WH. If found, the contents of the file are displayed at the screen. You can create your own help files and store them in the HELP directory.

CATALOGS Directory

The CATALOGS directory contains files that hold messages that are issued by certain programs. When one of these programs runs, it obtains all of its output messages from the related message file. The file names generally begin with the character “>”. For example, the file >LK000 contains messages used by LINK.

LIBRARIES Directory

This directory is automatically accessed by LINK when relocating a program. The LIBRARIES directory should contain all libraries that might be required to accomplish a program load. At a minimum, this directory should contain the libraries specified at generation time to be searched whenever a program is linked. Other libraries may also be included here.

PROGRAMS Directory

Executable programs (file type extension .RUN) reside in the PROGRAMS directory. This directory is automatically searched for programs such as DL, SP, IO, and WH. For example, when DL is entered from CI, CI searches for /PROGRAMS/DL.RUN.

SCRATCH Directory

The SCRATCH directory is a convenient location for programs that need a work area on the disc. The SCRATCH directory is not mandatory; however, if the working directories are write-protected, the SCRATCH directory can be used for work files for programs such as EDIT. You should create the SCRATCH directory on a disc volume with a large amount of free space. Use the CI command `prot /SCRATCH rw/rw` to allow everyone write access to the directory.

CMDFILES Directory

The CMDFILES directory is a convenient location for CI command files, which should have the file type extension CMD. If User-definable Directory Search Path (UDSP) 2 is not defined, this directory is automatically searched for the command file during execution of the CI TR command. Refer to the RTE-A User Manual for more information on the TR command and UDSP.

CI Commands Implemented in Separate Programs

The Command Interpreter (CI) delegates some of its functions to auxiliary programs. While CI itself executes most CI commands, several are executed by auxiliary program CIX, and others are auxiliary programs in themselves. For example, CI implements the DL command by running the program DL.

For each of these programs, the memory-image program file must be accessible to FmpRunProgram. It is most convenient to locate these programs in the directory PROGRAMS; however, refer to the description of UDSP in the RTE-A User Manual for other options. The following are the auxiliary programs in the CI set, and the commands they execute.

ask.run	ASK
cix.run	AT,CL,CO,CR,CRDIR,DC,IN,MO,PROT,PU,OWNER,TM,UNPU,RN
dl.run	DL
io.run	IO
li.run	LI
path.run	PATH - VC+ ONLY
sp.run	SP - VC+ only; implemented by four programs (including sp.run): /programs/smp.run /programs/spget.run /programs/outpt.run
wh.run	WH

Format System

It is a good idea to generate a system with the FORMC, FORMT, or FORMF program in it, using the target system I/O configuration. FORMC is used for sparing bad tracks on the 7908, 7911, 7912, 7914, 7933, 7941/42, 7945/46 (CS/80) discs, while FORMT is used for formatting 7906, 7920, 7902, 9121 and 9895 mini-discs, and 7925 discs. All other non-CS/80 discs are spared transparently using the FORMF utility. The FORMT program requires a dedicated system with only one ID segment, the one for FORMT. Put this system, the “format” system, together using the BUILD program.

Create two boot command files: one to boot the target system (with the reconfigured disc structure) from a floppy disc, and one to boot the new system from your hard disc.

Store the format system, the target system, CI, D.RTR, and INSTL program files, the two boot command files, and the snapshots for the systems on one floppy disc or cartridge tape. Boot the target system from the disc/tape and verify operation.

If you are using a non-CS/80 disc, boot either FORMF or FORMT and format your disc. If you are using a CS/80 disc, use FORMC to spare any known bad tracks.

Boot the target system from floppy disc or cartridge tape. Mount, initialize, and run the INSTL program on the reformatted hard disc cartridges. Copy the target system file, its snapshot file, the boot command file, the D.RTR program file, and the CI program file to the hard disc.

Boot the target system from the hard disc and copy your files from the backup floppy discs or cartridge tape.

Booting the Target System

Now that your files are installed on the hard disc at a bootable location, you can boot your system. This also is the time to rename your SYSTEM and PROGRAMS directories if you built a new one with a different name.

The general boot procedure is:

1. At the terminal that is set up as the virtual control panel, press the BREAK key to enter VCP mode. The VCP will prompt for input by displaying the contents of all of the registers, followed by the VCP prompt "VCP>".

2. Enter a boot command of the form

`%Bdd[ffbusc[file]]`

The boot string must be continuous (no embedded blanks). In the "ffbusc" substring, all digits must be octal, and leading zeros can be omitted. If all digits are zero, the substring can be omitted.

`%B` = identifies following string as a boot command (load a system and execute it).

`dd` = device from which you are booting:

DC = disc drive or Cartridge Tape Drive (CTD) via HP-IB

DI = 248x integrated discs

DS = disc drive via NS/1000 or DS/1000-IV link.

MT = magnetic tape

RM = PROM module

`ff` = file number. If this parameter is 0, you can omit it when booting from disc.

`b` = bus address. For disc, this is the HP-IB address (0 to 7) of the disc unit that contains the system file.

`u` = unit or head number:

Disc	Unit/Head #
7902	0 - 1
7906	0 - 3
9134A (4-vol)	0 - 3
7908/11/12/14/33/42/46 disc	0
7908/11/12/14/33/42/46 CTD	1
9144A (standalone CTD)	0

`sc` = select code of the interface card for the device that contains the system file.

file = name of the system file or the boot command file (default is /SYSTEM/BOOT.CMD or SYSTEM for a FMGR cartridge.) This is used if you are loading from disc.

Full details on the boot procedure and examples are contained in Appendix H of this manual.

Loading a system from disc involves almost all of the parameters in the ffbusc string. You do not have to be concerned with the file number unless you are loading hardware diagnostic programs for sequential execution. The file number can be defaulted to 0 (no offset).

For the bus address, specify the HP-IB address, in the range of 0 to 7, of the disc drive that contains the system file.

If you are booting from a 7902 disc drive, set the unit/head parameter to 0 to indicate the left drive, 1 to indicate the right drive.

If you are booting from a CS/80 disc (for example, 7908, or 7914), set the unit/head parameter to 0.

If you are using a 7906 drive, set the unit/head parameter to the number of the head that will read the system file.

If you do not specify the ffbusc[file] string, the VCP/loader ROM assumes a default string of 2027 <defaultfile> for booting from disc; this loads the default file from the disc LU that starts at head 0, cylinder 0, and sector 0 of the disc drive that has bus address 2 on the HP-IB connected to select code 27.

The boot extension (BOOTEX) should reside at the beginning of that disc. The VCP/loader ROM loads that file into memory and executes it.

What happens next depends on what file you specified in your boot string. If the file is a type 1 file, BOOTEX assumes it is a system file, loads it into memory, and executes it. If the file is a type 4 file, BOOTEX assumes it is a boot command file and follows its instructions for booting. If BOOTEX cannot find the file you specified, it goes into interactive mode and prompts for boot instructions. To supply the boot instructions interactively, use the same commands contained in the boot command file as described previously.

If you do not specify a file, BOOTEX looks for the file named /SYSTEM/BOOT.CMD, or SYSTEM if the LU is a FMGR cartridge. If the file is a type 1 file, BOOTEX assumes it is a system file, loads it into memory, and executes it. If the file is a type 4 file, BOOTEX assumes it is a boot command file and follows its instructions for booting. If BOOTEX cannot find either of these files, it goes into interactive mode and prompts for boot instructions. After you have entered your boot string, press RETURN to boot your target system. After your target system boots, check to see that it is running. There are no standard tests for this; just execute a few commands to convince yourself that the system is operating as it should.

If your system does not work properly, or if it does not boot at all, you must go back to the host system and start over again. If the error is in the partition structure or in one of the programs, modify your boot command file (if you used one) and try booting the target system again. If the error is more fundamental, regenerate your system using RTAGN and then start the installation process from the beginning.

Checking the Target System for Proper Operation

Once you have finished loading all of the programs for your target system, you are ready for the system final checkout. The following tests may help you to find errors in your system. While these tests are not exhaustive, they do exercise the software and equipment well enough to test for major hardware failures or generation errors.

Use the FTEST program that came on your primary system disc. FTEST functionally tests each peripheral device on your system. If your system includes a laser printer, use LTEST to check out that peripheral. Both tests are described in the RTE-A Primary System Software Installation Manual (part no. 92077-90038).

Use WH to check the status of the programs, DL to list the contents of a file directory, and LI to list a file. List files to all of the output devices on the system. Read files onto disc from all of the input devices on the system.

Use EDIT to create the source file for a simple program. Compile and load the program, and check to make sure that it works properly.

Dump a source file to an output device or file. Confirm that the contents of the original file and the new file are identical. In addition, you should check to see that your application programs yield appropriate results.

In an environment using several systems with different snapshot files, RPL checksum errors may occur. This happens the first time you run a program on a system where the RPL files generated do not match those in the snapshot file used to load the program. Such RPL checksum errors affect your program only if the program uses an instruction not found in the system currently operating. For example, a program linked for an A600 will run on an A900 system, but the converse is not true if the program used instructions such as SIN or VMPY. You may ignore these errors if you are certain that the programs to be run will not be affected. The RPL checksum error occurs only the first time a program is run or RP'd.

In a system using CDS programs (VC+), if you do not correctly match the CDS fault handler module %CDSFH with a CDS RPL file, you will get either undefined or duplicate symbols. If you use a non-CDS RPL file with %CDSFH, undefined symbol errors will occur. If you omit %CDSFH, duplicate entry point errors will occur.

Identify any problems; then regenerate your system to correct those problems and start the software installation process again.

Setting Up the Startup Program & WELCOME File

During the boot process you were allowed to designate a program as the system startup program. The startup program is scheduled for execution immediately after boot.

The startup program can be any program: an application program, CI, or a user-written startup program. For a disc-based system you should designate CI as the startup program. As the startup program, CI will execute a transfer file named WELCOME<xx>.CMD::SYSTEM. The <xx> part of the file name is filled in from the second parameter of the BOOTEX ST command. For instance, ST,,1 will cause CI to transfer to WELCOME1.CMD::SYSTEM. The number can be in the range of 1 to 99. Default (ST without the second parameter) means that no transfer file will be executed.

The transfer file can include commands to schedule other potential startup programs; in addition, it can include commands to set up your system the way you want it, automatically. (If you are not familiar with the use of transfer files, see the explanation in the RTE-A User's Manual.)

CM as a Secondary Program

In a non-multiuser system, CI can be used as a secondary program scheduled on interrupt from a terminal. It will be called CM and it will execute one command and then terminate. To enable CI this way, use the following command:

```
CN,lu,40b,programe,,CM
```

CI must be restored to the programe specified. The parameter CM is passed to the program and instructs it to accept only one command and then exit. The three commas must be included as placeholders. You may wish to RP a different copy of CI as a primary program for each terminal, and another as a secondary program. The following sequence in the welcome file will enable primary and secondary programs for two terminals.

```
* RP copies of CI for various terminals
RP,CI.RUN::PROGRAMS,CI1
RP,CI.RUN::PROGRAMS,CI3
RP,CI.RUN::PROGRAMS,CM1
* enable each terminal for the right CI
CN,1,20B,CI1
CN,1,40B,CM1,,CM
CN,3,20B,CI3
CN,3,40B,CM3,,CM
```

Note that this applies only to non-multiuser systems. In a multiuser system, PROMT should be the primary program and there should not be any secondary program. PROMT will schedule CM as required in a multiuser system. Also note that when CI completes the welcome file, its ID segment will be released. The primary and secondary program must be RP'd to be scheduled. Thus you do not want the same copy of CI to execute the welcome file and be the primary or secondary program.

Backing Up the Target System

Once your system has been installed and is operating correctly, copy the system, snapshot, boot control files and the type 6 program files for backup onto a removable disc or magnetic tape. This will allow you to recover in the event of a disc hardware failure or an accidental purging of your system. The following methods can be used for system backup.

The TF (or FC) utility will not create bootable system files; these should not be used for system backup.

The ASAVE utility can be used on-line to save any disc LU containing the above system files onto magnetic tape or CS/80 CTD. This LU can then be restored using an off-line memory-based RTE-A system containing the ARSTR utility. Refer to the RTE-A Utilities Manual for more information on ASAVE and ARSTR.

For CS/80 discs with CTD (7908, 7911, 7912, 7914, 7942, and 7946), you can alternately use the off-line save procedure (the pushbutton save procedure) for system backup. This is a built-in off-line backup of an entire disc onto a CTD tape cartridge. A verify program, PBV, is provided for verifying the off-line backup/restore. The procedure is described in the Primary System Installation Manual and the RTE-A Utilities Manual.

The COPYL utility copies an entire disc LU to another LU on a similar disc. Both source and destination LUs must have the same number of tracks and sectors per track. Refer to the RTE-A Utilities Manual for more information on COPYL.

All or selected files residing on a mounted disc cartridge can be copied to another mounted disc cartridge using the CI copy (CO) command. It allows files to be backed up between unlike discs (different sectors per track or different number of tracks). For example, with the copy (CO) command, files on a 7906 disc can be backed up onto a floppy disc.

Be sure to back up your master disc or primary disc also, using one of these methods.

Setting Switches for Autoboot

You can set the processor card of the computer for automatic boot, autoboot, whenever the computer is powered up. Autoboot is enabled by setting switches 1, 2, and 3 on the left side of the processor card. The various settings are listed in Table 9-1.

Table 9-1. Processor Card Switch Settings

Card Switches*			Computer Action
1	2	3	
1	0	0	Transfer control to the VCP. This is a common setting if you do not want automatic boot.
1	0	1	Load the system into memory from the PROM card. The PROM card must have a select code of 22B.
1	1	0	Load the system into memory over the DS link. The default file number (0) and the default select code (24) will be used.
1	1	1	Load the system into memory from disc. The system is loaded from the BOOTEX default file on the disc LU that begins at head 0, cylinder 0, and sector 0 of the disc drive. The boot extension program (BOOTEX) must be located at the start of that disc LU. The disc drive must have an HP-IB address of 2, and the HP-IB card must have a select code of 27.

* 1 = UP = OPEN; 0 = DOWN = CLOSED

These settings are further affected by the setting of switch 8, the auto-restart switch. Set this switch UP to enable the auto-restart feature, DOWN to disable it. If auto-restart is enabled and memory is saved by a battery backup unit, the computer will continue where it left off when power failed. Otherwise, the computer will boot or go into VCP mode, according to the settings of switches 1, 2, and 3.

Installing Optional Subsystems

For installation of subsystems such as IMAGE or the NS/1000 and DS/1000-IV Distributed System (DS) networks included in your system, refer to the appropriate subsystem manuals. If your system has the optional NS/1000 or DS/1000-IV, proceed to the DS Transparency Software Installation given below.

For installation of the optional PC file transfer application software (AdvanceLink), refer to the AdvanceLink manual. The monitor program must be linked and placed in the /PROGRAMS directory in RTE-A. An HP 12040D or HP 12100A MUX is required for AdvanceLink.

It is recommended that a short timeout be set for the PC and that the Monitor program not be aborted with the OF command. The Monitor program will recover within the timeout limits set. A BR command from another terminal will force Monitor to terminate within one timeout period.

In case Monitor hangs up, you may set timeout value to zero "0" with TO, reconfigure the Mux port, and enable program scheduling.

Installing DS Transparency Software

The DS transparency software allows access to files on other RTE-A systems connected via the NS/1000 or DS/1000-IV network. Installation of the DS network is described in the NS/1000 and DS/1000-IV manuals. The DS transparency software installation follows the NS/DS network installation.

Two monitors, DSRTR and TRFAS, are used by the NS/DS transparency software. DSRTR is called when the system is the host system originating the access request, and TRFAS is called when the system is the remote system receiving the access request. Both monitors are required at all systems using the NS/DS transparency software for remote file access.

Install the DS transparency software as follows:

1. Load DSRTR and TRFAS for your system, using the supplied load files and relocatables. Your system must have the NS/1000 or DS/1000-IV labeled common area.
2. Restore (RP) the ID segments for DSRTR and TRFAS when you RP the NS/DS programs (before initializing the multiuser system).
3. Make sure that TRFAS is scheduled by DINIT. This can be done by specifying the monitor scheduling as /D, using the default set of monitors. If you list the monitors explicitly, you must include TRFAS. Do not schedule DSRTR from DINIT.
4. Initialize DS by running DINIT. You will be able to check the DS transparency setup by entering "WH AL" after DS is initialized. You should see that DSRTR is RP'd and dormant and that TRFAS is associated with a session called "TRFAS SESSION", with TRFAS waiting on a class number.

You can test the DS file access setup by entering a CI command such as:

```
LI welcome1.cmd::system>27
```

where the local system is node 27. This should list the file, although the following message will also be displayed:

```
DSRTR: no such file NODENAMES::SYSTEM
```

This file is used to associate node names with node numbers. DSRTR looks for it the first time it is scheduled. If it finds such a file, it will read it to build a table of names for node numbers.

The NODENAMES file should contain entries of the form:

```
* <comment>  
or  
node# nodename [comment]
```

As an example:

```
*Test System 1
1  SYS1
*Test System 2
2  SYS2
*Central Systems
3  Central 1  central system 1
4  Central 2  central system 2
```

Numbers must be separated from names by one or more spaces. Comments are optional and names (up to 16 characters are allowed) must conform to the file naming conventions.

In the multiuser environment, several more user entries should be added during generation, as sessions are created to handle file names with accounts specified.

When DSRTR is first scheduled after boot-up, the NODENAMES file is read into memory. If the file is modified, DSRTR will need to re-read the file. Scheduling DSRTR with no parameters causes DSRTR to re-read the NODENAMES file:

```
RU,DSRTR
```

The system manager may define a default logon account for the DS transparency software to use when a request comes from another node without an explicitly specified account name. When such a request arrives, all of the file accesses are done under the environment defined for this default account, including capability level and LU restrictions. If no default account is defined, the default "TRFAS SESSION" is used. The default logon is defined by running DSRTR in the following manner:

```
RU,DSRTR,ID,logon-name
```

The logon name specified must include the password, if there is one. To eliminate the default logon name, the logon-name parameter must be left off (i.e.,RU,DSRTR,ID). The logon name specified need not exist when DSRTR is run (a warning will be issued). But if it does not exist when a request arrives, the request will fail ("no such account" error).

The default logon name must be set every time DS is initialized. It should be done in the welcome file at bootup. The first time a remote request arrives (scheduling TRFAS), a session is created for this default logon and this session will continue to be used until the session is killed (via GRUMP) or DS is re-initialized.

DS transparency software uses the same amount of SAM as other DS functions. You will have problems if you only have 1k words of SAM, so you should allocate about 4k words of SAM. Remote file access through the DS transparency software can be accomplished through nodes that do not have TRFAS or DSRTR as long as the monitors are set up at the source and destination systems.

Remote file access performance will probably be about a factor of four slower than local access, depending on link speed and network topology.

Installing a Memory-Based System

The process of building and installing a memory-based RTE-A system can be summarized as:

1. Assemble the required system and snapshot files for the target system, and the type 6 program files.
2. Merge the system file and the program files with the BUILD program.
3. Install the merged system file on a bootable medium.
4. Boot the target system.
5. Test the target system.
6. Back up the target system.
7. Optionally, set switches for automatic boot.

Assembling the Required Files

The files required are the system and snapshot files for your target system and the desired type 6 (executable) files. The system and snapshot files are produced by the generator. You must know what directory (or cartridge if on a FMGR disc cartridge) they are on.

The objective in assembling the type 6 program files is to have available a type 6 file for each program that will be part of your memory-based system. There are two ways you can get these files: use the existing type 6 files that were loaded on a similar system, or create them by running the loader on relocatable program files. (Refer to the LINK User's Manual for the definition of similar systems; the essential requirement is that the system common areas be identical in the two systems, and the same type of processor in some cases.)

It is much easier to use existing type 6 files. Most type 6 files loaded for Rev 5000 of RTE-A are transportable, and can be run on other Rev 5000 systems. (Programs loaded on earlier revisions cannot be transported to Rev 5000 systems.) If a program is not transportable, the BUILD program will prevent it from being merged into the system. Further details on what makes a program non-transportable can be found in the RTE-A System Design manual,

The BUILD program cannot load a program from relocatables. If there are some programs for which you do not have type 6 program files, or which were not loaded on a similar system, run LINK on the relocatable code for each of those programs to get the type 6 files. (Be sure to specify the snapshot file of the target system, not the host system.)

As soon as you have your system file, your snapshot file, and all of your type 6 files available, you are ready to run the BUILD program to create the merged system file for the target system.

Run BUILD to Create the Merged System File

The BUILD program combines the system file with type 6 files for all programs that will run in the target system. The result is a self-contained system with the necessary tools for all tasks. Remember that programs cannot be run from a program file on the disc in a memory-based system, thus any programs that are required must be included in the target system.

The following is a brief description of the BUILD program. Refer to Appendix I for further details.

BUILD operates in an RTE-A or RTE-6/VM host system. It can take its commands interactively or from a command file. The BUILD runstring is:

```
RU,BUILD[,command[,list[,output[,snap[,system[,op]]]]]]
```

where

command = file from which commands are taken. If this parameter is omitted, BUILD will default to interactive mode, prompting for commands at your terminal.

list = list file. Default is your terminal.

output = output file. The merged system will be output to this file. If this parameter is omitted, BUILD will prompt for the name. If the default size is too small, it can be changed by specifying a file size in the output file name.

snap = snapshot file produced by the generator.

system = system file produced by the generator, or the output file of a previous run of BUILD. BUILD does not modify this file.

op = optional parameter defining exit path on error when input is from a command file. Options are:

/A = abort and purge the output file.

/E = end and save the output file.

/C = continue processing commands.

Default path is /A, except that an unexpected EOF error on the command file causes a /E exit.

If the system and snapshot file parameters are omitted, you will be prompted for them. If the files do not exist, an error message will be displayed, and then you will be again prompted to enter the names.

Operational Phases

The BUILD program has three distinct phases:

- *Phase 1.* In this phase BUILD reads the command file or, if interactive, prompts you for any vital information that was not included in the runstring (output file, snapshot file, or system file). As soon as BUILD has this information it proceeds to the second phase.
- *Phase 2.* This phase allows you to specify the memory layout for your target system. BUILD asks first for the memory size of the system (in pages), and then asks if you want automatic reserved partition definition. If you answer YES, BUILD will automatically create reserved partitions of the proper size for each program that you merge with the system. If you answer NO, you will have to specify the reserved partitions you want in your system. You may also place programs into partitions manually during phase 3.

NOTE

If you wish to load programs over a DS link using the APLDR program, you must use manual partition definition.

If you ask for automatic reserved partition definition, BUILD proceeds directly to phase 3. If you ask for manual reserved partition definition, you must next define the partitions by entering the size (in pages) of each one. BUILD goes on to phase 3 when you end partition definition by entering /C, by defining partitions that take up all of memory, or by using up all of the memory descriptors (as defined during system generation). (NOTE: Because of the nature of a memory-based system, only reserved partitions are allowed; dynamic partitions can be used only in a disc-based system.)

- *Phase 3.* During this phase you can merge programs into the system. You can also reset the system security code, using the command

SS,newcode

The command

PT

displays the partition table so you can see which programs are in which partitions and how many pages they take up.

Use the RP command to load programs into the merged system file. A command of the form

RP,<program file>[,newname]

builds an ID segment for the specified program and loads it into the merged system file. Following each RP command, you can enter one or more of these subcommands to modify the way the ID segment for the program is constructed:

- PR,n sets the program priority to n. (Default is the value supplied by LINK.)
- SZ,nn sets the program size, including system common, to nn pages. This has the same function as the loader SZ command, but overrides it. For CDS programs (92078 only) the SZ command overrides the LINK heap (HE) command. You can use this command to create buffer space behind your program. (Such space can be made available to your program with a call to the LIMEM subroutine.)
- PA,nn[,c/d] loads the program into partition nn. If you specified manual partition definition, you can enter this subcommand for each program in the system. (If you do not specify a partition, BUILD loads the program into the smallest unused partition that will hold it.) This command is invalid if you specified automatic partition definition. For a CDS program, use the optional 'c/d' parameter to specify whether to place the code or data in partition nn. Use 'c' for code, 'd' for data. Default is data.
- ST[,p1...[,p5]]] makes this program the startup program, and optionally passes up to five RMPAR parameters to it. There can be only one startup program in the system, so if you specify more than one startup program only the last one is used.

After you have merged all of your program files with the system file, enter

/E

to signal the end of input to BUILD. The output file now contains your complete target system.

Sample Command File

The sample command file shown below contains typical commands used with the BUILD program. The commands are the same whether you run BUILD interactively or use a command file. To use a command file, just specify the file name as the command name in the BUILD runstring. To run BUILD interactively, specify the LU number of your terminal as the command name or omit the command from the runstring.

Comments can be placed in the command file but they must follow the last parameter. If a comment is included among the parameters, BUILD will try to parse the comment as a parameter. Use commas to supply null values for those optional parameters that you do not want to specify.

The runstring for the sample command file is

```
RU,BUILD,CMND,1,output,snap,system
```



The file CMND contains these commands:

NO,	no automatic partition definition
256,	memory size in pages
5,	size of first partition
32,	size of second partition
10,	size of third partition
13,	size of fourth partition
28,	size of fifth partition
32,	size of sixth partition
/C,	end of partition definition
RP,RUNL,,	merge RUNL into the system
PA,4,	load RUNL into partition 4
RP,COMND,,	merge COMND into the system
PA,1,	put COMND into partition 1
RP,DRTR,D.RTR,	merge D.RTR into the system
SZ,32,	size D.RTR to 32 pages
PA,2,	put D.RTR into partition 2
RP,WH,,	merge WH into the system
SZ,15,	size WH to 15 pages
PA,5,	put WH into partition 5
RP,APPLN,,	merge APPLN into the system
PA,6,	put APPLN into partition 6
ST,ST,FILE,,	make APPLN the startup program
/E,	complete the build process

After running the above command file, you now have a merged system file. The memory map for the system produced by our example command file is shown in Figure 10-1.

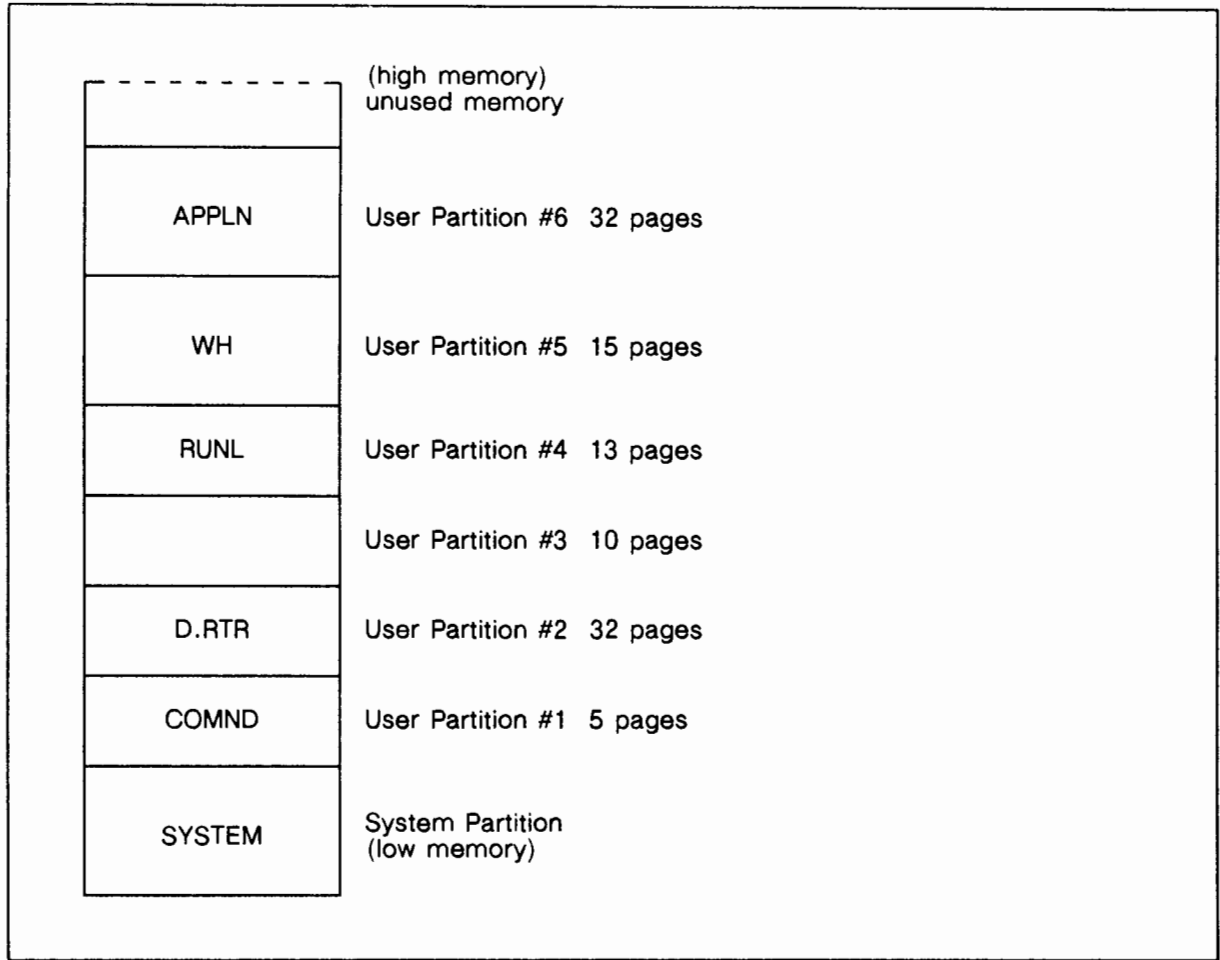


Figure 10-1. Memory Map of Sample Memory-Based System

Installing Merged System File on a Bootable Medium

The merged system file must be booted from the proper medium. The boot procedure is different for different media, as described in the following subsections.

PROM

If you plan to boot your memory-based system from a PROM module, the target system must be placed at the beginning of the PROM module. You can use a merged system file created by BUILD for your target system. You must burn the target system file onto the PROMs and then install the PROMs on one or more PROM modules. The method for burning the target system file onto the PROMs depends on what PROM burner you use; see the operating manual for your PROM burner.

CTD

To be bootable from the CTD tape, the system file can be stored on any file on the tape. Use the CSYS utility (described in the RTE-A Utilities Manual) to install the system file on the tape.

Magnetic Tape

To be bootable from magnetic tape, store the system file onto the tape as binary code. Use the CI operator command CO to copy the system file onto magnetic tape. Refer to the CI User's Manual for details.

Disc Volume

To be directly bootable from the disc, the system must be on an LU that starts at physical cylinder 0 and sector 0 of the disc drive. For a file system volume, free space must be reserved at the beginning of the disc for the bootable system files. Use the CI IN command to reserve space, as

```
IN,lu,blocks
```

The blocks parameter specifies the amount of space, in 128-word blocks, to be reserved by the file system for bootable systems.

Free space is defined in segments of 256 blocks.

If the system begins on cylinder 0, sector 0 of the disc, it is called bootable file 0. If it begins 256 blocks (128 words per block) further in on the disc it is called bootable file 1, starting at 512 blocks is bootable file 2, etc.

The system is placed in the reserved space using the FPUT utility:

```
RU,FPUT,file,LU,<bootable file#>
```

where:

file is the name of the file which needs to be installed.

LU is the LU of the destination disc.

bootable file# is the bootable block number. Bootable file 0 starts at block 0, bootable file 1 starts at block 256, bootable file 2 starts at block 512 and so on.

Refer to the RTE-A Utilities Manual for further discussion of FPUT.

FPUT will check that the space has been reserved, then write the contents of the file in the specified place on the disc. Previous data in that location will be replaced without warning. This file is now bootable using the boot commands described in the next section.

Directly from FMGR Cartridge

To be directly bootable from the disc, the system must be on an LU that starts at physical cylinder 0 and sector 0 of the disc drive. If the system begins on cylinder 0, sector 0 of the disc, it is called bootable file 0. If it begins 256 blocks (128 words per block) further in on the disc it is called bootable file 1, starting at 512 blocks is bootable file 2, etc. To install the system at bootable file 0 on a FMGR cartridge, the system must be the first file on the cartridge. To install the system at bootable file 1, the sum of the sizes of all previous files on the cartridge must be exactly 256 blocks, bootable file 2 will have exactly 512 blocks of files before it and so on. This is most easily guaranteed by having files of 256 or 512 blocks at the beginning of the cartridge which contain all of the systems that must be directly bootable from that cartridge.

From Disc, Using BOOTEX

BOOTEX is an HP supplied memory-based system used to boot in other systems. To use BOOTEX, it must have been previously installed. If it has not been installed, use INSTL, described in the Utilities Manual, to install BOOTEX. (No installation setup is required to boot over a DS link.)

Using BOOTEX, you can boot your memory-based system from any file on the same disc LU as BOOTEX. It is not necessary to place the file at a special location. BOOTEX will use the file name supplied at boot to locate the file. Note: the system must be on the same LU as BOOTEX. It does not matter if that LU is a FMGR cartridge or a file system volume.

Booting the Target System

Now that your merged system file is installed in the proper format at a bootable location, you can boot it up. The general boot procedure is:

1. At the terminal that is set up as the virtual control panel, press the BREAK key to enter VCP mode. The VCP will prompt for input by displaying the contents of all of the registers, followed by the VCP prompt "VCP>".
2. Enter a boot command of the form

`%Bdd[ffbusc[file[,SS]]]`

The VCP accepts uppercase or lowercase characters. Each character of the *ffbusc* string is an octal digit. Leading zeros in the *ffbusc* string can be omitted; blanks are not allowed in the string. If you make an error entering the bootstring characters, BACK SPACE to the character in error, correct the error, re-enter the remainder of the string then press RETURN.

where:

`%B` = identifies the string as a boot command (load a system and execute it).

`dd` = device from which you are booting:

RM = PROM module

DC = disc drive or Cartridge Tape Drive (CTD) via HP-IB

DI = 248x integrated disc controller

DS = disc drive via DS/1000-IV

MT = magnetic tape drive via HP-IB

`ff` = file number: not required for `%BRM`.

For `%BDS` (boot over DS), the *ff* parameter must not exceed 77777.

For `%BDC` (boot from disc), this is the bootable file number. This is the same number given as the third parameter to FPUT if the system is on a new file system volume. Bootable file 0 starts at cylinder 0, track 0, sector 0. Bootable file 1 starts at block 256, bootable file 2 starts at block 512 and so on. This is generally not used when you are booting using BOOTEX.

`b` = bus address. For disc, this is the HP-IB address (0 to 7) of the disc unit that contains the system file. For other devices, including DS, the VCP ignores the *b* parameter, but requires a 0 as a place holder if the *ff* parameter is not zero. On DS, the destination VCP needs no information about the location of the system file on the source system.

u = unit or head number:

Disc	Unit/Head #
7902	0 - 1
7906	0 - 3
7908/11/12/14/33/42/46 (Disc)	0
7908/11/12/13/33/42/46 (CTD)	1
9144 Standalone CTD	0

sc = select code of the interface card for the device that contains the system file.

file = name of the merged system file. This parameter is only used by BOOTEX, and specifies the name of the system file or a BOOTEX command file. If you have installed your system to be booted by BOOTEX, put the full name of the file here. The default name is /SYSTEM/BOOT.COMD if BOOTEX is installed on a file system volume, SYSTEM if BOOTEX is installed on a FMGR cartridge. Omit when you load via DS, CTD, CTU or PROM.

SS = load and halt. If you append SS to the system file name, the system file will be loaded and the computer will stop with a VCP prompt (VCP>). A %R or %E in response to the VCP prompt will start execution.

Full details on the boot procedure are contained in Appendix H. Note that there are minor differences in the boot procedure for the different devices. These are:

- *PROM module.* The VCP/loader ROM will load the system directly from the select code specified. File number, bus address, and unit number are not needed for the PROM module. If you do not specify the select code (*ffbusc* defaulted), the VCP will use a default of 22 (load from the PROM module connected to select code 22).

- *Disc.* Loading a memory-based system from disc uses the *ff* parameter as the bootable file number if you are booting directly, without using BOOTEX.

For the bus address (b), specify the HP-IB address, in the range of 0 to 7, of the disc drive that contains the system file.

If you are booting from a 7902 disc drive, set the unit parameter (u) to 0 to indicate the left drive, 1 to indicate the right drive.

If you are using a 7906 drive, set the unit parameter to the number of the head that will read the system file (0 - 3).

If you are booting from a CS/80 disc, set the unit parameter to 0 for disc.

If you are booting using BOOTEX, the file parameter specifies the name of your system file.

If you do not specify the *ffbusc* string, the VCP/loader ROM assumes a default string of 2027 for booting from disc. This loads bootable file 0 from the disc LU at bus address 2, head or unit number 0 on select code 27.

- *CTD Tape.* When you load your system from a CS/80 Cartridge Tape Drive (CTD), the file number and unit number must be entered. The file number specifies the absolute position of the file on the tape (as specified in the CSYS program). The unit number for the CTD is 1. For the standalone 9144 CTD, the unit number is 0.
- *Magnetic Tape.* When you load your system from magnetic tape, the loader ROM will rewind the tape and then position the tape to the file specified. Default is the current tape position (*ff* = 0). The default *busc* string is 4027.
- *DS Link.* When you load your system over a DS link, the program PROGL, running on the neighbor computer system, translates the file number (*ff*) into a file descriptor. The default descriptor is Pnnnnn:0:0, where nnnnn is *ff*, a maximum of five octal digits. If you default the *ff* parameter in the boot string, the default file name is P00000:0:0.

If file name Pnnnnn:0:0 is not satisfactory for your purpose, you can supply your own translation routine when you load PROGL at the host system. To do this, write a subroutine and load it with PROGL under the name #DNFL. The calling sequence in PROGL is:

```

JSB #DNFL           ;the translation subroutine must be named #DNFL.
DEF *+7            ;pass first two parameters, return last four.
DEF <linklu>       ;LU number of link requesting download.
DEF <nnnnn>        ;file number (ff) from destination boot string.
DEF <filnam>       ;six-character name of merged system file.
DEF <secode>       ;integer security code of merged system file.
DEF <crn>          ;integer cartridge reference number.
DEF <node>         ;integer node number of source system.

```

Only the store-and-forward version of PROGL will recognize the <node> parameter.

If you use the default translation Pnnnnn:0:0, then do not provide translation subroutine #DNFL.

The select code (*sc*) in the boot string must be the select code on the destination system of the DS link between the destination computer and the neighbor computer running PROGL. The default is 24.

Boot-string parameter *b* for DS is zero; therefore, if you can default the file number to zero (source file P00000:secode:crn) and the select code to 24, the boot command becomes simply:

```
%BDS
```

After you have entered your boot string, press RETURN to boot your system.

Testing the Target System

After your target system boots, check to see that it is working properly. There are no standard tests for a memory-based system, so the best test is probably to check the application programs in your system to see that they give you the proper results.

If your system does not work properly, or if it does not boot, go back to the host system and start over again. If the error is in the partition structure or in one of the programs, use `BUILD` to rebuild the merged system file and continue the installation process from that point. If the error is more fundamental, regenerate your system using `RTAGN` and then start the installation process from the beginning.

Backing Up the Target System

Once you are satisfied that your target system is working properly, make a backup copy of it. There is more than one way to do this. You can use the `TF` utility to copy your files to a CTD, mag tape, or removable disc, or you can use the `CO` command to make the copy on a removable disc. You can also use the `ASAVE` utility to back up one or more disc LUs to magnetic tape or CTD. If you have a CS/80 disc with CTD, you can use the off-line backup procedure to back up the whole disc. Refer to the RTE-A Utilities Manual for details.

As long as your system, including common, is 32 pages or less, the destination VCP can copy your system over a DS link to a neighbor computer through a dialog with DS/1000-IV programs `DSVCP` and `VCPMN` running on the neighbor computer. The resulting copy can then be used for a DS boot. If the destination system itself includes DS/1000-IV, then you can transfer a copy of the destination system file to another node on the DS network by using the `REMAT ST` command, regardless of system size.

If you use the `CO` command or the `TF/FC` utility to back up your system, make sure you have a copy of the `BOOTEX` file in your target system. You must rename this file so it can be copied by `TF/FC`.

Setting Switches for Automatic Boot

You can set the processor card of the computer for automatic boot whenever the computer is powered up. Automatic boot is enabled by setting switches 1, 2, and 3 on the left side of the processor card to the proper values. The switch settings are shown in Table 10-1.

Table 10-1. Processor Card Switch Settings

Card Switches*			Computer Action
1	2	3	
1	0	0	Transfer control to the VCP. This is a common setting if you do not want automatic boot.
1	0	1	Load the system into memory from the PROM card. The PROM card must have a select code of 22.
1	1	0	Load the system into memory over the DS link. The default file number (0) and the default select code (24) will be used.
1	1	1	Load the system into memory from disc. The system is loaded from bootable file 0 of the disc LU at HP-IB address 2, unit or head 0, select code 27. If this file is BOOTEX, it will load either /SYSTEM/BOOT.COMD if it is a file system cartridge or SYSTEM if it is a FMGR cartridge. If BOOTEX finds this file to be a boot command file, the commands in it will be executed.
* 1 = UP = OPEN; 0 = DOWN = CLOSED			

These settings are further affected by the setting of switch 8, the auto-restart switch. Set this switch UP to enable the auto-restart feature, DOWN to disable it.

NOTE

Do not enable auto-restart unless battery backup is installed. To do so would produce unpredictable results in the event of a momentary power failure.

If auto-restart is enabled and memory is saved by a battery backup unit, the computer will continue where it left off when power failed. Otherwise, the computer will boot or go into VCP mode, according to the settings of switches 1, 2, and 3.

Multiuser and Spooling Setup

This chapter is applicable only for systems with the HP 92078A Virtual Code+ (VC+) option, and assumes that the RTE-A system is booted and running. Throughout this chapter, the Primary System disc I/O configuration is used for demonstration purposes.

Group and User Management Program (GRUMP)

The Group and User Management Program, GRUMP, is used to establish and maintain the multiuser account system. The first time GRUMP is run, it does the following:

1. prompts for the LU of the disc where the global directory USERS is to be placed,
2. creates directories /USERS and /USERS/HELP,
3. creates the MASTERACCOUNT file,
4. prompts for the logon prompt,
5. creates the LOGONPROMPT and MASTERGROUP files, and
6. creates group configuration files SYSTEM.GRP and NOGROUP.GRP (for mandatory groups SYSTEM and NOGROUP), and creates the user configuration file MANAGER (for mandatory user MANAGER).

MANAGER is given the following permanent attributes:

1. capability level of 31,
2. superuser bit set,
3. all entries in the LU access table set, and
4. membership in groups SYSTEM and NOGROUP.

The operator can then modify the unique user attributes of MANAGER that can be changed: real name, password, number and depth of USDPs, and the default logon group. The following example illustrates a first run of the program GRUMP on a new RTE-A system:

CI> GRUMP

Initializing multiuser files.

Which LU should ::USERS directory go on? [0] 16

MASTERACCOUNT::USERS created and initialized.

Enter logon prompt [Please log in]: <cr>

LOGONPROMPT::USERS created and initialized.

MASTERGROUP::USERS created and initialized.

SYSTEM.GRP::USERS created and initialized.

NOGROUP.GRP::USERS created and initialized.

MANAGER::USERS created and initialized.

Initialization of multiuser environment complete.

Do you want to modify MANAGER (Yes/No) [N]: Y

Logon Name of MANAGER can never be modified.

Enter users real name [???] : James T. Kirk

Enter password (a <cr> gives no password) : Admiral

Capability of Manager can never be changed.

LU access table of MANAGER can never be modified.

Enter UDSPs:depth [0:0] : <cr>

Modified unique user information.

Do you wish to include the user in an existing group
other than NOGROUP (Yes/No) [N] : N

Which group should be the default logon group [SYSTEM]: <cr>

GRUMP> ex

NOTES:

1. Specifying LU 0 for the directory /USERS allows the file system to arbitrarily choose the LU to place /USER on.
2. In this example, the operator said yes to altering the unique user attributes of MANAGER. The user could also have said no and done the modifications later.
3. The operator can only modify certain attributes, but is notified of those that cannot be changed.
4. The LU access table referred to in the message "LU access table of MANAGER can never be changed." is an LU access table which is used to control access to disc LUs. Details of the LU access table are in the RTE-A System Manager's Manual.
5. The prompt "Enter UDSPs:depth [0:0] :" refers to the User-Definable Directory Search Path capability that allows you to define a specific path for searching for a file. Details of the UDSPs are in the RTE-A User's Manual.
6. Since there are only two existing groups on the system at this time and MANAGER is in both of them, you should answer NO to the prompt "Do you wish to include the user in any existing group other than NOGROUP".
7. Since MANAGER is only in groups SYSTEM and NOGROUP at this time, you should leave SYSTEM as the default logon group. NOGROUP is the only alternative to SYSTEM as the default logon group and this is not advised because NOGROUP is only intended for use in systems not using groups.
8. SYSTEM is made the default logon group for MANAGER, and /SYSTEM is made the default working directory for MANAGER.SYSTEM, but the directory /SYSTEM is not created. If these are not appropriate, change them with the ALTER USER command in GRUMP. To change the default logon group, enter "MANAGER." for the user.group parameter in the ALTER USER command. To change the start-up command, default logon directory, logoff program/command file, or any other group-dependent user information for MANAGER, run the ALTER USER command, specifying the corresponding MANAGER.groupname.

Note that if you leave /SYSTEM as the default working directory, you must create the actual directory. This can be done using the ALTER USER command in GRUMP with MANAGER.SYSTEM as the user.group parameter. GRUMP asks for the default working directory with /SYSTEM as the default. If you enter a carriage return, GRUMP prompts to see if you want to create the directory, and for the disc LU on which it should be placed if you say to create it.

At this point, directories /USERS and /USERS/HELP have been created and /USERS contains six files: LOGONPROMPT, MANAGER, MASTERACCOUNT, MASTERSGROUP, SYSTEM.GRP and NOGROUP.GRP. PROMT should now be enabled as the primary program at the terminal, and you should log on as MANAGER at that terminal.

The enabling command (for LU 1) is:

```
CI> cn 1 20b PROMT
```

PROMT may be enabled as the primary program for the terminal LU at generation time. This is encouraged, to avoid inadvertently overlooking this. To log on, enter the following:

```
Please log on: MANAGER
Password? <cr>
CI>
```

Any needed directories can be created as follows:

```
CI> crdir /SYSTEM 16
CI> crdir /SYSTEM/NOTES
```

The first command creates the global directory SYSTEM on LU 16. The LU parameter is optional; if it is not supplied, the directory will be placed on the first available LU. The second command creates the subdirectory NOTES in the global directory SYSTEM. Since the global directory is already located on LU 16, the LU parameter in the second command is not needed. If an LU is specified, it will be ignored. The second command requires that the global directory /SYSTEM exists.

Ownership of directories can be displayed and altered as follows:

```
CI> owner /SYSTEM
owner of /SYSTEM is MANAGER          MANAGER is the owner and
                                       associated group has not been
                                       assigned to directory /SYSTEM.

CI> owner /MANAGER
owner of /MANAGER is MANAGER.SYSTEM  MANAGER is the owner and the
                                       group SYSTEM is the associated
                                       group of the directory /MANAGER.

CI> owner /SYSTEM MANAGER             Assigns MANAGER as the owner of
                                       /SYSTEM and the default logon group of
                                       group of MANAGER as the associated group.

CI> owner /SYSTEM MANAGER.CONTROL     Assigns MANAGER as the owner of
                                       /SYSTEM and the group CONTROL as the
                                       associated group.
```

The owner/group/other protection (described in the RTE-A User's Manual) may be used in connection with ownership of these directories, to allow all users to access them, but not change them. For example, global directory SYSTEM can be accessed but not changed by users other than the owner if the following command is given:

```
CI> prot /SYSTEM rw/r/r
```

Global directory HELP may be freely accessed (allowing both creation and purging of files) if the following command is given:

```
CI> prot /HELP rw/rw/rw
```

Note that you must be the owner of the entity, a superuser, or a user with high enough capability to issue the protection command. For more information on the protection command, refer to the RTE-A User's Manual (part no. 92077-90002).

GRUMP help files should be copied to /USERS/HELP with the HP-supplied transfer file GRUMPHLP.CMD. GRUMP may then be run to create other user and group accounts. This can only be done by the system manager, a superuser or a user with high enough capability. For more information on GRUMP, refer to the RTE-A System Manager's Manual (part no. 92077-90056).



Establishing Multiuser Accounts

There are two types of accounts in the multiuser environment: group accounts and user accounts. A group is a set of users who share common functions, applications, and/or resources. Group accounts are used to assign selected resources to a specific set of users. User accounts provide the system with the information necessary to set up and maintain the operating environment for each user in the system.

If Security/1000 is off or not installed, the user accounts are divided into two types: standard and superuser. Superusers have greater capabilities in commands available to them; this is detailed in the RTE-A User's Manual (part no. 92077-90002). If Security/1000 is on, the commands available to users are filtered by capability levels; this is detailed in the RTE-A System Manager's Manual (part no. 92077-90056).

The group and user accounts are managed in an account system by the Group and User Management Program, GRUMP. The account system requires a global directory USERS, which contains the following files: GROUPACCOUNT, LOGONPROMPT, MANAGER, MASTERACCOUNT, NOGROUP.GRP and SYSTEM.GRP, as well as a subdirectory HELP for the GRUMP help files. All of these are created automatically by GRUMP the first time it is run. Each user and group also has an account file residing on the directory /USERS. These files are called group and user configuration files respectively. Files on /USERS should only be modified by GRUMP or with the HP-supplied multiuser account utilities.

PROMT Program

This program facilitates prompting from the terminal and access to the user account system. It must be RP'd and enabled as the primary program at every terminal in a multiuser system. Terminals having PROMT enabled as the primary program operate differently from those with PROMT not enabled, therefore all terminals should be enabled to avoid any confusion in system operations.

PROMT takes care of initializing the multiuser system the first time it runs. It will automatically RP LOGON, and CI (calling it CM) if they are not already RP'd. PROMT will search for the following files:

```
LOGON.RUN::PROGRAMS
CI.RUN::PROGRAMS      [note, not CM]
```

PROMT displays the logon prompt, located in the file LOGONPROMPT::USERS created by the program GRUMP the first time it is run. This file may be changed by using EDIT if the prompt needs to be changed. In order to modify this file, PROMT must be OF'd (without the ID parameter) so that PROMT will reread the file. Note that OF,PROMT,ID will make PROMT unavailable, preventing multiusers from logging on and likely requiring a RP command to restore PROMT, or a system reboot.

Once logged on, you can encounter difficulty because of down devices, hung programs, or other problems that cause CI to become unavailable. In these cases, since PROMT is still the primary program enabled at each terminal, it schedules a copy of CI (called CM) instead of displaying the logon message in response to terminal interrupts. CM has almost all the capabilities of CI, but allows only one command to be executed at a time. This is to allow error recovery. Programs run from CM are always run without wait. If both CM and CI are busy, PROMT will display the "SYSTEM>" prompt to allow limited error recovery. Refer to the RTE-A User's Manual for more explanation of exception conditions.

Be sure to specify sufficient SAM, since both PROMT and CM will be affected if the system runs out of SAM. In some cases, PROMT may still display the CM program prompt but there is not enough SAM to contain the command entered. In such cases, the command will be ignored. Refer to Chapter 7 for instructions on allocating SAM.

LOGON Program

LOGON is scheduled by PROMT to validate the user.group name entered. If the name validates, LOGON performs the logon process, allowing the user access to the system. This process initiates a session with the attributes specified in the corresponding user and group configuration files (working directory, startup program, etc.) and runs the startup program for that session.

When a session terminates, if multiuser accounting is enabled, the operating system sends the session's accounting information to LOGON via class I/O. LOGON then processes this data and updates the accounting information saved in the user's configuration file.

Spooling Initialization and Use

The RTE-A spooling system has two major features: outpooling and LU redirection. Outpooling is directing the output to a file and subsequently to an LU; LU redirection is rerouting the output from one LU to a different LU. Output may also be directed to remain in a file and not output to any LU. (This last feature is enabled when you specify an output spool file instead of defaulting to the file provided by SP.) If it does not already exist, the directory SPOOL will be created by the spool system to hold the outpool files and an information file. The following four executable programs (type extension .RUN) must be located on directory PROGRAMS:

```
SP
SMP
SPGET
OUTPT
```

To initialize the spooling system, enter the spool system from CI using the SP command:

```
CI> SP
RTE A Spooling System
Type ? for help
- -
```

Entering a "?" at this point will fill the screen with the available commands.

```
RTE A Spooling System
Type ? for help
-?
```

The Spool System Commands are:

```
QU = shut down the spool system
IN = initialize the spool system
ST = print spool system status
PU = terminate output and purge if a spool file
    Expected format: PU, filename
ON = start spooling to an LU
    Expected format: ON [,LU [,filename [,NC ]]]
    or to commence LU redirection from LUS to LUD
    Expected format: ON , LUS , LUD
OF = stop spooling to an LU
    Expected format: OF [,LU]
LI = list file to LU
    Expected format: LI, filename [,LU [,NC ]]
RE = restart spool output
    Expected format: RE, filename [,-lines]
LO = start/stop error logging
    Expected Format: SP,LO,ON,filename or SP,LO,OF
EX = leave the spool monitor
```

Note: - the default LU is 6; NC means No Carriage Control

The IN command allows initialization of the spooling system. The ST command shows the current status of the system.

- ST

Spool System Not Initialized

- IN

Maximum Spool Files= 24

NOTE - To modify the maximum spool files count, shut the spool system down (SP,QU) and re-size SPGET (SZ,SPGET)

Spool System Initialized

- ST

Redirection Status (your session only):

Spool File Status:

Maximum Spool Files = 24

No Spool Files in Use!

The default size of program SPGET allows for approximately 30 spool files. If SPGET is increased by 1 page, approximately three more files may be queued up on the system. This is recommended for high-use systems. The following example illustrates the processes of shutting down the spooling system, resizing SPGET, and reinitializing. The QU command shuts down the spooling system, and the EX command exits from the SP program back to CI.

- QU

Spool System Shutting Down!

- EX

CI> SZ SPGET

Last Addr= 36441 Min.Part. 22 EMA/WS= 0 Mseg= 0 VMA = 0

CI> SZ SPGET 23

CI> SZ SPGET

Last Addr= 36441 Min.Part. 23 EMA/WS= 0 Mseg= 0 VMA = 0

CI> SP

RTE A Spooling System

Type ? for help

- IN

Maximum Spool Files= 28

NOTE- To modify the maximum spool files, shut the spool system down (SP,QU) and re-size SPGET (SZ,SPGT)

Spool System Initialized

- EX

CI>

On the primary system, LU 25 describes an HP-IB instrument with HP-IB address 1 on an HP-IB card with select code 25. An HP 82905B printer is often connected to this LU as an inexpensive means of obtaining hard copy output. LU 6 has traditionally been the printer LU; to redirect output to LU 25, enter the following commands to the SP program:

```
- ON 6 25
LU redirection started from LU 6 to LU 25
- EX
CI> _
```

Now, if the command "CO BOOT.CMD::SYSTEM 6" is entered at the terminal, the output will be rerouted to the printer at LU 25 instead of LU 6. Output will begin to come out at the printer immediately. To terminate the redirection from LU 6 to 25, enter the following command from SP:

```
- OF 6
LU redirection terminated from LU 6 to LU 25
- _
```

In order to set up a spool file to a particular LU, the ON command is again used. This time no redirection LU is specified and, if no file name is specified, a default output spool file is created by the spool system. Any output to the LU is then stored in the default file until the spool system is again entered and the output is terminated using the OF command. At that point, all of the output is immediately spooled from the default file to the LU.

```
- on 6
Spooling started from LU 6 to OUTSPOOL01.SPL::SPOOL
- ex
```

```
CI> co boot.cmd::system 6
CI> co temporary.file::texts 6
CI> co letternine.txt::texts 6
CI> sp
```

```
RTE A Spooling System
Type ? for help
- st
Spool File Status
Filename = OUTSPOOL01.SPL::SPOOL
Owner   = SCOUT Terminal LU = 1
Spool LU      = 6      File status = Actively spooling
Approximate Line Count =      21
```

```
- of 6
Spooling terminated from LU 6 to OUTSPOOL01.SPL::SPOOL
- st
Spool File Status
```

```
Filename = OUTSPOOL01.SPL::SPOOL
Owner    = SCOUT Terminal LU = 1
Spool LU = 6      File status = Outputting to device
Approximate Line Count = 21
- ex
```

The spool file OUTSPOOL01.SPL::SPOOL can be restarted or purged from the spooling system using the RE or PU commands. If purged, the file itself is removed. If errors are encountered, messages and suggested solutions are displayed from the spool system.

You may specify your own file to replace the default file of

OUTSPOOL01.SPL::SPOOL

in the ON command. In this case, spooling output is queued into the specified file. When the OF <lu> command is then issued, the output does NOT automatically begin at the LU. Only in the case of a default spool file name is output automatically sent from the file to the LU when the OF command is issued. To output the file to the re-directed LU, the spool LI command may be used.

SP Program

This is the main program of the spooling system, and it should be given high priority (default is 80). It is the user interface to the spooling system and is started from CI with the SP command. For more information about the various commands available in SP, refer to the RTE-A User's Manual. The SP ST (status) command shows the current status of the spooling system.

SMP Program

The SMP program is the programmatic interface in the spool system. It handles communications between programs and the spooling system.

SPGET Program

Performs class I/O get calls on the spool system class number where data is waiting to be spooled. This program may be increased in size to allow for more spool files in the system. This program should be run at low priority in order to avoid degrading system performance (default is 100).

OUTPT Program

This program handles the output of a spooled data file to the specified LU. It should be given relatively high priority in order that OUTPT may take advantage of the sophisticated RTE-A system hardware I/O capabilities. The default priority is 80.

Shared CI

With the VC+ option, you can take advantage of shared programs to reduce memory requirements in systems with many users. Without the VC+ option, each copy of CI requires 32 pages, plus 32 pages for CM. With VC+, a shareable version of CI is included, which only requires 20 pages of data per copy, plus one code partition of about 26 pages that is shared by all the copies.

Since the main portion of the Command Interpreter is composed of two programs, CI and CIX, the shareable version can be loaded using the LINK command files #CICDS and #CIXC.

There should not be any noticeable difference between this version of CI and the unshared version, other than the memory requirements. Note that since CI uses data space for file copy and mask operations, you can increase the performance of CI by providing more data space.

CDS Library

A library called \$BGCDS is included with VC+. This library contains the CDS versions of some of the larger routines in \$FMP and \$SYSLB, primarily the non-FMGR FMP calls and multiuser subroutines. You can take advantage of this library if you have a program that makes FMP calls only from CDS code. By using \$BGCDS, you can reduce the amount of data space used for system subroutines.

When loading programs, library \$BGCDS should be searched before \$BIGLB. This will give you the CDS versions of the FMP routines. If you have programs containing any FMP calls in non-CDS code, LINK will abort with an error message. These calls may be from HP-supplied libraries. For example, the FORTRAN I/O library makes FMP calls from non-CDS code. If you have any such calls, do not use \$BGCDS.

Error Messages

This appendix provides the error messages that may occur during the system generation and installation procedures. The BUILD program error messages also are included.

RTAGN Errors

AL:LO not allowed with PSeudo-driver

A map set may not be locked to an IFT associated with a pseudo driver. Remove the AL or the PS command.

Background swap priority error

Ensure that the value is in the range 0 through 32767.

Bad parameter

The parameter is not the correct type or is out of range. Review the command description for a definition of legal parameters.

Base page overflow

Reduce the number of base page links by using current page linking, or by ordering the module differently to reduce cross page references.

Blank common command out of order

Ensure that the command to specify the number of words of blank common, *COM,numWords*, is in the correct format. *numWords* must be in the range 0 through 32767.

Buffer limit error

Ensure that the buffer limit parameter in the DVT command is of the form *BL:BU:ll:ul* or *BL:UN:ll:ul*, that *ll* is less than *ul*, and that both are numerics in the range 0 through 32767.

Can't fix up unidentified external references in partitions 32 through 127

System modules that reference externals in other partitionable modules must be relocated in partitions 1 through 31.

Can't mix drivers and system modules in same partition

System modules and I/O drivers cannot be relocated into the same OS/driver partition.

Checksum error

The file may not be a relocatable file or may be incorrect type. Specify the correct file and/or retry.

Class number specification error

The command to specify class numbers occurred out of order in the command file or was specified incorrectly.

- 1) Check the format of the command: *CLAS,n*.
- 2) Check that it follows the END which terminates system relocation.
- 3) Check that *n* is numeric and in the proper range.

Common usage not allowed

Illegal module for the generator; use of system common is not allowed. This error will occur if the library \$BIGLB is searched during system common relocation.

Corrupt relocatable file

The file specified in an RE, SE or MS command is not a valid relocatable (type-5) file. Ensure that the file is not actually a source file or some other non-relocatable file. The file may be internally corrupt, in which case a new copy of the file must be created or obtained.

Device priority error

Ensure that the device priority specified in the DVT command is of the form *PR,priority* and that *priority* is less than 63 (77B).

Device type error

Ensure that the device type parameter in the DVT command is of the form *DT:deviceType* and that *deviceType* is in the range 0 through 77B.

Driver parameter area size error

Ensure that this parameter is specified before the driver parameters are specified. Ensure that the specification is less than 511.

Driver parameter error

Driver parameter is out of range, or more parameters were specified than are allowed in the DX parameter.

- 1) Check that the parameters are numeric and less than 32767.
- 2) Increase the specification of the driver parameter extension with the DX command.

Duplicate entry points

Two subroutine entry point names are the same. Remove, rename, or recode the duplicate module.

DVT command or parameter error

The command to specify DVTs occurred out of order in the command file, or was specified incorrectly.

- 1) Check the format of the command: *DVT,....*
- 2) Check that it follows an IFT or DVT command.
- 3) Check the format of the command parameters.

EMA usage not allowed

Any reference to EMA is illegal. Remove the offending module, or remove the offending reference in the module.

This error is also reported if the generator encounters ALLOC or DEBUG type relocatable records. (ALLOC records describe common blocks.) Use \$ALIAS NOALLOCATE directive for FORTRAN programs to have an alternate record created.

ENT record follows DBL record in module

A module relocated in an OS partition does not meet the requirement that all ENT records must precede all DBL records in the relocatable file. This module was not produced by an HP-supported language processor or is corrupt. This module can only be relocated in the unpartitioned part of system relocation.

Exceeded available capacity of system file**Specify larger system file size**

A system file that is to be booted must be contiguous; that is, it cannot have file extents. Resize the system file.

Expected "Node" command

The command to specify the node lists occurred out of order in the command file, or was specified incorrectly; an LU in the list was already in another node list.

- 1) Check the format of the command: *NODE,lu,lu...*
- 2) Check that they follow the two ENDS that terminate the IFT and DVT generation.

Fixup table overflow, size up

The generator requires a larger background partition to relocate the modules. Use the system SZ command for this.

Generator aborted.

A file already exists and may not be overlaid, or incomplete command file. Specify new files when running the generator; purge or rename existing generator files. Check the command file for completeness. Check that all the required END commands are used.

GEN record information missing or invalid

This error only occurs during OS/driver partition relocation. The partitioning information in this GEN record in this module is incorrect. If the module is an HP-supplied system module, the relocatable file has become corrupt and a new copy should be obtained. If the module is user-supplied, the GEN record information should be corrected, or the module should be relocated in the unpartitioned portion of system relocation.

ID segment specification error

The format of the command is *ID,numSegs*, where *numSegs* is numeric and non-negative.

IFT command or parameter error

The command to specify IFTs occurred out of order in the command file, was specified incorrectly, or its entry point could not be found. Check the format of the command and its parameters.

Illegal ALlocation type (must be LO or DY)

The only valid parameters to the AL command are LO (locked), or DY (dynamic).

Illegal bound

Check the format of the LOCC or BLOCC commands. In the *LOCC,addr command*, *addr* must be numeric and non-negative.

Illegal driver name

The driver name must be a legal entry point name. The driver relocatable file must be specified correctly.

Illegal LU number

The LU number must be a positive integer in the range of 1 to 255.

Illegal MR or offset value for entry point in OS partition

This error occurs only during relocation of a system partition. The partitioning information contained on the GEN record(s) in this module does not match the information found on the ENT record(s) for the module, or this module is not partitionable because of the type of entry points it contains. If the module is HP-supplied, it has become corrupt and a new copy should be obtained. If the module is user-supplied, it must be corrected to meet the restrictions for partitionable modules or relocated in the unpartitioned portion of system relocation.

Illegal model number

The model number must be a positive integer less than 32767.

Illegal module name

The module name must not be numeric.

Illegal record or checksum error

A record in the file is of incorrect format or is corrupt. Recompile the module (if possible) and retry. Check that the file is a relocatable file.

I/O port map not available for this IFT

Twenty-four port maps may be locked. A map is locked for any IFT with select code less than 50B or when an AL:LO command is included in an IFT specification. All 24 have been locked. No I/O can occur to this IFT.

Interface type error

The interface type specified for this IFT is not numeric or is out of range. Check the specification of the interface type. It must be numeric and in the range 0 through 77B.

Internal error in port map allocation

An internal error in RTAGN has occurred. Please forward appropriate documentation to your HP representative. The problem may be relieved by relinking RTAGN, reordering the IFT commands (along with their DVT commands), or specifying different port map allocation commands (AL).

**Internal error reading output file**

An internal error in RTAGN has occurred. Please forward appropriate documentation to your HP representative. The problem may be relieved by reordering the OS/driver partitions or eliminating as many OS/driver partitions as possible.

Internal table overflow, size up

Use the system command SZ to give the generator more memory for internal tables.

Interrupt table command or parameter error

The INT command to specify interrupt table entries was out of order in the command file or was specified incorrectly.

- 1) Check the format of the command for your system.
- 2) Check that the INT commands follow the END that terminates the list specification.
- 3) Check that the select code specified is within range.
- 4) Check that the select code is specified as an octal number in the range 0 to 77B. The "B" suffix is required.

Link error, current page used

This is a warning message only; the generation defaults to using current page linking. Check the format of the LINK command in the initialization phase. Only *LINK,BP* or *LINK,CP* are allowed.

LU specification error

An LU number is out of range, or has already been assigned to a DVT. An LU number must be in the range 0 to 255. An LU may be assigned to only one DVT.

Memory overflow

The relocated system is too large. Reduce the size of the operating system: reduce the number of tables or partitions. Use base page linking rather than current page linking if possible.

Missing system entry point

A system entry point required by the generator could not be found. An expected system module has not been relocated.

Model number does not match driver

Check for the correct model number and subparameter. Check for a correct file namr.

No DVT entry or no IFT entry

The DVT or IFT was missing or out of order. Check to ensure that there is an END after the last driver partition. Add the DVT or IFT entries.

Only one tag area allowed

Multiple TG commands were found, or the TG command was entered after a partitionable module was relocated (the TG command must precede the module RE command). Remove or combine the multiple TG commands, or move the command entry to immediately follow the VCTR relocation command.

OS/Driver partition size exceeded limit

The OS/driver partition exceeded eight pages. Decrease the number of system modules or drivers relocated in the partition.

OS module or driver not partitionable

The system module or driver relocated in the partition did not have the partitionable record in the source. This is a warning message only; the module/driver is relocated into the partition. Verify that the module or driver does indeed meet the requirements for being relocated into a partition.

PA command must appear before partitioning phase

Move the PA command into the system relocation phase.

Partition specification error

An illegal parameter appears in the RS command.

Partitioning mapping module missing (MAPOS)

The system module MAPOS is required if any system partitions are defined in the generation.

Privileged interrupt error (fatal)

The interrupt table specification for a privileged entry point is incorrect. Check that only privileged devices are in the neighboring select codes (in the appropriate groups of four as described in the IFT command, under the SC parameter, in chapter 5).

Queuing error

The queuing option for the IFT or DVT must be in the form *U:PR* or *QU:FI*.

Record out of sequence

An attempt was made to relocate a module not beginning at the first record of the module. Recompile the module and retry.

Resource number specification error

The RESN command to specify resource numbers occurred out of order in the command file or was specified incorrectly.

- 1) Check the format of the command: *RESN,n*.
- 2) Check that it follows the CLAS command.
- 3) Check that *n* is numeric and in the proper range.

SAM specification error

The size of SAM must be a positive integer, =<32767.

Segment RTAGn not found

Make sure that segments RTAG1, RTAG2, RTAG3, and RTAG4 have been properly RP'd.

Segmented program not allowed

Segmentation is not allowed during any portion of a system generation.

Select code error

The offending select code has already been assigned to an IFT. Select codes must be numeric and in the range 20B to 77B.

Shared program specification error

Shared program specification must be positive and non-zero.

Snap entry duplicate when truncated

All snap file symbols must be unique in their first 16 characters. Change one of the symbols, reassemble, and try again.

Snapshot file error

A read error occurred on the snap file that is created by the generator. Retry the generation.

Specified module not found

The module is not in the file specified by a RElocation, SEarch, or MSearch command. Ensure that the module exists and that the form of the offending command properly references it.

Spool limit specification error

Check to ensure that the spool limits are numeric.

(SPOOL not included in this system, syntax check only)

The system module SPOOL has not been included in this generation, so the spool buffer limit command is not pertinent. The parameters are verified.

Symbol table overflow, size up

The generator requires a larger partition to relocate the modules. Use the system SZ command to accomplish this.

System CDS libraries specified for non-CDS system

The system generation does not include CDS capability, part of VC+ only. Specifying system CDS libraries has no effect, because LINK will not load any CDS programs using this snapshot file.

System CDS libraries not specified for CDS system

The set of CDS libraries was not specified at generation. These libraries must be specified during the memory allocation phase, following the non-CDS libraries.

System memory block specification error

System memory block specification must be positive and non-zero.

Table extension error

Ensure that the table extension parameter of the DVT command is of the form *TX:numWords* and that *numWords* is numeric and in the range of 0 through 511.

Tag area overflow

All the words in the tag area have been used and more are required. Ensure that a TG command that allocates a tag area of sufficient size appears after relocation of the VCTR module.

Time-out value error

Ensure that the time-out parameter in the DVT command is specified in the form *TO:timeValue* and that *timeValue* is in the range of 0 through 32767.

Timeslice or priority error

Both values must be in the range of 0 through 32767.

Too many module names

More module names have been specified in PA commands than can be processed by RTAGN(20).

Too many partitions specified

Only 127 OS/Driver partitions can be defined during the partition relocation phase. Remove or consolidate system modules and/or drivers as necessary to stay within the limit.

Undefined driver entry point

Correct the entry point specified or relocate the driver with the required entry point in the system relocation phase.

Undefined externals remain

Search the library that contains the required reference, or relocate the routine that contains the reference as an entry point.

Unknown command

Use a command legal to the particular generator phase. Use legal format for the command, parameters, and data.

Upper spool limit should be 1,000 words less than SAM

The spool limit should be 1,000 words less than SAM to allow enough SAM for other system functions.

User table specification error

The user table specification must be positive and non-zero.

WARNING: LOCC set backward

The relocation pointer has been set backward to overlay a portion of the previously-created operating system. Ensure that it is really necessary that this portion of the generation be overlaid.

WARNING: No system messages included

No modules were relocated during the system message relocation phase. If the system file created is actually booted and used, no messages will be printed by the system when a system error occurs (program abort, swap file overflow, illegal interrupt, etc.).

WARNING: System common module not type 6

The generator expected the relocation of a type 6 module. This is a warning only, since other module types may use system common. (The system controls access to type 6 modules.)

BUILD Program Error Messages

If BUILD is being run interactively, it will report any errors and repeat the appropriate prompt. If BUILD is running through a command file, then the detection of an error will invoke the error path defined by the runstring option.

Errors are described below, with an indication of the exit path in the case of command file input. All errors are preceded with a *error - , and warnings with a *warning - indication. In parentheses after each description is an indication of the command(s) where the error is applicable. All messages should be assumed to be errors unless a warning is specifically indicated.

FMP errors (file system errors) will be reported in the same way as CI and most other utilities, with a description of the error and the name of the file. For instance if you give the name of a file which does not exist, the message '*error - No such file PROGRAM.RUN' will be displayed.

Bad parameter type

A parameter of the wrong type was specified (e.g., a file namr rather than a number). (All prompts and commands with required parameters can result in this error.)

Bad parameter type in the runstring.

The user gave a runstring parameter which was numeric where a file namr (or null parameter) was expected. Lack of a fixed position instruction which all systems contain. (System file prompt)

Cannot RP a program with overlays.

The user tried to RP a program with overlays. Since BUILD makes memory image systems, and cannot assume the program will be on a disc accessible by this memory based system, BUILD can only RP programs without overlays. (RP) Note however that CDS programs with multiple code segments can be loaded into memory based systems. Thus very large programs (up to 4MB) can be loaded into memory based systems (92078 only).

Corrupted program file.

Before relinking a program file, BUILD insures the header ID segment checksum is correct. This error occurs if an incorrect checksum is detected, or if for any other reason BUILD concludes the file has been corrupted (e.g., it has been truncated). (RP)

Default error path (abort) used

The error path specified in the runstring was not /C, /E or /A. The default error path, /A, will be used if an error occurs while running non-interactively.

Illegal snapshot

The checksum for the snapshot header was incorrect, or some label required by BUILD could not be found. (Snapshot file prompt)

Incorrect file type.

The user tried to RP a file which was not of the correct type. BUILD will only RP a type 6 file. (RP)

Invalid command in present context

The command was not recognized by BUILD. This error occurs for all undefined commands, and will also occur should the user try the PA command when automatic partitioning is requested, or an RP option before an RP command.

No free ID segment.

The program could not be loaded because all ID segments have been used. Although no more programs can be RP'd once this error occurs, interactive users can still specify other commands (PT, SS,etc.) (RP only)

No free partition large enough for program

The user did not specify automatic partition construction, and all remaining partitions are too small to contain the program which was requested to be loaded. (RP,SZ)

No free partition

The program could not be loaded because all partitions have been used. If auto partitioning, this means there are no more memory descriptors. (RP)

No memory descriptor available for shareable EMA area.

A program that was RP'd accesses a shareable EMA area. BUILD was trying to allocate memory for the area, but all memory descriptors had been used for programs and other shareable EMA areas. Either rerun BUILD and RP fewer programs, or regenerate your system and specify more memory descriptors, using the RS command (/E).

Not a system image

The specified system image could not possibly be a system image due to its type, size, or other characteristic.

Not enough memory for program

The program being RP'd is too large to fit in the remaining memory. (RP)

Not enough remaining memory for shareable EMA area.

A program that was RP'd accesses a shareable EMA area. Build was trying to allocate memory for the area, but not enough memory remained after all the programs had been placed in memory. You are trying to fit too many programs and shareable EMAs into memory (/E).

Output file too small for system.

The output file created to hold the bootable system is too small. There is not enough space for the generated system (or any programs). If you specified a file size when giving the output file name, give a sufficiently larger size.

Parameter out of range

The user gave a parameter which is outside of the range of sensible responses. (Partition and memory size prompts, PA,PR, SZ)

Partition in use.

A user tried to load a program into a partition which already has been used. (PA, not automatic partition construction)

Partition is too small.

The user specified a partition number with the PA command which is of a smaller size than required by the program. (PA)

Program already loaded.

The user tried to load the same program twice. The newname option can be used to eliminate this problem. (RP)

Program not transportable, must be reloaded.

The program cannot be loaded into this system. This can be a result of using system common, this system may be significantly different from the one which this program was loaded for or the program may access unusual system entry points which are not transportable. Reload the program for this system.

Ran out of disc space <filename>

This is an FMP error, but one which is fairly frequent. If the size of the output file is not specified, BUILD uses a default size of 2048 blocks. If there is not enough space for this file, this message will appear. Either specify that the file should go on a disc with more space, or specify a file size which is smaller. After BUILD finishes, this file will be truncated to the smallest number of blocks possible.

Shared program table full, program not shared.

The program was shareable, but there was no room in the shared program table. The program will run but not as a shareable program. Additional copies will require additional code partitions. (VC+ only) (RP)

Specified size is smaller than the minimum required.

A user attempted to give a program size which was less than the minimum required by the program. (SZ)

System not for snap

The system image was not for the snapshot file given. (System file prompt)

System not generated for CDS programs.

The program uses CDS instructions, but the system lacks the necessary routines to run CDS programs. Regenerate the system to include the CDS software. (HP 92078 option only) (RP)

***Warning – RPL checksum does not match.**

When loading in a type 6 file, BUILD noticed a change in the RPL checksums. The program was loaded in, but any routines called which should have been RPL'd may not have been installed properly. Note that different RPL checksums will be calculated if the RPLs in the system were specified in a different order when the system was generated. (RP)

VCP Loader Errors

The following are the error codes in response to errors in VCP loader commands %B and %L.

LOADER ERROR 2	Select code less than octal 20
LOADER ERROR 3	No card with that select code

Cartridge Tape Loader Errors

LOADER ERROR 110	File forward error: status in B-Register
LOADER ERROR 111	Checksum error
LOADER ERROR 112	No data before EOF

LOADER ERROR 120	Write error: status in B-Register
------------------	-----------------------------------

PROM Card Loader Errors

LOADER ERROR 211	End of programs
LOADER ERROR 212	Bad Format
LOADER ERROR 213	System larger than 32k: must start on card boundary
LOADER ERROR 214	Write not allowed to ROM

DS/1000 Loader Errors

LOADER ERROR 310	TO after CLC 0: is select code a DS card?
LOADER ERROR 311	Checksum error: is P file absolute binary?
LOADER ERROR 312	TO after download request
LOADER ERROR 313	TO after file number
LOADER ERROR 314	Bad transfer (central generated): status in B-Register
LOADER ERROR 315	TO after buffer request
LOADER ERROR 316	TO after count echo
LOADER ERROR 317	TO waiting for data
LOADER ERROR 320	TO after VCP mode request DS write
LOADER ERROR 321	Remote node will not accept data: status in B-Register
LOADER ERROR 325	Record out of sequence

Disc Loader Errors

LOADER ERROR 411	TO reading disc type or error in Initialization process: check HPIB address, card, cable
LOADER ERROR 412	TO reading status: is device a disc?
LOADER ERROR 413	Status error: status in B-Register
LOADER ERROR 414	TO during file mask
LOADER ERROR 415	TO during seek
LOADER ERROR 416	TO during read/write command
LOADER ERROR 417	TO during DMA of data
LOADER ERROR 420	Parity error during DMA transfer
LOADER ERROR 421	TO during FIFO flush
LOADER ERROR 422	TO during DSJ command
LOADER ERROR 423	Bad DSJ return: returned value in B-Register. Probably there is no valid BOOTEX on the disc and the disc transfer has overwritten memory.
LOADER ERROR 460	Disc not identifiable: disc ID in B-Register

Magnetic Tape Loader Errors

LOADER ERROR 510	Time out during initialization/read ID.
LOADER ERROR 511	Time out when issuing end/select unit.
LOADER ERROR 512	Magnetic tape off line.
LOADER ERROR 513	No write ring.
LOADER ERROR 514	Time out during End command.
LOADER ERROR 515	Time out waiting for rewind completion.
LOADER ERROR 517	Time out waiting for DMA transfer.
LOADER ERROR 520	Parity error during DMA transfer.
LOADER ERROR 521	Time out doing a PHI flush.
LOADER ERROR 522	Time out waiting for DSJ.
LOADER ERROR 523	Bad DSJ response.
LOADER ERROR 525	Time out waiting for Mag Tape Not Busy.
LOADER ERROR 530	Time out after issuing a command.
LOADER ERROR 531	Parallel Poll time out after issuing a command.
LOADER ERROR 535	Bad status after read/write command.
LOADER ERROR 550	No Data transfer (read only).
LOADER ERROR 560	Not mag tape ID.

HP 12022A Disc Interface Loader Errors

LOADER ERROR 610	Time out after SDH (Sector Drive Head) for read/write.
LOADER ERROR 611	Time out after cylinder high.
LOADER ERROR 612	Time out after cylinder low.
LOADER ERROR 613	Time out after sector.
LOADER ERROR 614	Time out after sector count.
LOADER ERROR 615	Time out after read/write command.
LOADER ERROR 616	Time out after DMA read/write transfer.
LOADER ERROR 617	Parity error during transfer.
LOADER ERROR 620	Fixed disc not ready.
LOADER ERROR 630	Time out after request status register.
LOADER ERROR 631	Time out after read status register.
LOADER ERROR 632	Time out after waiting for not busy.
LOADER ERROR 633	Time out after request error register.
LOADER ERROR 634	Time out after read error register.
LOADER ERROR 635	Status Error: A-Register = status register; * B-Register = error register. *
LOADER ERROR 650	Time out after SDH register for restore.
LOADER ERROR 651	Time out after restore.
LOADER ERROR 660	Disc not defined.

Other Loader Errors

LOADER ERROR 1024 or
LOADER ERROR 1025 Possible meanings:

1. Booting from CS/80 disc that has just been push button restored from CTD tape or booting diagnostics directly from the tape. The CTD tape may not have been certified/formatted before data was stored to it.
2. Booting from a CTD tape in ASAVE format.
3. Booting from the CS/80 disc was not successful. Bootex may be corrupted.
4. Faulty tape control board in the CS/80 runstring.
5. Incorrect VCP file number in the runstring.

* Refer to the CS/80 Disc Driver DD*33 section of the RTE-A Driver Reference Manual (92077-90011) for descriptions of the Status Register and the Error Register.

Worksheets

This appendix provides the blank worksheets needed as you plan and complete the system generation answer file. The worksheets are designed so that you can fill them in, copy them into a generator answer file, and run the generator on the answer file to produce the new system files.

The worksheets included in this appendix are:

- Initialization and System Relocation Worksheet
- System I/O Configuration Worksheet
- IFT/DVT Worksheet
- Table Generation Worksheet
- Memory Allocation Worksheet

 * Initialization and System Relocation Worksheet *
 *

*
 * INITIALIZATION PHASE
 *

LINKS,CP ,Current or Base page linking

*
 *

* SYSTEM RELOCATION PHASE (Use one of the following RPLs)
 *

*re /rte_a/%rpl40,,	A400		NO
*re /rte_a/%rpl41,,	A400	NO	YES
*re /vcplus/%rpl42,,	A400	YES	NO
*re /vcplus/%rpl43,,	A400	YES	YES
*re /rte_a/%rpl60,,	A600	NO	NO
*re /rte_a/%rpl61,,	A600+	NO	YES
*re /vcplus/%rpl63,,	A600+	YES	YES
*re /rte_a/%rpl70,,	A700	NO	NO
*re /rte_a/%rpl71,,	A700	NO	YES
*re /vcplus/%rpl72,,	A700	YES	NO
*re /vcplus/%rpl73,,	A700	YES	YES
*re /rte_a/%rpl90,,	A900	NO	YES
*re /vcplus/%rpl91,,	A900	YES	YES

er

re %VCTR::master,,
 tg 950

Entry Points
 Number of Tags Required

* SYSTEM MODULES
 *

re %UTIL::master,,	System Variables, Utilities
re %ABORT::master,,	Abort Processing
re %IORQ::master,,	I/O Request Processing
re %MAPS::master,,	Dynamic Mapping System Routines
re %PROGS::master,,	Program State Processing
re %EXEC::master,,	Program Control Module
re %SAM::master,,	System Available Memory
re %TIME::master,,	Time Scheduling of Programs
re %SCHED::master,,	Allows Programmatic Program Scheduling
re %STRNG::master,,	Allows Passing of Runstrings
re %LOCK::master,,	Allows LU Locks and Resource Numbers
re %MAPOS::master,,	Partitioned OS tag routines
re %MEMRY::master,,	Memory manager
re %LOAD::master,,	Program loading and swapping
re %ERLOG::master,,	Error Logging

```
re %OPMSG::master,,
re %SYCOM::master,,
re %RTIOA::master,,
re %IOMOD::master,,
re %CLASS::master,,
re %id*43::master,,
re %SPOOL::,,
```

Operating System Messages
Operator Commands
Real-Time I/O Control
I/O Subroutines
Class I/O
Power Fail Driver
Required for VC+ systems only

*

* NON-PARTITIONED DRIVERS

*

```
REL,
REL,
REL,
REL,
REL,
REL,
REL,
REL,
REL,
```



*

* SEARCH LIBRARIES

*

```
se $SYSLB.LIB::master,,
se $MATH.LIB::master,,
pa PERR,XCMND,STAT,DSQ,VEMA
pa CDSFH
ms $SYSA.LIB::master,,
```

Search the System Library
Search the Math Library

Search the System Dummy Library

*

```
end,
```

Terminate System Relocation

*

*

* OS PARTITIONS

*

*

```
re %XCMND::master,,
al
re %STAT::master,,
re %DSQ::master,,
end
```

Operator Command Extension

Status Commands

*

*

```
* re %CDSFH::master,,
al
re %PERR::master,,
re %VEMA::master,,
end
```

CDS Systems Only

Parity Error Handling
Virtual Memory

*

*


```
* Driver partitions
*
*
re %dd*33::master
end
*
re %id*66::master
re %id*00::master
end
*
re %id*37::master
re %id*52::master
end
*
re %idm00::master
re %dd*23::master
end
*
re %dd*00::master
re %id*50::master
re %dd*30::master
end
*
re %dd*12::master
re %adv00::master
re %dd*20::master
end
*
re %ddc12::master
end
*
re %id*27::master
re %dd*24::master
end
*
*       end driver partition
*
```

 *
 * System I/O Configuration Worksheet *

IFT, _____, SC: _____, E _____, QU: _____, TX: _____,
 Interface Select Entry Point Queuing Table
 Driver Namr Code Entry Point Queuing Table
 (Default file) Code Entry Point Queuing Extension

IT: _____, AL: _____

 Interface Port Map
 Type Allocation

Device Name:					
Device Driver:					
Defaults File:					
Model Number:	M	M	M	M	M
Logical Unit:	LU:	LU:	LU:	LU:	LU:
Device Type:	DT:	DT:	DT:	DT:	DT:
Device Priority:	PR:	PR:	PR:	PR:	PR:
Time Out:	TO:	TO:	TO:	TO:	TO:
Buffer Limits:	BL: ::	BL: ::	BL: ::	BL: ::	BL: ::
Table Extension:	TX:	TX:	TX:	TX:	TX:
Driver Extent:	DX:	DX:	DX:	DX:	DX:
Driver Prams:	_____	_____	_____	_____	_____
start #	DP:1	DP:1	DP:1	DP:1	DP:1
1	:	:	:	:	:
2	:	:	:	:	:
3	:	:	:	:	:
4	:	:	:	:	:
5	:	:	:	:	:
start #	DP:6	DP:6	DP:6	DP:6	DP:6
6	:	:	:	:	:
7	:	:	:	:	:
8	:	:	:	:	:
9	:	:	:	:	:
10	:	:	:	:	:
start #	DP:11	DP:11	DP:11	DP:11	DP:11
11	:	:	:	:	:
12	:	:	:	:	:
13	:	:	:	:	:
14	:	:	:	:	:
15	:	:	:	:	:
Queuing:	QU:	QU:	QU:	QU:	QU:
Node 1:					
Node 2:					
Node 3:					
:					

```

*****
*
*           Table Generation Worksheet
*
*****

```

```

*
* INTERFACE TABLE and DEVICE TABLE GENERATION
*

```

```

* IFT Format:
*

```

```

* IFT,{namr},SC:sc,{Entry point},{QU:qu},{TX:tx}{IT:it}{,AL,al}{,PS}
*

```

```

* DVT Format:
*

```

```

*     DVT, {namr},{Mmodel number},-
*         LU: logical unit,-
*         {Entry point},-
*         {TO: time out},-
*         {BL:bb:lower limit: upper limit},-
*         {DT: device type },-
*         {TX: table extension},-
*         {DX: driver extension words},-
*         {DP: start #: parameters ... },-
*         {QU: queuing option}

```

```

* Use the IFT/DVT worksheets

```

```

* END ,End of DVT input
* END ,End of IFT input
*

```

```

* NODE SPECIFICATION

```

```

* NODE,
* NODE,
* NODE,
*
* END ,End of NODE input
*

```

```

* INTERRUPT TABLE GENERATION
*

```

```

* INT, ,Format: INT,SC,entry point
*
* END ,End of Interrupt table input

```

```

*****
*
*           Memory Allocation Worksheet           *
*
*****

```

```

*
CLAS,           ,Class number allocation
*
RESN,           ,Resource number allocation
*
ID,             ,ID Segment allocation
*
RS,             ,Memory descriptor allocation
*
SAM,            ,SAM allocation
*
SL,             ,Spool limits
*
BG,             ,Background swap priority allocation
*
QUANTUM,        ,Time slice value
*
SP,             ,Shared programs
*
MB,             ,Memory block for DS (optional)
*
US,             ,Number of concurrent users (VC+ only)
*               ,required, set to 1 for non VC+ system.
*
LB,             ,LOGOF buffer limit, blank for non VC+
*               ,system or for default for VC+ system.
*

```

```

*****
*
*           Memory Allocation Worksheet (continued)
*
*****

```

```

*
*   LABELED SYSTEM COMMON RELOCATION
*
REL,
*
REL,
*
REL,
*
.
.
END           ,End of Labeled system common relocation
*
*   UNLABELED COMMON and LIBRARY SPECIFICATION
*   COM,           ,Allocate unlabeled system common
*
RE,%MSGTB::master,,           Message table
END
**   Relocate message module
RE,%$MS000::master
END
END,           ,End message relocation
*
*   System Libraries
*
lib sec1000.lib
lib $bigds.lib
lib $biglb.lib
*
*
end
*
*   CDS System Libraries
*
*
lib sec1000cds.lib
lib $bigcds.lib
lib $sec1000.lib
lib $biglb.lib
*
end

```


Configuring a Disc Subsystem

HP supplies a system configuration which may be used with the default values found in the relocatable code for the disc device driver. Logical unit numbers may be assigned to these default cartridges by using the LU:lu and Mmodel number parameters in the DVT command string:

```
DVT, %DD*33::RE, M7908_CF:0, LU:16, ...  
DVT, %DD*33::RE, M7908_CF:1, LU:17, ...
```

M7908_CF:x is the model number and default value designator. To see what these default values are for the various discs, refer to the appropriate table contained in Appendix E of this manual.

If you want to configure the disc specifically, you need to enter commands as shown below, where the letters a through h represent the values you want placed in the driver parameter locations in memory:

```
DVT,%DD*33::RE,M7908_CF:0,LU:16,DP:1:a:b:c:d:e,DP:6:f:g:h  
DVT,%DD*33::RE,M7908_CF:1,LU:17,DP:1:a:b:c:d:e,DP:6:f:g:h
```

Here, values a through e are to be placed in driver parameter locations 1 through 5, and values f, g and h are to be placed in locations 6, 7 and 8. DP:1 points to the first five locations, DP:6 points to the next three locations.

Disc Subsystem Categories

There are three categories of disc subsystems: the CS/80 (command-set 80 and subset 80) discs with and without Cartridge Tape Drive (CTD), Multiple Access Controller (MAC) discs, and the compact disc subsystems. The model numbers associated with each type of drive are:

CS/80:	7908P/R	-	16Mbyte	(fixed disc)
	7941A	-	24Mbyte	(fixed disc without CTD)
	7942A	-	24Mbyte	(fixed disc with CTD)
	7911P/R	-	28Mbyte	(fixed disc)
	7907A	-	41Mbyte	(one fixed, one removable disc)
	7945A	-	55Mbyte	(fixed disc without CTD)
	7946A	-	55Mbyte	(fixed disc with CTD)
	7912P/R	-	65Mbyte	(fixed disc)
	7957A	-	82Mbyte	(fixed disc)
	7958A	-	130Mbyte	(fixed disc)
	7914P/R	-	132Mbyte	(fixed disc with/without CTD)
	7914CT	-	132Mbyte	(fixed disc with CTD)
	7914ST	-	132Mbyte	(fixed disc with 7974A tape unit)
	7936A	-	300Mbyte	(fixed disc)
	7933A	-	404Mbyte	(fixed disc)
	7935A	-	404Mbyte	(removable disc)
	7937A	-	571Mbyte	(fixed disc)
MAC:	7906M/S	-	20Mbyte	(1 fixed, 1 removable disc)
	7920M/S	-	50Mbyte	(removable disc)
	7925M/S	-	120Mbyte	(removable disc)
Compact:	248x*	-	10Mbyte	(integral fixed)
		-	14.5Mbyte	(integral fixed)
		-	20Mbyte	(integral fixed)
	12122A	-	20Mbyte	(fixed)
		-	270Kbyte	(floppy)
		-	630Kbyte	(floppy)
	9121S	-	270Kbyte	(floppy)
	9121D	-	2x270Kbyte	(floppy)
	9122D	-	2x630Kbyte	(dual-sided floppy)
	9133XV	-	15Mbyte	(fixed)
		-	270Kbyte	(floppy)
	9133D	-	15Mbyte	(fixed)
		-	630Kbyte	(floppy)
	9133H	-	20Mbyte	(fixed)
		-	630Kbyte	(floppy)
9133L	-	40Mbyte	(fixed)	
	-	630Kbyte	(floppy)	
9895A	-	2.3Mbyte	(dual-sided floppy)	

* Micro/1000 only

Disc Configuration Steps

Disc configuration involves several steps:

1. Divide the disc into groups of contiguous tracks.
2. Allocate spare tracks for each of these groups (9895, 9121, and MAC discs only).
3. Assign a logical unit (LU) number to each of these groups of tracks. The operating system will allow access to the information on the tracks through the LU number.
4. Fill out the IFT/DVT Worksheet and create the answer file the generator will use to configure your disc subsystem.

Configuring Non-CS/80 Discs

This section covers the physical structure of a disc drive, the logical structure of a disc, and the driver parameters associated with a disc subsystem.

Physical Disc Structure

The magnetic storage medium in a disc drive is the disc platter. The disc drive contains one or more platters, each of which has either one or two surfaces available for storage of data. Access to the data stored on the surfaces is accomplished through disc heads.

Disc heads are electro-mechanical devices that move over a surface, reading data from the disc or writing data to the disc as the disc rotates. There is one head per disc surface. If the disc has more than one head, all of the heads are connected to the same servo-motor, thus all heads move together.

Logical Disc Structure

Information on each surface is organized into tracks, which are concentric rings where information is actually stored and accessed. Each track is divided up into blocks of 128 words.

The disc driver allows information to be written across track boundaries. This means that a system disc may operate in one of two modes: surface mode or cylinder mode. When you configure a disc, the LUs on that disc must be all in surface mode or all in cylinder mode. Each disc type has a standard mode that is determined by access considerations. Access modes are further discussed in the sections on the individual disc types.

Surface Mode

In surface mode, each disc LU is made up of tracks that are all on one disc surface and are accessed serially. For example, if you have a large file that starts on track 0 of surface 0, it continues on track 1 of surface 0, then track 2, track 3, and so on. This is the standard configuration for the 7906M, and the only mode supported for boot of this disc.

Cylinder Mode

In cylinder mode, tracks are arranged in groups of cylinders. A disc cylinder includes all tracks with a given track number on all surfaces of the disc. For example, cylinder 5 is made up of track 5 on surface 0, track 5 on surface 1, and so on for all surfaces of the disc. The tracks are on separate surfaces immediately above and below one another, as shown in Figure C-1. This is the standard configuration for a 7920 or 7925, and the only mode supported for boot of these discs.

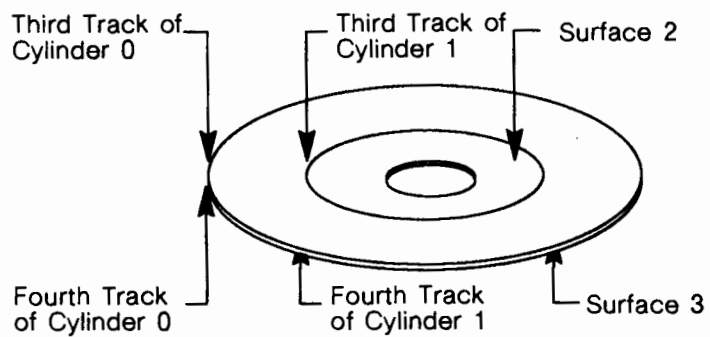
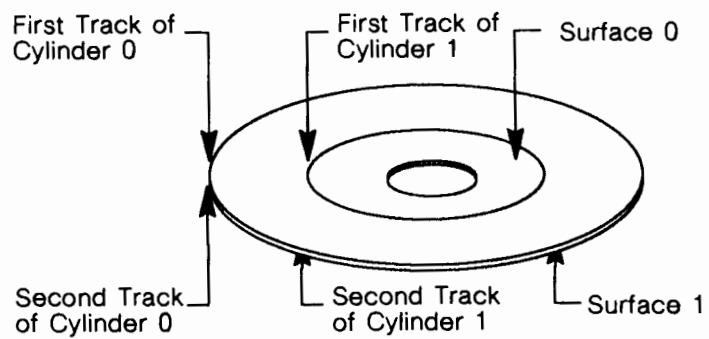


Figure C-1. Arrangement of Disc Cylinders

In cylinder mode, all of the tracks in one cylinder are accessed before the first track on the next cylinder is accessed. For example, if you have a large file that starts on track 0 of surface 0, it continues on track 0 of surface 1, and on through track 0 of the last surface. The file then continues on track 1 of surface 0, track 1 of surface 1, and so on. See Figure C-2 for the access sequence.

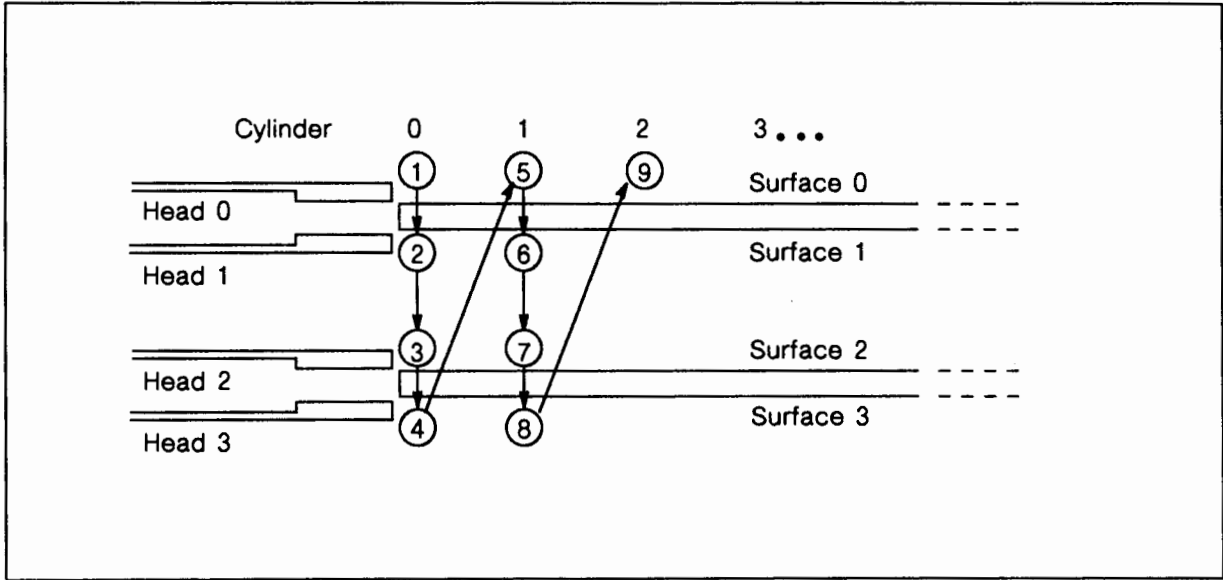


Figure C-2. Track Access in Cylinder Mode

File System Considerations

For a disc divided into file system volumes, a small number of large volumes is recommended. This allows allocating files from a common pool of free space, which makes better use of space. The files can be logically organized into different directories and subdirectories. For FMGR cartridges, disc space is logically divided by LUs, with one directory per LU. Dividing a disc into several LUs is one means of organizing data on the FMGR cartridge.

If your system has files on FMGR cartridges as well as the RTE-A file system, it is best to have at least one LU of each type in your system. In general, it is desirable to have a few FMGR disc LUs, with the remaining disc space allocated to one or two very large file system volumes. For more information on the file system, refer to Chapter 10 of the System Design Manual.

Non-CS/80 Disc Driver DD*30 Parameters

Configuring a disc into the desired logical units involves setting driver parameters in the DVT for each LU. These parameters will contain all necessary track map information. The driver parameters for the non-CS/80 disc DVT entries are the following:

- Driver Parameter
- 1: HP-IB Address
 - 2: Disc Drive Unit Number
 - 3: Starting Head for LU
 - 4: Starting Cylinder for LU
 - 5: Number of Spares for LU
 - 6: Number of Tracks for LU
 - 7: Number of Blocks per Track for Disc
 - 8: Number of Surfaces for LU

To help in configuring the disc, you could name the driver parameters on your IFT/DVT worksheet (Appendix B) as follows:

Driver Prams:	-----	-----	-----	-----	-----
start #	DP: 1	DP: 1	DP: 1	DP: 1	DP: 1
1 HP-IB Addr.	:	:	:	:	:
2 Unit Number	:	:	:	:	:
3 Start Head	:	:	:	:	:
4 Start Cyl	:	:	:	:	:
5 Spares	:	:	:	:	:
start #	DP: 6	DP: 6	DP: 6	DP: 6	DP: 6
6 Tracks	:	:	:	:	:
7 Blocks/Track	:	:	:	:	:
8 Surfaces	:	:	:	:	:

HP-IB Address (DP1)

The HP-IB address is a switch-selectable number that identifies a disc (or any other device) connected to an HP-IB card. The HP-IB address must be a unique number for each device. (Directions for setting this number may be found in the reference manual for your device.) Enter the HP-IB address as Driver Parameter 1.

Disc Drive Unit Number (DP2)

This number is important only for the 9895, 7906M, and 7925M disc drives and the 248x integrated hard disc. For any other type of disc drive, enter 0. The disc drive unit is a switch-selectable number that identifies the drive to the disc controller. Directions for setting this number are contained in the reference manual for your disc.

Starting Head for LU (DP3)

Disc head numbers start at 0 (the top surface). This parameter defines the head (or surface) on which a disc LU begins.

Starting Cylinder for LU (DP4)

Cylinder numbers start at 0 (the outermost cylinder). This parameter defines the cylinder on which a disc LU begins.

Number of Spares for LU (DP5)

Each disc LU is assigned a number of spare tracks to be used if a bad track is detected. When a bad track is spared, all references to that track are switched to the spare (good) track. There are no specific rules for sparing tracks, but it is common to reserve approximately 2% of each LU for spares.

The utility FORMT is used for sparing tracks on the 9895, 7906M, 7920M, 7925M, and 9121 discs. For all other discs, set the number of spare tracks to zero.

Number of Tracks for LU (DP6)

This parameter is the total number of tracks in the disc LU. This includes all tracks (on all surfaces) of the LU, but does not include spare tracks. It is the total number of tracks that will be available for your use:

$$\begin{aligned} &[(\text{number of heads}) * (\text{number of cylinders})] - \text{number of spare tracks} \\ &= \text{number of tracks for LU} \end{aligned}$$

Number of Blocks per Track for Disc (DP7)

This number indicates the amount of storage available on each track of the disc. This number is different for each disc type and is given in the sections on the individual disc types. Each block is equivalent to 128 words (or 256 bytes).

Number of Surfaces for LU (DP8)

This parameter specifies the number of surfaces that make up the LU. In effect, this tells whether the LU is configured in surface mode or cylinder mode. You must specify either one surface (surface mode) or the total number of surfaces on the disc drive (cylinder mode). This number must be the same for all disc LUs referring to the same disc controller.

The number of surfaces is usually either one (for surface mode) or all (for cylinder mode). The 7906M disc drives, however, can be used in shortened cylinder mode, which can allot two surfaces to a cylinder. This allows the top (removable) platter to be configured separately from the bottom (fixed) platter. Shortened cylinder mode is provided because data can be stored on a 7906M removable platter in that mode on an RTE-6/VM system. In order to read such a platter on an RTE-A system, you would need to configure a drive in shortened cylinder mode. Note that cylinder mode has no performance advantage over surface mode on a 7906M, and you can boot from this disc only in surface mode.

Configuring CS/80 Discs

Cylinder mode is the only mode of the CS/80 discs. CS/80 discs are addressed by logical blocks, each of which has a three-word address associated with it. Each block is 128 words in length. The driver maps these blocks into logical units that represent tracks and sectors in a manner similar to that for the non-CS/80 discs. These areas are specified by the driver parameter area, which has the starting block of the LU (three words), the number of logical tracks, and the number of logical blocks per track for the LU. (The number of blocks in the LU is determined by multiplying the number of tracks by the number of blocks per track). In configuring your disc subsystem, do not allow the areas specified to overlap each other and do not exceed the highest block address of the disc.

CS/80 Driver Parameters

Configuring a disc into the desired logical units involves setting driver parameters in the DVT for each LU. These parameters will contain all necessary track map information. The driver parameters (DP:) for the DVT entries have the following form:

Disc Driver Parameters

DP1 HP-IB Address
DP2 Disc Drive Unit (8 bits); Volume Numbers (8 bits)
DP3 MS byte \
DP4 } Starting Block Number (Three words)
DP5 LS byte /
DP6 Number of Tracks for LU
DP7 Number of Blocks per Track for Disc LU
DP8 Reserved (Must be 0)

Integrated CTD (Cartridge Tape Drive with 7908/11/12/14/42/46 drives) Driver Parameters

DP1 HP-IB Address
DP2 Unit Number (8 bits); Volume Number (8 bits)
DP3 C(1 bit); Disc Unit Number(7 bits); Volume Number(8 bits)
DP4 \< Starting Block Address of Disc Cache
DP5 / (two words)
DP6 \< Address of First Cache Block; dynamically set by driver
DP7 / (two words)
DP8 Reserved (Must be 0)

Standalone CTD (without disc drive) Driver Parameters

DP1 HP-IB Address
DP2 Unit Number (8 bits); Volume Number (8 bits)
DP3 0
DP4-8 Reserved

DP4 through DP9 are reserved for future use if DP3, bit 15, is set to 0.

To help in configuring the disc, modify the driver parameter section of the IFT/DVT worksheet to include the above parameters explicitly:

Disc Driver Parameters		LU Number				
DP1	HP-IB Addr.	DP: 1	DP: 1	DP: 1	DP: 1	DP: 1
DP2	Unit & Vol #	:	:	:	:	:
DP3	} Start } Block	:	:	:	:	:
DP4		:	:	:	:	:
DP5	} Number	:	:	:	:	:
DP6	Tracks	:	:	:	:	:
DP7	Blocks/Track	:	:	:	:	:
DP8	Reserved	DP: 8	DP: 8	DP: 8	DP: 8	DP: 8
	:					
	:					

CTD Driver Parameters		LU Number				
DP1	HP-IB Addr.	DP: 1	DP: 1	DP: 1	DP: 1	DP: 1
DP2	CTD U/V #	:	:	:	:	:
DP3	Cache U/V #	:	:	:	:	:
DP4	} Disc Cache } Block	:	:	:	:	:
DP5		:	:	:	:	:
DP6	} Disc Cache } Block	DP: 6	DP: 6	DP: 6	DP: 6	DP: 6
DP7		:	:	:	:	:
DP8	Reserved	:	:	:	:	:
	:					
	:					

HP-IB Address (DP1)

The HP-IB address is a switch-selectable number that identifies a disc (or any other device) connected to an HP-IB card. Directions for setting this number may be found in the reference manual for your device. Enter the HP-IB address as driver parameter 1.

Disc Drive Unit Number (DP2)

The disc drive unit is a number that identifies the drive to the disc controller. Directions for setting this number may be found in the reference manual for your disc.

For cached access to a Cartridge Tape Drive, the unit and volume parameter (CTD driver parameter 3) has the most significant bit, bit 15, set to 1. The disc unit number is contained in the upper 7 bits while the disc volume number is in the lower 8 bits.

Cached access cannot be used with the 9144 standalone CTD. With this device, the most significant bit of DP3 is set to 0. The remainder of the word is reserved for future use.

Starting Block of Disc (DP3,DP4,DP5)

Driver parameter 3 is the most significant part of the address and driver parameter 5 is the least significant. If DP3 is set to zero, then DP4 and DP5 form a double-integer starting-block address. The formula used in calculating the block address is:

$$\text{Most Significant Block} = \text{Previous Starting Block} + (\text{Number of Tracks} * \text{Number of Blocks/Track})$$

For example, a sample generation includes the following:

```
DVT,%DD*33::RE,LU:16,M7912_CF:0,DP:1:0:0,DP:3:0:0:400,DP:7:48:0
DVT,%DD*33::RE,LU:17,M7912_CF:1,DP:1:0:0,DP:3:0:0:19200:943,DP:7:48:0
:
DVT,%DD*33::RE,LU:18,M7912_CF:2,DP:1:0:0,DP:3:0:0:64750:320,DP:7:48:0
DVT,%DD*33::RE,LU:19,M7912_CF:3,DP:1:0:0,DP:3:0:1:14592:266,DP:7:48:0
:
```

The above indicates that device driver DD*33 is being used for a 7912 disc drive. The first subdivision of the 7912 (M7912_CF:0) starts at block number 0 (parameters 4 and 5 are zero). There are 400 tracks in this subdivision (parameter 6 equals 400). There are 48 blocks per track (parameter 7). The second subdivision of the 7912 (M7912_CF:1) starts at block 19200 (parameter 5) and contains a total of 943 tracks (parameter 6).

Number of Tracks for LU (DP6)

The Number of Tracks parameter is the total number of tracks in the disc LU. To calculate the value of DP6, take the total number of blocks and divide by the blocks/track number (DP7). The value for DP6 must not exceed 32767.

Number of Blocks per Track for Disc (DP7)

The Number of Blocks per Track for Disc indicates the amount of storage available on each track of the disc. Each physical block contains 128 words or 256 bytes.

The default for this parameter is 48 blocks per track. Be aware that when using this default value, there may be some unused blocks (fewer than 48) on a CS/80 disc. For example, the 7933 disc has a capacity of 1,579,872 blocks. Dividing this by the default 48 blocks/track yields 32,914 tracks. Note that the file system has a limitation of 64 blocks per track; you cannot mount a disc LU with a track size greater than 64 blocks per track.

Reserved Parameter (DP8)

The Reserved parameter must always be 0.

Starting Block Address of Disc Cache

All accesses to and from the Cartridge Tape Drive are made via a disc cache. The size of the disc cache must be in 256 physical blocks; therefore, if the layout of the disc is defined differently from the primary, be sure there is room for the 256 physical blocks for the disc cache.

CAUTION

If you are changing the track map layout of a CS/80 disc drive which has an integral (Linus) tape drive, take precautions regarding the starting block address and size of the tape cache to prevent spurious file corruption.

Address of First Cache Block

Driver parameters 6 and 7 should be zeros. This value is set dynamically by the driver to indicate the current tape block address of the cache data.

Sample CS/80 Worksheets

Sample 7907 Worksheet, Fixed Disc (Unit 0)

LU	18
DP1	0
DP2	0
DP3	0
DP4	0
DP5	64750
DP6	320
DP7	48
DP8	0

Sample 7907 Worksheet, Removable Disc (Unit 1)

LU	29	30	31
DP1	0	0	0
DP2	400B	400B	400B
DP3	0	0	0
DP4	0	0	0
DP5	0	19200	64750
DP6	400	943	320
DP7	48	48	48
DP8	0	0	0

Sample 7908 Worksheet

LU	16	17
DP1 HP-IB Addr.	0	0
DP2 Unit & Vol #	0	1
DP3 Start	0	0
DP4 Block	0	0
DP5 Number*	0	19200
DP6 Tracks	400	943
DP7 Blocks/Track	48	48
DP8 Reserved	0	0

* Starting Blk Add = Previous Start Blk + (# of Trks * Blocks/Trk)

Sample CTD Worksheet

LU	24
DP1 HP-IB Addr.	0
DP2 Unit & Vol #	400B
DP3 Cache Bit, Disc U/V #	100000B
DP4 Start Block	0
DP5 Number	64464
DP6 Address of Disc	0
DP7 Cache Block	0
DP8 Reserved	0

The unit number for the Cartridge Tape Drive is 1, represented in the upper 8 bits in DP2 as 400B. The size of the disc cache is 286 blocks, derived from the difference between 64750 (the maximum number of blocks on the 7908 disc) and 64464, of which only the first 256 blocks are used by the driver.

Sample 7911 Worksheet

LU	20	Range
DP1	0	0-7
DP2	0	0
DP3	0	0
DP4	0	0-1
DP5	27392	0-65535
DP6	352	1-13664
DP7	48	48
DP8	0	0

The double-integer quantity cannot exceed 109824: DP4 = 1 64K block; DP5 = 44288 (109824 - 65536).

The starting block address calculations and the CTD worksheet are the same as those for the 7908 disc.

Sample 7912 Worksheet

LU	22
DP1	0
DP2	0
DP3	0
DP4	3
DP5	20223
DP6	821
DP7	48
DP8	0

The double word quantity cannot exceed 256256: DP4 = 3 64K blocks; DP5 = 59648 (256256 - 196608).

The starting block address calculations and the CTD worksheet are the same as those for the 7908 disc.

Sample 7914 Worksheet

LU	23
DP1	0
DP2	0
DP3	0
DP4	3
DP5	59648
DP6	5413
DP7	48
DP8	0

The double-integer quantity cannot exceed 516096: DP4 = 7 64K blocks; DP5 = 57344 (516096 - 458752).

The starting block address calculations and the CTD worksheet are the same as those for the 7908 disc.

Sample 7933/35 Worksheet

LU	25	26
DP1	0	0
DP2	0	0
DP3	0	0
DP4	7	14
DP5	57344	23968
DP6	8862	13300
DP7	48	48
DP8	0	0

The double-integer quantity cannot exceed 1,579,872: DP4 = 24 64K blocks; DP5 = 7008(1,579,872 - 1,572,864).

The starting block address calculations are the same as those for the 7908 disc.

Sample 7941/7942 Worksheet

LU	19
DP1	0
DP2	0
DP3	0
DP4	1
DP5	14592
DP6	266
DP7	48
DP8	0

The starting block address calculations and the CTD worksheet (7942 only) are the same as those for the 7908 disc.

Sample 7945/7946 Worksheet

LU	21
DP1	0
DP2	0
DP3	0
DP4	1
DP5	44288
DP6	2229
DP7	48
DP8	0

The starting block address calculations and the CTD worksheet (7946 only) are the same as those for the 7908 disc.

Sample 9122D Worksheet

LU	0	1
DP1	0	0
DP2	0	400B
DP3	0	0
DP4	0	0
DP5	0	0
DP6	154	154
DP7	16	16
DP8	0	0

Sample 9133H Floppy Worksheet

LU	21
DP1	0
DP2	400B
DP3	0
DP4	0
DP5	0
DP6	154
DP7	16
DP8	0

Special Considerations for the 7902 Disc Drive

The organization of the 7902 dual flexible disc drive is shown in Figure C-3. The 7902 has two independent drives; each drive has one platter with two surfaces and two heads, 0 (top) and 1 (bottom). Each surface has 67 tracks and each track has 30 blocks. The two sets of heads are distinguished by the switch-selectable unit numbers of the drives.

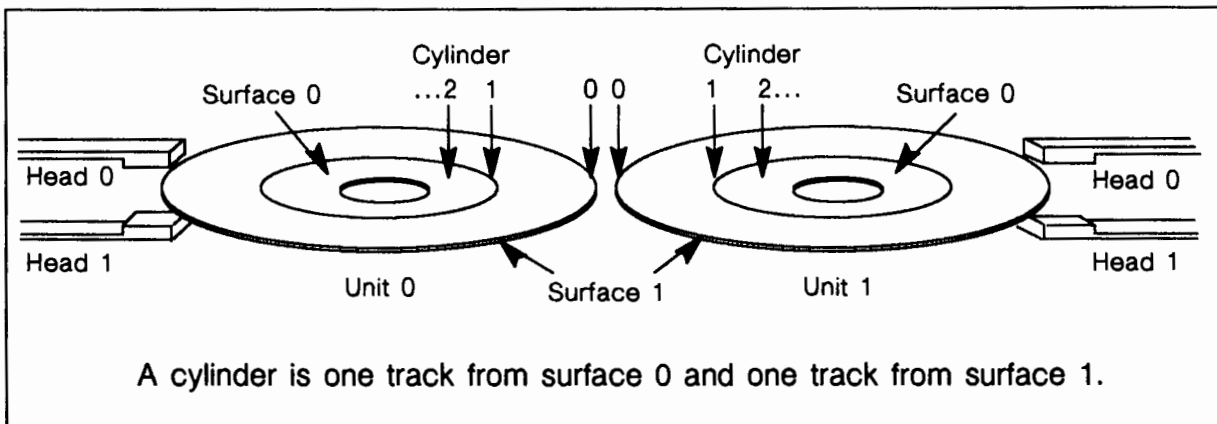


Figure C-3. 7902 Dual Flexible Disc Drive

The 7902 Disc Configuration Worksheet, along with an example of its use, is shown at the end of this section. The top of the 7902 Worksheet represents both surfaces of each of the flexible disc drives. Typically, one logical unit is assigned to each unit of a flexible disc. In this case you can use the defaults provided by the disc driver relocatable file for most of the driver parameters; all you need to supply is the HP-IB address of the 7902 and the unit number of the drive for each LU.

If you want to specify a different configuration, you can diagram your LUs schematically on the 7902 worksheet. Be sure that the tracks and spares are contiguous for each LU, and that the LUs do not overlap. The 7902 must be configured in cylinder mode, and each LU must begin on surface 0. An LU on the 7902 must be completely contained on one drive.

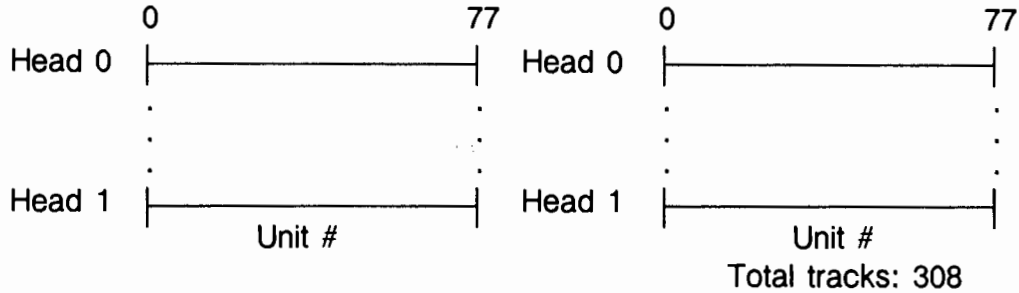
To check to see that you have not overlapped any LUs or wasted any disc space, add up all of the tracks and spares of all the LUs on the disc drive. This number should equal the Total Tracks number shown on the 7902 Worksheet.

You should make one LU out of each of the flexible disc drives of the 7902. Each flexible disc has a capacity of roughly 1 Megabyte, and if that is split the resulting LUs will be quite small. In addition, the convention of one LU per flexible disc is fairly common, so by using this convention you can easily transport your flexible discs between systems.

After you have filled out the 7902 Worksheet, you can transfer the information directly to the driver parameter portion of the IFT/DVT Worksheet.

Sample 7902 Disc Worksheet

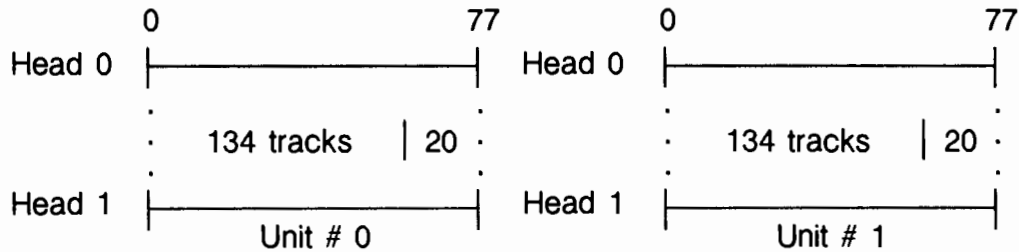
Cylinders:



LU									
DP1 HP-IB Addr.									
DP2 Unit Number									
DP3 Start Head	0	0	0	0	0	0	0	0	0
DP4 Start Cyl									
DP5 Spares	0	0	0	0	0	0	0	0	0
DP6 Tracks									
DP7 Blocks/Track	30	30	30	30	30	30	30	30	30
DP8 Surfaces	2	2	2	2	2	2	2	2	2

7902 Disc Worksheet Sample

Cylinders:



Each unit - 134 tracks used, 20 unused: Total tracks: 308

* Disc LU	20	21							
DP1 HP-IB Addr.	2	2							
DP2 Unit Number	0	1							
DP3 Start Head	0	0	0	0	0	0	0	0	0
DP4 Start Cyl									
DP5 Spares	3	3	0	0	0	0	0	0	0
DP6 Tracks	134	134							
DP7 Blocks/Track	30	30	30	30	30	30	30	30	30
DP8 Surfaces	2	2	2	2	2	2	2	2	2

* Use default file %DD*30,M7902:0, and M7902:1 for DP2 through DP8.

Special Considerations for Model 6 Integrated Disc Drives

On certain HP 1000 computers, flexible mini discs are available. These drives support dual-sided flexible mini discs. Usually two independent disc drives are configured in the same cabinet with a single controller. Each drive has one platter with two surfaces (and two heads). Each surface has 70 tracks, and each track has 16 blocks. Four tracks are reserved as spares and one track is used for the directory, leaving 65 tracks available for user files.

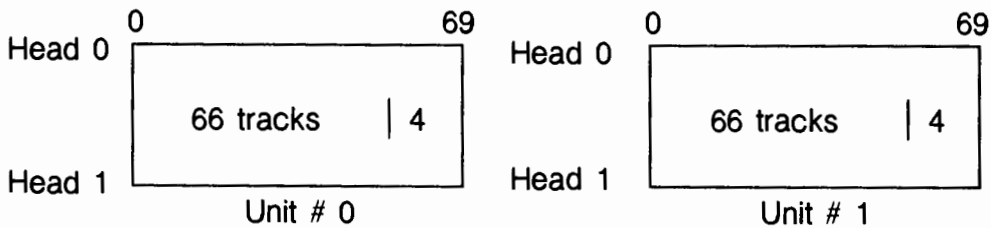
The mini-disc controller for the Model 6 is located in the backplane, and a set of dip switches on this board select the HP-IB address of the two disc units. Mini discs may be daisy-chained from the same HP-IB board that controls the 7902, 7906, 7908, 7911 and 7912 disc drives. The unit numbers of the two mini-disc drives are located on boards just above the drives themselves. Changing these switches requires dismantling the drive unit to a certain extent and should not be performed haphazardly. The left and right drive units are set to 0 and 1 (zero and one) initially, and it is usually not necessary to to change these numbers.

The 16 physical sectors per track of each flexible mini disc correspond to logical blocks. Logical blocks and physical sectors are the same size of 128 words, where one word is two bytes long. 65 tracks multiplied by 16 blocks per track gives 1040 blocks. This is the maximum space available on one HP mini disc.

It is wasteful and cumbersome to try to define more than one logical unit per mini disc. This example shows two disc units, each defined as one LU:

Sample Model 6 Integrated Disc Worksheet

Cylinders:

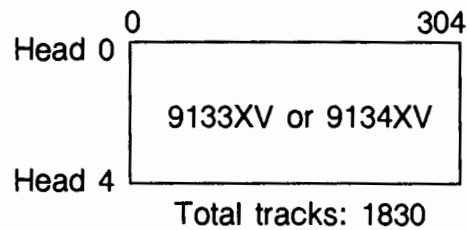
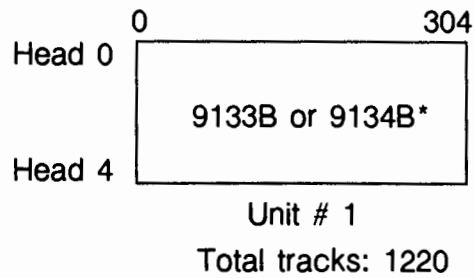
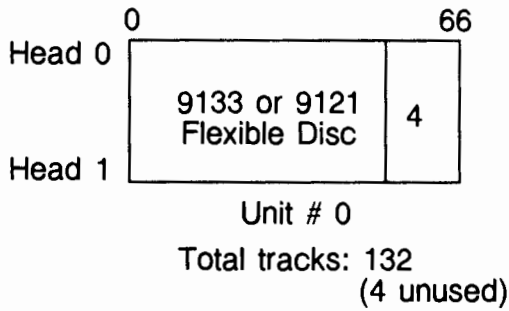


Each unit - 66 tracks used, 4 tracks unused: Total tracks: 140

* Disc LU	32	33						
DP1 HP-IB Addr.	2	2						
DP2 Unit Number	0	1						
DP3 Start Head	0	0						
DP4 Start Cyl	0	0						
DP5 Spares	0	0						
DP6 Tracks	66	66						
DP7 Blocks/Track	16	16						
DP8 Surfaces	2	2						

Sample 9133/9134/9121 Disc Worksheet

Cylinders:



* The only difference between the 9133 and 9134 is the micro floppy drive, which only shares the HP-IB port. It has its own address and appears to the driver as identical to a 9121S.

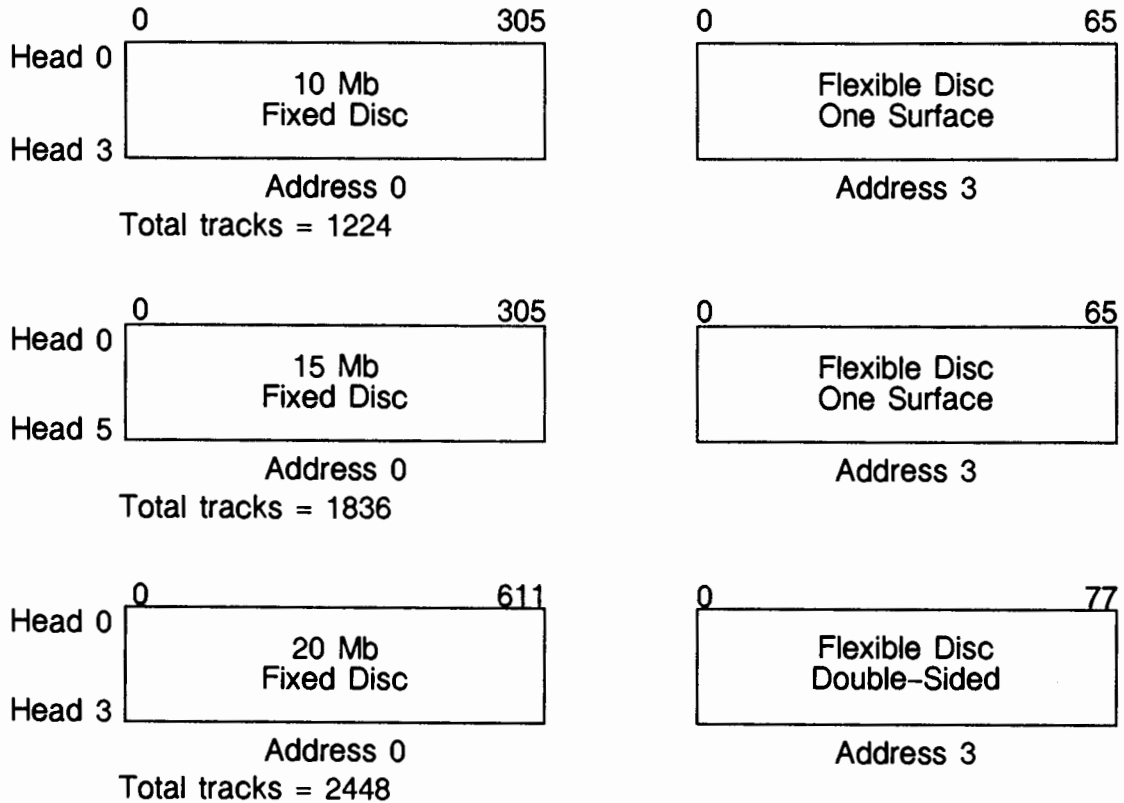
	Flx. 9133 XV/B	9121D		9134A 4-Vol			
DP1 HPIB Address	3	3	3	7	7	7	7
DP2 Unit Number	0	0	1	0	1	2	3
DP3 Start Head	0	0	0	0	0	0	0
DP4 Start Cyl	0	0	0	0	0	0	0
DP5 Spares	0	0	0	3	3	3	3
DP6 Tracks	66	66	66	134	134	134	134
DP7 Blks/Track	16	16	16	30	30	30	30
DP8 Surfaces	2	2	2	2	2	2	2
		9133B/4B Single Volume		9133XV/34XV Single Volume			
DP1 HPIB Address	7	7	7	7	7	7	7
DP2 Unit Number	0	0	0	0	0	0	0
DP3 Start Head	0	0	0	0	0	0	0
DP4 Start Cyl	0	118	152	0	104	159	236
DP5 Spares	0	0	0	0	0	0	0
DP6 Tracks	472	136	612	624	330	462	414
DP7 Blks/Track	31	31	31	31	31	31	31
DP8 Surfaces	4	4	4	6	2	6	6

Microsystems with Integrated Peripherals (248x)

The integrated peripherals options on microsystems provide two disc drives, one fixed and one removable.

Configuration:

Cylinders:



		mf*	10Mb Hard Disc				15Mb Hard Disc				20Mb Hard Disc		mf**
LU		39	36	37	38	55	56	57	58	59	60	61	
DP1	HPIB Address	3	0	0	0	0	0	0	0	0	0	3	
DP2	Unit Number	0	0	0	0	0	0	0	0	0	0	0	
DP3	Start Head	0	0	0	0	0	0	0	0	0	0	0	
DP4	Start Cyl	0	0	118	152	0	104	159	236	0	489	0	
DP5	Spares	0	0	0	0	0	0	0	0	0	0	0	
DP6	Tracks	66	472	136	616	624	330	462	420	1956	492	154	
DP7	Blks/Track	16	31	31	31	31	31	31	31	31	31	16	
DP8	Surfaces	1	4	4	4	6	6	6	6	4	4	2	
DP9	Reserved	0	0	0	0	0	0	0	0	0	0	0	

* Single-sided microfloppy

** Double-Sided microfloppy

Address and Unit Specification on Microsystems

You can select between different ports and drives on the Disc Interface card (12022A). First, the drive Address (DP1) allows you to select between microfloppies, or one of two hard discs. Each microfloppy has its own port, both of which are Address 3. The Unit number (DP2) selects microfloppy unit 0 or unit 1. Note that only one microfloppy, unit 0, is currently provided for the microsystem.

Hard discs have no Unit specifications, and thus this parameter is set to zero. Hard discs are selected by Address 0 or 1. The proper address parameter is factory set and should never be changed.

Special Considerations for the 7906M Disc Drive

The 7906M disc drive has two platters for data storage. One of these platters (the top one) is removable; the other one is fixed in the drive and may not be removed. Each of these platters has two surfaces, with 411 tracks per surface and 24 blocks per track.

The organization of the surfaces and heads of the 7906M disc drive for generation purposes is shown in Figure C-4. The four surfaces available for data storage are numbered 0, 1, 2, and 3. Each surface is accessed by one head, also numbered 0, 1, 2, and 3.

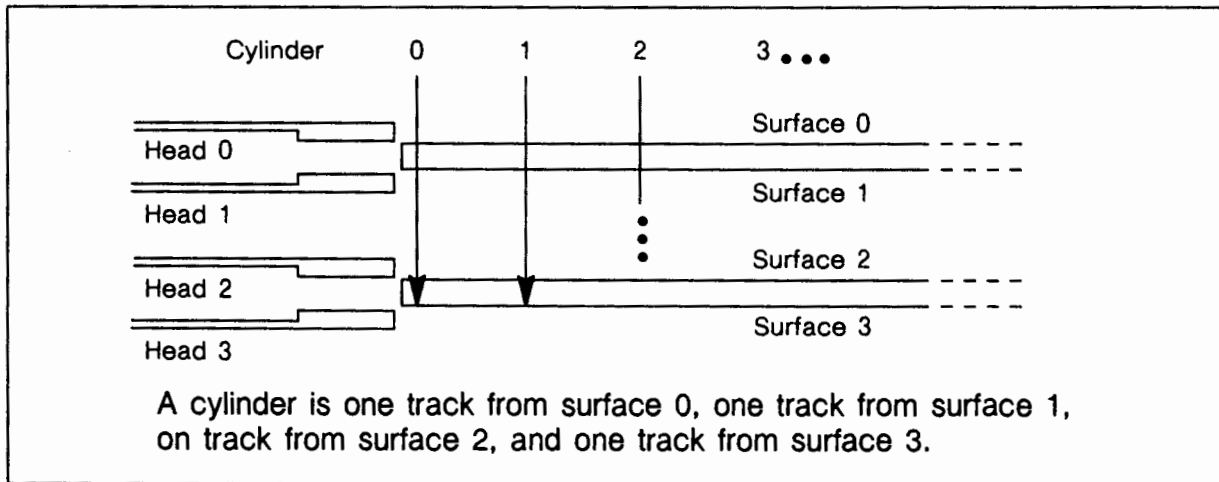


Figure C-4. 7906M Disc Drive

The 7906M disc drive may be divided into several LUs. There are three modes in which the 7906M LU can operate: surface, cylinder, or shortened cylinder. Surface mode uses only one surface per LU, cylinder mode uses all four surfaces for each LU, and shortened cylinder typically uses two surfaces per LU.

Surface mode is recommended with the 7906M disc drive. Due to the physical structure of the disc drive, there is no advantage in access time when cylinder mode is used. There is, however, a significant disadvantage to using cylinder mode. Cylinder mode requires that LUs span all four surfaces of the drive, two of which are removable. When the removable platter (surfaces 0 and 1) is removed and replaced with another platter, the LUs (which cover all four surfaces) will contain mixed data from the old fixed platter and the new removable platter. The results may be interesting but not usable. Surface mode avoids such problems, since LUs in surface mode do not cross surface boundaries. If you replace the top platter, the information on the bottom platter is undisturbed. Further, the VCP ROMs will only boot a system from a 7906M if it is in surface mode.

LU integrity is also preserved in shortened cylinder mode, because the cylinders include only the two sides of either the fixed or the removable platter. Better performance can be expected in surface mode than in shortened cylinder mode, but if your need is to read data that has been stored in shortened cylinder mode, you can configure the drive for that purpose. To configure the disc in shortened cylinder mode, use device type 34B (DT:34B in the DVT statement), and specify driver DDM30 (%DDM30).

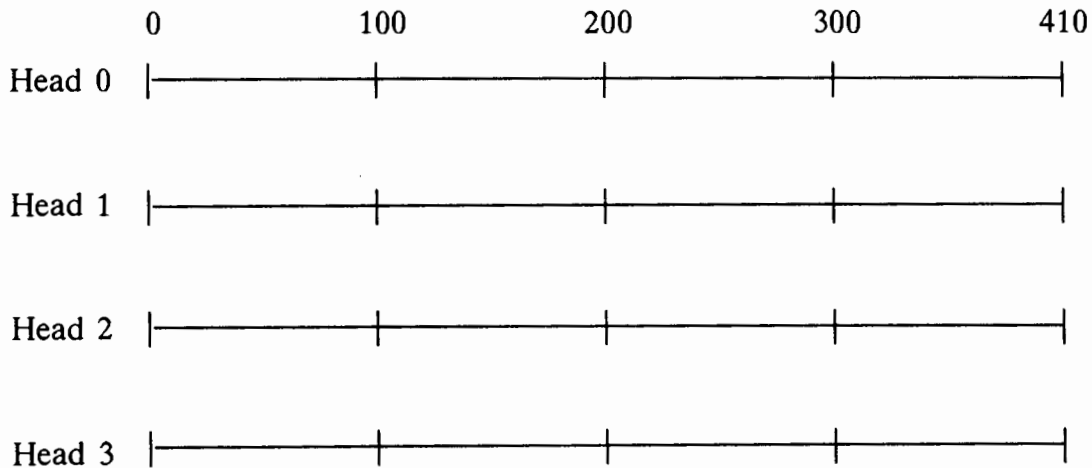
While up to 31 LUs may be defined for one 7906M disc drive, a good plan is to define no more than six. The total number of tracks on a 7906M is 1644. Of these, you should use approximately 1600 to make up your LUs and distribute the remaining tracks over the LUs as spares. The FORMT disc utility program lets you use spare tracks to replace bad tracks. FORMT is described in the RTE-A Utilities Reference manual.

Using the Worksheet

The 7906M Disc Configuration Worksheet, along with an example of its use, is shown at the end of this section. The worksheet contains a schematic diagram of the four surfaces of the 7906M drive. Use the worksheet to map the drive in such a way that the tracks and spares are contiguous for each LU and the tracks and spares of one LU do not overlap those of another LU. Using the rough rule of thumb of allowing 2 or 3 spare tracks per 100 data tracks, allocate the tracks to the LUs as desired and fill in the driver parameters to match. You can then use this information to fill in the driver parameter portion of the IFT/DVT Worksheet.

7906M Disc Configuration Worksheet Cylinders:

Cylinders:



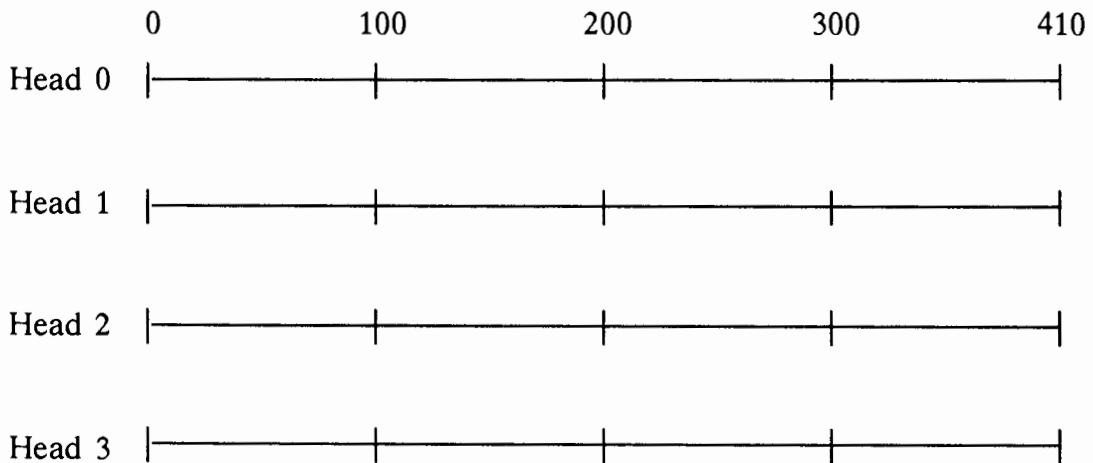
Total Tracks: 1644

Disc LU	20	21							
DP1 HP-IB Addr.									
DP2 Unit Number	0	0	0	0	0	0	0	0	0
DP3 Start Head									
DP4 Start Cyl									
DP5 Spares									
DP6 Tracks									
DP7 Blocks/Track	48	48	48	48	48	48	48	48	48
DP8 Surfaces									

EXAMPLE

7906 Disc Configuration Worksheet

Cylinders:



Total Tracks: 1644

Disc LU	20	21							
DP1 HP-IB Addr.									
DP2 Unit Number	0	0	0	0	0	0	0	0	0
DP3 Start Head									
DP4 Start Cyl									
DP5 Spares									
DP6 Tracks									
DP7 Blocks/Track	48	48	48	48	48	48	48	48	48
DP8 Surfaces									

Special Considerations for MAC Drives

MAC (Multiple Access Controller) disc drives include the 7906M, 7920M and 7925M. Up to eight of these drives can be connected to one controller. Connecting a multi-access controller requires some special considerations at system generation.

Assign a unique HP-IB address to the controller and set the switch on the controller to that address. Assign a unique unit number, from 0 to 7, for each disc drive connected to the controller and set the unit selector switch on each drive to the number you assign to that drive. Finally, list all the LUs configured for a given controller in the same NODE statement.

Figure C-5 shows the form of the generator answer file statements for a single-CPU MAC disc generation setting up three discs on one controller. The configuration uses all default file values except the HP-IB address (DP1) and the unit number (DP2).

If a MAC drive is to be the system disc, it must be assigned as unit 0. It is not possible to boot from any other unit number. Note, however, that it is possible to boot from any head of a 7906 disc as long as the disc is assigned as unit 0 and is configured in surface mode.

If two or three CPUs are to be connected to this controller, the answer files for all CPUs must specify the same track mapping and the device type must be explicitly stated as DT:35B in each of the DVT statements. For example:

```
DVT,%DDM30::18,M7906:0,LU:40,DT:35B,DP:1:2:0
```

The device type in Figure C-5 is the default, 32B. All defaults are given in the tables of default file values in Appendix E.

```

*
* HP-IB #1
*
IFT,%ID*37::18,SC:27B
*
* HP-IB #1 BUS LU
*
DVT, , ,LU:9,TO:2000,DT:77B,TX:0,DX:1,DP:1:36B,PR:0
*
*           MAC DISCS           LU 40-62           HP-IB ADDRESS 2
*
*           MAC 7906 LU 40-43 UNIT 0
*
DVT,%DDM30::18,M7906:0,LU:40,DP:1:2:0
DVT,...           ....:1,LU:41,...
DVT,...           ....:2,LU:42,...
DVT,...           ....:3,LU:43,...
*
*           MAC 7925 LU 44-52 UNIT 1
*
DVT,%DDM30::18,M7925:0,LU:44,DP:1:2:1
DVT,...           ....:1,LU:45,...
DVT,...           ....:2,LU:46,...
DVT,...           ....:3,LU:47,...
DVT,...           ....:4,LU:48,...
DVT,...           ....:5,LU:49,...
DVT,...           ....:6,LU:50,...
DVT,...           ....:7,LU:51,...
DVT,...           ....:8,LU:52,...
*
*           MAC 7920 LU 53-62 UNIT 2
*
DVT,%DDM30::18,M7920:0,LU:53,DP:1:2:2
DVT,...           ....:1,LU:54,...
DVT,...           ....:2,LU:55,...
DVT,...           ....:3,LU:56,...
DVT,...           ....:4,LU:57,...
DVT,...           ....:5,LU:58,...
DVT,...           ....:6,LU:59,...
DVT,...           ....:7,LU:60,...
DVT,...           ....:8,LU:61,...
DVT,...           ....:9,LU:62,...
.
.
NODE 40,41,42,43,44,45,46,47,48,49-
      50,51,52,53,54,55,56,57,58,59,60,61,62
.

```

Figure C-5. Example MAC Disc Generator Answer File Statements



GEN Records

GEN records are relocatable records used to pass information to RTAGN. The information can be used to augment commands contained in the generation answer file. The term GEN is an assembler pseudo opcode used to identify the following record as generation instructions to be assembled into relocatable code.

The format of a GEN record is:

GEN n,<ASCII string>

where

n is the number of words in the ASCII string (2 ASCII characters per word).

ASCII string is the information to be included in the relocatable record.

There are two major types of GEN records recognized by RTAGN – driver parameter default information records and system partitioning information records.

Parameter Default Records

Default files are used to supply IFT and DVT parameters during generation. They are relocatable files that contain GEN records with the appropriate parameter information. Typically, these GEN records are contained in the file of relocatable code for a driver, but if you want to establish different default parameter values you can construct your own default file.

When the generator searches a file for default values, it ignores everything but GEN records. Thus, your default file does not have to contain any information other than GEN records before it executes the running commands in the command string. Because the GEN records are ignored when the code for a driver is relocated, a driver file can contain GEN records and be used as a default file.

The ASCII string contains IFT or DVT parameter strings in the same format as used in the generator answer file. Continuation lines are not allowed in GEN records.

Processing

When a default file is specified in an IFT or DVT command string (for example, IFT,%ID*00 . . .), the generator searches that file for GEN records before it executes the remaining commands in the command string. The generator uses the information in these records to set the default values for the specified parameters. If a parameter is specified more than once, the last value specified is used.

The generator uses the following method to decide which GEN records to use:

1. If the model number is not specified in the IFT or DVT command string, the generator reads and uses all GEN records until:
 - a. a GEN record containing a model number is encountered, or
 - b. no more GEN records are found.Any remaining GEN records are ignored.
2. If a model number is specified in a DVT command string only, the generator reads and uses all GEN records as above. After the first model number is encountered, the generator skips all subsequent GEN records until it encounters a model number that matches the one specified in the command string. After the match is made, the generator reads and uses all GEN record information until the next model number is read. Any GEN records after that are ignored.

Note that for a model number to match, all model subparameters must also match. That is, M2645 is not the same as M2645:1.

This method of processing GEN record information allows one default file to contain information for several different devices and several different LUs on each device. For example, the file for driver DD*30 contains default values for the LU configuration of the 7902 and the 7906 disc drives.

After the generator retrieves the default values from the default file, it then finishes execution of the command string from the generator command file. This allows the commands in the answer file to override any default values.

Sample Default File

This section shows the format of a default file. The MACRO, NAM, and END statements must be included. The lines beginning with an asterisk (*) are comment lines. Before it can be used, this file must be assembled by the Macro Assembler.

Format of the Default GEN Record

The format of the GEN record in a device driver is given in two sections. The first section gives general information that can be applied to all devices controlled by the given driver. The second section includes information specific for the given device driver. This section must start on a new line and begin with a model number. Any parameters needed for the specific device follow the model number. Additional model numbers and parameters always begin on a new line.

MACRO,L,R
NAM D7906

*

* Start of General Defaults

*

GEN 25,EDD.30,BL:UN:100: 200, TX:32
GEN 25,DX: 8, PR: 2, TO: 200

*

* Start of Model Number Defaults

* Begin defaults for first logical unit

*

GEN 25,M7906:0, DT: 31
GEN 25,DP: 2: 0: 0: 0: 4: 202, DP: 7: 48: 1

*

* Begin defaults for second logical unit

*

GEN 25,M7906:1, DT: 31
GEN 25,DP: 2: 0: 0: 206: 3: 202, DP: 7: 48: 1

*

* Begin defaults for third logical unit

*

GEN 25,M7906:2, DT: 31
GEN 25,DP: 2: 0: 1: 0: 2: 202, DP: 7: 48: 1

*

* Begin defaults for fourth logical unit

*

GEN 25,M7906:3, DT: 31
GEN 25,DP: 2: 0: 1: 204: 5: 202, DP: 7: 48: 1

*

* Begin defaults for fifth logical unit

*

GEN 25,M7906:4, DT: 31
GEN 25,DP: 2: 0: 2: 0: 3: 202, DP: 7: 48: 1

*

* Begin defaults for sixth logical unit

*

GEN 25,M7906:5, DT: 31
GEN 25,DP: 2: 0: 2: 205: 4: 202, DP: 7: 48: 1

*

* Begin defaults for seventh logical unit

*

GEN 25,M7906:6, DT: 31
GEN 25,DP: 2: 0: 3: 0: 11: 400, DP: 7: 48: 1

*

* End of defaults for the 7906 disc lus

*

END



System Partitioning Records

System partitioning information records provide the generator with the information it needs to properly relocate system modules into the OS/driver partition area. There are two basic types of system partitioning information records: entry records and module flag records.

When a system module that contains system partitioning information records is relocated in the non-partitioned portion of system memory, the GEN records are ignored. The information is not needed.

All partitionable system modules provided by Hewlett-Packard contain the necessary GEN records. In general, knowledge about system partitioning information records will only be useful for creating non-HP custom partitionable system modules.

Processing

Entry Records

For each entry to a partitioned module, a special interface routine called a “tag” must be constructed. The tag calls a system routine that prepares for entry into the module by mapping in the needed system partition.

When RTAGN finds a system partitioning entry information GEN record, it allocates space in the tag area for the necessary tag. The length and contents of the tag are determined from the information on the record, as described in the Format section of this appendix.

Several errors can occur in processing system partitioning entry records. Records that do not follow the proper format are flagged as errors, as well as entries that occur on entry GEN records but do not have corresponding ENT records. A third error occurs when the relocatable file does not adhere to the normal record order observed by Hewlett-Packard language processors: all ENT records must precede all DBL records. (HP relocatable record formats are described in the MACRO reference manual.)

Module Flag Records

System partitioning module flag records are used to create module flags for partitionable system modules. (See the RTE-A System Design manual for information about module flags.) Partitionable modules cannot use the normal mechanism for creating module flags.

When RTAGN finds a system partitioning module flag record, it allocates one word in the tag area and stores the specified value in it. When the module flag is referenced, the value from the tag area will be accessed.

Format of System Partitioning GEN Records

Entry Records

The format of a system partitioning entry record is:

GEN <n>,OS,EN,<symbol>:parm count>:<entry type>, . . .

where

- n is the number of words.
- OS,EN specifies a system partitioning entry record.
- symbol is the entry point symbol in the module (i.e., \$EX22 is an entry point in the MEMRY module).
- parm count is the number of parameters in the calling sequence of the entry point symbol, used for JSB type entries only. This number must be 0, 1, or 2; for non-JSB type entries, the number must be 0.
- entry type is the type of call or entry. Valid types are JSB (subroutine call), JMP (jump to routine) and TC (entry on interrupt via trap cell).

Module Flag Records

The format of a system partitioning module flag record is:

GEN <n>,MF,<symbol>:<value>, . . .

where:

- n is the number of words.
- MF identifies the record as a system partitioning module flag record.
- symbol is the name of the module flag (i.e., \$CDS.. is a module flag for the CDSFH module).
- value is the desired value for the flag, either 0 ("real" module) or -1 ("dummy module"). It is not recommended that dummy modules be made partitionable; therefore, MF records for dummy modules are not necessary.

Device and Interface Driver Tables

This appendix contains information that describes supported disc drives, and tables that show the device driver correspondence (Table E-1), the standard entries for the answer file (Table E-2), and the default device file values (Table E-3 through E-22). Note that the loaded driver size may vary, depending on your system requirements.

The convention HP uses to refer to driver names has been changed from DD.nn to DD*nn. The HP driver relocatable filenames of the form %DD.nn have also been changed from %DD.nn to %DD*nn. HP driver names in the NAM statement (DD.nn) and driver entry points (DD.nn) remain the same.

DD*nn	Referenced driver name
%DD*nn	Driver relocatable name
DD.nn	Driver name in NAM statement
DD.nn	Driver entry point

The supported disc drives are listed below, in the order of their capacity:

CS/80, SS/80:	7908P/R	- 16Mbyte	(fixed disc)
	7941A	- 24Mbyte	(fixed disc without CTD)
	7942A	- 24Mbyte	(fixed disc with CTD)
	7911P/R	- 28Mbyte	(fixed disc)
	7907A	- 21/21Mbyte	(one fixed, one removable disc)
	7945A	- 55Mbyte	(fixed disc without CTD)
	7946A	- 55Mbyte	(fixed disc with CTD)
	7912P/R	- 65Mbyte	(fixed disc)
	7957A	- 82Mbyte	(fixed disc)
	7958A	- 130Mbyte	(fixed disc)
	7914P/R	- 132Mbyte	(fixed disc with/without CTD)
	7914CT	- 132Mbyte	(fixed disc with CTD)
	7914ST	- 132Mbyte	(fixed disc with 7974A tape unit)
	7936A	- 300Mbyte	(fixed disc)
	7933A	- 404Mbyte	(fixed disc)
	7935A	- 404Mbyte	(removable disc)
	7937A	- 571Mbyte	(fixed disc)
	MAC:	7906M/S	- 19/10Mbyte
7920M/S		- 50Mbyte	(removable disc)
7925M/S		- 120Mbyte	(removable disc)

Compact:	248x*	-	10Mbyte	(integral fixed)
		-	20Mbyte	(integral fixed)
		-	270Kbyte	(floppy)
		-	630Kbyte	(dual-sided floppy)
	9121S	-	270Kbyte	(floppy)
	9121D	-	2x270Kbyte	(floppy)
	9122D	-	2x630Kbyte	(two dual-sided floppies)
	9133XV	-	15Mbyte	(fixed)
		-	270Kbyte	(floppy)
	9133D	-	15Mbyte	(fixed)
		-	630Kbyte	(floppy)
	9133H	-	20Mbyte	(fixed)
		-	630Kbyte	(floppy)
	9133L	-	40Mbyte	(fixed)
		-	630Kbyte	(floppy)
	9134D	-	15Mbyte	(fixed)
	9134H	-	20Mbyte	(fixed)
	9134L	-	40Mbyte	(fixed)
	9153B	-	20Mbyte	(fixed)
	12122A*	-	20Mbyte	(fixed)

* Micro/1000 only

CS/80 Configuration Scheme

The standard configuration for CS/80 discs is based on a building block scheme in which all CS/80 discs are subdivided into logical units of the same size. The smallest capacity disc, the 16-Mbyte Model 7908, is divided into two units, LU 16 and LU 17, plus LU 24 for the CTD cache. The next larger-capacity disc (the 20-Mbyte Model 7907) uses all of the default values for LUs 16, 17, and 24, plus the values for its additional disc logical units. Each larger-capacity disc then uses all of the default values for the next smaller disc (including LU 24 if the CTD is a part of the disc), plus the file values for each additional disc logical unit. Appendix K contains the procedure for converting to this standard HP CS/80 disc configuration.

Table E-1. Device/Driver Correspondence

HARDWARE CATEGORY	PART NUMBER	DRIVER	APPROX. DVR SIZE
TERMINALS	2392A/2393A/2397A 262x A/P (Includes 2382) 2635 2645/264x 700/41*	DDC00 DDC00 DDC00 DDC00 DDC00	1056
CASSETTE TAPE UNITS	2645/264x	DD*20	697
PRINTERS	2225D (Using RS232 MUX) 2227A (Using RS232 MUX) 2228A (Using RS232 MUX) 2235D (Using HP-IB) 256x (Using HP-IB) 256x (Using RS232 MUX) 2608S/2563A 2631 (Using ASIC) 2631 (Using HP-IB) 2635 (Using ASIC) 2687A (Using HP-IB) 2932A (Using ASIC) 2932A (Using HP-IB) 2933A (Using ASIC) 2933A (Using HP-IB) 2934A (Using ASIC) 2934A (Using HP-IB)	DDC00 DDC00 DDC00 DD*12 DDC12 DDC00 DDC12 DDC00 DD*12 DD*12 DDC00 DDC00 DD*12 DD*12 DDC00 DD*12 DDC00 DD*12	1563 702
DISCS	248x 7902 Flexible Disc 9121 Flexible Micro Discs 9122 Flexible Micro Discs 9133A/V/XV 9134 5-1/4" Winchester Disc 9153B 20 MB Disc 9895A Flexible Mini Discs 7906M, 7920M, 7925M	DD*27 DD*30 DD*30 DD*33 DD*30 DD*30 DD*33 DD*30 DDM30	537 1023

* The 700/41 low-cost terminal does not support CI stack or EDIT screen mode. It will not work with RTE-A utilities and therefore cannot be used as a console or boot terminal. The 700/41 should be connected to the MUX only.

Table E-1. Device/Driver Correspondence (Cont.)

HARDWARE CATEGORY	PART NUMBER	DRIVER	APPROX. DVR SIZE
CS/80 DEVICES	7907	DD*33	1696
	7908	DD*33	
	7911	DD*33	
	7912	DD*33	
	7914	DD*33	
	7933	DD*33	
	7935	DD*33	
	7941/7942	DD*33	
	7945/7946	DD*33	
	CTD (integrated)	DD*33	
	9133D/H/L	DD*33	
	9134D/H/L	DD*33	
	9144 CTD (Standalone)	DD*33	
MAGNETIC TAPE	7970E HP-IB Mag Tape	DD*23	590 1095
	7974A HP-IB Mag Tape	DD*24	
	7978A HP-IB Mag Tape	DD*24	
	7978B HP-IB Mag Tape	DD*24	
	7979/80 HP-IB Mag Tape	DD*24	
INTERFACE CARDS	12005 Async. Serial Interface	ID*00/01	473/1003 136 1448 190 256 368 887 1381 1484
	12008A Prom storage module	ID*36	
	12009 HP-IB Interface	ID*37	
	12153 Writable Control Store	ID*41	
	12205A A900 WCS	ID*42	
	12006 Parallel Interface	ID*50	
	12022A Int Disc Interface	ID*27	
	12040 RS232 Multiplexer	IDM00	
	12044 DS/1000 Dir Connect	ID*66	
PLOTTERS	7440 (Using HP-IB)	ID*37	
	7440 (Using RS232 MUX)	DD*00	
	7470 (Using HP-IB)	ID*37	
	7470 (Using RS232 MUX)	DD*00	
	7475 (Using HP-IB)	ID*37	
	7475 (Using RS232 MUX)	DD*00	
	7550 (Using HP-IB)	ID*37	
	7550 (Using RS232 MUX)	DD*00	
	7580 (Using HP-IB)	ID*37	
	7580 (Using RS232 MUX)	DD*00	
	7585 (Using HP-IB)	ID*37	
	7585 (Using RS232 MUX)	DD*00	
	7586 (Using HP-IB)	ID*37	
	7586 (Using RS 232 MUX)	DD*00	
7595/6 (Using HP-IB)	ID*37		
7595/6 (Using RS 232 MUX)	DD*00		

ration Phase DVT Entries	Node List Entries
lu J:lu I:lu	NODE, cartridge 0 lu, ..., cartridge 3 lu NODE, cartridge 3 lu, cartridge 4 lu, cartridge 13 lu NODE, cartridge 11 lu, cartridge 12 lu, cartridge 14 lu
P:1:20004B, TX:57, DP:5:CI:20040B:0, DP:9:CM:20040B:20040B:CM P:5:CI:20040B:0, DP:9:CM:20040B:CM I:1:20004B, TX:57, DP:5:CI:20040B:0, DP:9:CM:20040B:20040B:CM P:5:CI:20040B:0, DP:9:CM:20040B:20040B:CM xxx = 2392 or 262x) J (:1 = printer) U:lu I:1:2004B, TO:3000	NODE, lu of term DVT, lu of printer DVT
/2, LU:lu (:1=left CTU, :2=rht CTU)	NODE, lu of term DVT[, lu of left CTU [, lu of right CTU]]
I:1:2B, TX:57 I:1:2B, TX:57 I:1:2B, TX:57 I:1:2B, TX:57 IT:12B, DP:1:6 B address I:1:000002B, TX:57 I:1:HPIB address I:1:000002B, TX:57** IP:1:HPIB, address	
DT:77B, DX:1, DP:1:9 XX:2, DP:1:2, DT:45B DT:77B, DX:1, DP:1:5	
I:1:20004B, DP:5:CI:20040B, TX:57 DT:77B, DX:1:HPIB address DT:77B, DX:1, DP:1:5 I:1:2004B, TO:3000 I:1:20004B, DP:5:CI:20040B, TX:57 DT:77B, DX:1:HPIB address DT:77B, DX:1, DP:1:5 I:1:2004B, TO:3000 I:1:20004B, DP:5:CI:20040B, TX:57 DT:77B, DX:1:HPIB address I:1:20004B, DP:5:CI:20040B, TX:57 DT:77B, DX:1:HPIB address I:1:2004B, TO:3000 DT:77B, DX:1, DP:1:5	
I (:0=left disc, :1=rht disc) .U:lu, DP:1:HPIB address /9, LU:lu, DP:1:HPIB address /8, LU:lu, DP:1:HPIB address .U:lu, DP:1:HPIB address /9, LU:lu, DP:1:HPIB address /8, LU:lu, DP:1:HPIB address IT:33B, TX:72, DX:8, DP:1:0, PR:0 IT:33B, TX:72, DX:8, DP:1:0, PR:0	NODE, cartridge 0 lu, cartridge 1 lu NODE, cartridge 0 lu, cartridge 1 lu, cartridge 2 lu, cartridge 3 lu NODE, cartridge 0 lu, cartridge 1 lu, ..., cartridge 9 lu NODE, cartridge 0 lu, cartridge 1 lu, ..., cartridge 8 lu NODE, cartridge 0 lu, cartridge 1 lu, cartridge 2 lu, cartridge 3 lu NODE, cartridge 0 lu, cartridge 1 lu, ..., cartridge 9 lu NODE, cartridge 0 lu, cartridge 1 lu, ..., cartridge 8 lu NODE, cartridge 0 lu, ..., cartridge 7 lu NODE, cartridge 0 lu, ..., cartridge 9 lu

Model Number		IF DVR	Modules Relocated		Table Generation Phase IFT Entries	Table Gene
2480	10Mb Integrated Discs	ID*27	%ID*27		IFT, %ID.27, SC:sc	DVT, %GEN27, M2480:0/1/2/3, LU:
2480	15Mb Integrated Discs	ID*27	%ID*27		"	DVT, %GEN27, M2480:3/4/5/6/7, L
2480	20Mb Integrated Discs	ID*27	%ID*27		"	DVT, %GEN27, M2480:11/12/14, LL
2392A/2393A	(Using RS232 MUX) (Note A)	IDM00	%DD*00	%IDM00	IFT, name, SC:sc, TX:20	DVT, name, M26xx, LU:lu, DF
2392A/2393A	(Using RS232 ASIC)	ID*00	%DD*00	%ID*00	" , SC:sc	" , M26xx, LU:lu, DI
2397A	(Using RS232 MUX) (Note A)	IDM00	%DD*00	%ID*00	" , SC:sc, TX:20	" , M26xx, LU:lu, DF
2397A	(Using RS232 ASIC)	ID*00	%DD*00	%ID*00/01	" , SC:sc	" , M26xx, LU:lu, DI
262x	A/P (Includes 2382)	ID*00/01	%DD*00	%IDM00 *	" "	" , M2xxx, LU:lu (2
2635		ID*00/01	%DD*00	%ID*00/01"	" "	" , M2635:0/1, LU:l
2645	Terminal	ID*00/01	%DD*00	%ID*00/01"	" "	" , M2645/M264x, L
700/41	Terminal (Note C)	IDM00	%DD*00	%IDM00	" , SC:sc, TX:20	DVT, name, M755x, LU:lu, DF
2645/264x	Cassette Tape Unit	ID*00/01	%DD*20		IFT, name, SC:sc	DVT, name, M2645/M264x:1
2225D	(Using RS232 MUX) (Note A)	IDM00	%DD*00	%IDM00	IFT, name, SC:sc, TX:20	DVT, name, M2225, LU:lu, DF
2227A	(With 12040 MUX) (Note A)	IDM00	%DD*00	%IDM00	" "	" , M2631, LU:lu, DF
2227A	(With 12100A OBIO)	ID400	DDC00 ID400, REL	DDC01	" "	" , M2631, LU:lu, DF
2228A	(With 12040 MUX) (Note A)	IDM00	%DD*00	%IDM00	" "	" , M2631, LU:lu, DF
2235D	(Using HP-IB)	ID*37	%DD*12	%ID*37	IFT, name, SC:sc	DVT, name, M2932A, LU:lu, C
256x	(Using HP-IB)	ID*37	%DDC12	%ID*37	" , SC:sc	DVT, %DDC12, , LU:lu, DP:1:HP
256x	(Using RS232 MUX) (Note A)	IDM00	%DD*00	%IDM00	" "	DVT, name, M256x, LU:lu, DF
2631	(Using ASIC), 2932A/33A/34A	ID*00/01	%DD*00	%ID*00/01"	" "	" , M2631, LU:lu
2631	(Using HP-IB), 2932A/33A/34A	ID*37	%DD*12	%ID*37	" "	" , M2631, LU:lu, DF
2687A	(Using RS232 MUX) (Note A)	IDM00	%DD*00	%IDM00	" "	" , M2631, LU:lu, DF
2608S		ID*37	%DDC12	%ID*37	" "	" , M2608S, LU:lu, C
3852A	(Using HP-IB)	ID*37	%ID*37		IFT, name, SC:sc	DVT, name, , LU:lu, TO:500, I
3852A	(GPIO)	ID*50	%ID*50		" "	" , , LU:lu, TO:500, I
35401	(Using HP-IB)	ID*37	%ID*37		" "	" , , LU:lu, TO:500, I
7440	(Using RS232 MUX) (Note A)	IDM00	%DD*00	%IDM00	IFT, name, SC:sc, TX:20	DVT, name, M744x, LU:lu, DF
7440	(Using HP-IB)	ID*37	%ID*37		" , SC:sc	DVT, , LU:lu, TO:500, I
7510A	(Using HP-IB)	ID*37	%ID*37		" , SC:sc	DVT, name, , LU:lu, TO:500, I
7510A	(With 12040 MUX) (Note A)	IDM00	%DD*00	%IDM00	" , SC:sc, TX:20	" , M755x, LU:lu, DF
7550	(Using RS232 MUX) (Note A)	IDM00	%DD*00	%IDM00	" , SC:sc, TX:20	" , M755x, LU:lu, DF
7550	(Using HP-IB)	ID*37	%ID*37		" "	" , LU:lu, TO:500, I
7570A	(Using HP-IB)	ID*37	%ID*37		" "	" , LU:lu, TO:500, I
7570A	(With 12040 MUX) (Note A)	IDM00	%DD*00	%IDM00	" , SC:sc	" , M755x, LU:lu, DF
7585	(Using RS232 MUX) (Note A)	IDM00	%DD*00	%IDM00	" , SC:sc, TX:20	" , M758x, LU:lu, DF
7585	(Using HP-IB)	ID*37	%ID*37		" , SC:sc	" , , LU:lu, TO:500, I
7586	(Using RS232 MUX) (Note A)	IDM00	%DD*00	%IDM00	" , SC:sc, TX:20	" , M758x, LU:lu, DF
7586	(Using HP-IB)	ID*37	%ID*37		" , SC:sc	" , , LU:lu, TO:500, I
7595/6	(Using RS232 MUX) (Note A)	IDM00	%DD*00	%IDM00	IFT, name, SC:sc, TX:20	DVT, name, M755x, LU:lu, DF
7595/6	(Using HP-IB)	ID*37	%ID*37		IFT, name, SC:sc	DVT, name, , LU:lu, TO:500, I
7902	Flexible Disc	ID*33	%DD*30	%ID*37	IFT, name, SC:sc	DVT, name, M7902:0/1, LU:l
7906H		ID*33	%DD*30	%ID*37	" "	" , M7906:0/1/2/3, l
7920H		ID*37	%DD*30	%ID*37	" "	" , M7920:0/1/.../8
7925H		ID*37	%DD*30	%ID*37	" "	" , M7925:0/1/.../7
7906M/S	20Mb (1 fixed, 1 removable disc)	ID*37	%DDM30	%ID*37	" "	" , M7906:0/1/2/3, l
7920M/S	50Mb (removable disc)	ID*37	%DDM30	%ID*37	" "	" , M7920:0/1/.../8
7925M/S	120Mb (removable disc)	ID*37	%DDM30	%ID*37	" "	" , M7925:0/1/.../7
7936	(Using HP-IB)	ID*37	%ID*37		" "	" , , LU:lu, TO:500, I
7937	(Using HP-IB)	ID*37	%ID*37		" "	" , , LU:lu, TO:500, I

eneration Phase DVT Entries	Node List Entries
<p>:500,DT:33B,TX:72,DX:8,DP:1:0,PR:0 :500,DT:33B,TC:72,DX:8,DP:1:0,PR:0 Y_CF:0/1,LU:lu,DP:1:HPIB address :0/1,LU:lu,DP:1:HPIB address :0/1/2,LU:lu,DP:1:HPIB address :0/1/2/3,LU:lu,DP:1:HPIB address Y_CF:1,LU:lu,DP:1:HPIB address 1/2/3/4,LU:lu,DP:1:HPIB address /2/3,LU:lu,DP:1:HPIB address :0/1/2,LU:lu,DP:1:HPIB address ,LU:lu,DP:5:0:66:16:2</p>	<p>NODE, cartridge 0 lu, ..., cartridge 7 lu NODE, cartridge 0 lu, ..., cartridge 5 lu NODE, cartridge 0 lu, cartridge 1 lu NODE, cartridge 0 lu, cartridge 1 lu, microfloppy lu NODE, cartridge 0 lu, ..., cartridge 2 lu, microfloppy lu NODE, cartridge 0 lu, ..., cartridge 3 lu, microfloppy lu NODE, (put microfloppy lu into disc node list) NODE, cartridge 0 lu, cartridge 1 lu, cartridge 2 lu, cartridge 3 lu NODE, cartridge 0 lu, cartridge 1 lu</p>
<p>_CF:0/1/2/,LU:lu,DP:1:HPIB address _CF:0/1/2,LU:lu,DP:1:HPIB address :0/1,LU:lu,DP:1:HPIB address :0/1/2/3/4,LU:lu,DP:1:HPIB address :0/1/2/3/4/5/6,LU:lu,DP:1:HPIB address :0/1/2/3/4/5/6/7,LU:lu,DP:1:HPIB address :0/1/2/3/4/5/6/7/8/9,LU:lu,DP:1:HPIB address :0/1/2/3/4/5/6/7/8/9/10/11,LU:lu,DP:1:HPIB address 500,DT:33B,TX:72,DX:8,DP:1:0,PR:0 500,DT:33B,TX:72,DX:8,DP:1:0,PR:0 :0/1/2/3,LU:lu,DP:1:HPIB address :0/1/2/3,LU:lu,DP:1:HPIB address 0/1/2/3/4/5,LU:lu,DP:1:HPIB address :0/1/2/3/4/5,LU:lu,DP:1:HPIB address 500,DT:33B,TX:72,DX:8,DP:1:0,PR:0 500,DT:33B,TX:72,DX:8,DP:1:0,PR:0 lu,DP:1:HPIB address :lu,DP:1:HPIB address</p>	<p>fixed cartridge 0,1,2 lu Removable cartridge 0,1,2 lu NODE, fixed cartridge 0,1,2 lu Removable cartridge 0,1,2 lu NODE, cartridge 0 lu, cartridge 1 lu, CTD lu NODE, cartridge 0 lu, ..., cartridge 4 lu, CTD lu NODE, cartridge 0 lu, ..., cartridge 6 lu, CTD lu NODE, cartridge 0 lu, ..., cartridge 7 lu, CTD lu NODE, cartridge 0 lu, ..., cartridge 9 lu NODE, cartridge 0 lu, ..., cartridge 9 lu NODE, cartridge 0 lu, ..., cartridge 7 lu NODE, cartridge 0 lu, ..., cartridge 9 lu NODE, cartridge 0 lu, ..., cartridge 3 lu NODE, cartridge 0 lu, ..., cartridge 3 lu NODE, cartridge 0 lu, ..., cartridge 3 lu NODE, cartridge 0 lu, ..., cartridge 5 lu NODE, cartridge 0 lu, ..., cartridge 5 lu NODE, cartridge 0 lu, ..., cartridge 7 lu NODE, cartridge 0 lu, ..., cartridge 5 lu NODE, CTD lu</p>
<p>1/2/3,LU:lu,DP:1:HPIB address lu,DP:1:HPIB address:Density lu,DP:1:HPIB address:Density 8:0,LU:lu,DP:1:HPIB address</p>	<p>* Use ID*01 if a 103/2.2 compatible Modem is part of the connection. ** DP1 must be set to 2 to disable ENQ/ACK.</p>
<p>xd devices) xd devices) xd devices),TX:57 xd devices) xd devices) xd devices),DX:3 xd devices) xd devices),TX:57</p>	<p>*** Where n is the number of HPIB devices that are using SRQ interrupts (for example, TX:138 allows two devices to use SRQ interrupts). If no devices use SRQ interrupts, allow TX to default to 135 words. **** For high speed devices, special cable-length restrictions apply. Refer to the 12009A HP-IB Interface Reference Manual for a description of the high speed data cable restrictions. HPIB address: The HP-IB driver, ID*37, requires that the HPIB address be entered into driver parameters 1 (DP1) at generation time. The HPIB address must be between 0 and 7 for discs, mag tapes and printers that use device drivers DD*12, DDC12, DD*33, DD*30, DDM30, DD*23 or DD*24.</p>



Model Number	IF DVR	Modules Relocated	Table Generation Phase IFT Entries	Table G
7957 (Using HP-IB)	ID*37	%ID*37	IFT,name,SC:sc	DVT,name , ,LU:lu,TO
7958 (Using HP-IB)	ID*37	%ID*37	" "	" , ,LU:lu,TO
9122D Dual 3-1/2" microfloppy	ID*37	%DD*33 %ID*37	" "	" ,M_FLOPP
9133D/9134D	ID*37	%DD*33 %ID*37	" "	" ,M9133_CF
9133H/9134H	ID*37	%DD*33 %ID*37	" "	" ,M9133_CF
9133L/9134L	ID*37	%DD*33 %ID*37	" "	" ,M9133_CF
Microfloppy integrated with 9133D/H/L	ID*37	%DD*33 %ID*37	" "	" ,M_FLOPP
9133XV 15Mb with microfloppy	ID*37	%DD*30 %ID*37	" "	" ,M9134x:0/
9134A 5-1/4" Winchester Disc	ID*37	%DD*30 %ID*37	" "	" ,M9134:0/1
9153B 20Mb with microfloppy	ID*37	%DD*33 %ID*37	" "	" ,M9133_CF
9895A Flexible Mini Discs	ID*37	%DD*30 %ID*37	" "	" ,M7902:0/1
7907A 41Mb CS/80 disc Fixed Pack	ID*37	%DD*33 %ID*37	IFT,name,SC:sc	DVT,name ,M7907FXD
7907A 41Mb CS/80 disc Removable Pack	ID*37	%DD*33 %ID*37	" "	" ,M7907REN
7908A 17Mb CS/80 disc with CTD	ID*37	%DD*33 %ID*37	" "	" ,M7908_CF
7911 28Mb CS/80 disc with CTD	ID*37	%DD*33 %ID*37	" "	" ,M7911_CF
7912 66Mb CS/80 disc with CTD	ID*37	%DD*33 %ID*37	" "	" ,M7912_CF
7914 132Mb CS/80 disc with CTD	ID*37	%DD*33 %ID*37	" "	" ,M7914_CF
7933 404Mb CS/80 disc Fixed Pack	ID*37	%DD*33 %ID*37	" "	" ,M7933_CF
7935 404Mb CS/80 disc Removable Pack	ID*37	%DD*33 %ID*37	" "	" ,M7933_CF
7936 (Using HP-IB)	ID*37	%DD*33 %ID*37	" "	" , ,LU:lu,TO
7937 (Using HP-IB)	ID*37	%ID*37	" "	" , ,LU:lu,TO
7941A 24Mb CS/80 disc	ID*37	%ID*37	" "	" ,M7941_CF
7942A 24Mb CS/80 disc with CTD	ID*37	%DD*33 %ID*37	" "	" ,M7941_CF
7945A 56Mb CS/80	ID*37	%DD*33 %ID*37	" "	" ,M7945_CF
7946A 56Mb CS/80 disc with CTD	ID*37	%DD*33 %ID*37	" "	" ,M7945_CF
7957 (Using HP-IB)	ID*37	%ID*37	" "	" , ,LU:lu,TO
7958 (Using HP-IB)	ID*37	%ID*37	" "	" , ,LU:lu,TO
9144 CTD (standalone)	ID*37	%DD*33 %ID*37	" "	" ,M9144,LU:
CTD	ID*37	%DD*33 %ID*37	" "	" ,MTAPE,LU
7970E HP-IB Mag Tape	ID*37	%DD*23 %ID*37	IFT,name,SC:sc	DVT,name ,M7970E:0/
7974A HP-IB Mag Tape	ID*37	%DD*24 %ID*37	" "	" ,M7974,LU:
7978A/B HP-IB Mag Tape	ID*37	%DD*24 %ID*37	" "	" ,M7978,LU:
7979/80 HP-IB Mag Tape	ID*37	%DD*24 %ID*37	IFT,name,SC:sc	DVT,name ,M7974/797
12005A Asynchronous Serial Interface	ID*00	%DD*00 %ID*00	IFT,name,SC:sc	(all DVT entries for associate
12009 HP-IB Interface ****	ID*37	%ID*37	" ,SC:sc, - TX:130+4*n***	(all DVT entries for associate
12040 Multiplexer Interface (Note B)	IDM00	%DD*00 %IDM00	" "	(all DVT entries for associate
12153 Writable Control Store	ID*41	%ID*41	" "	(all DVT entries for associate
12205 Control Store	ID*42	%ID*42	" "	(all DVT entries for associate
12006 Parallel Interface	ID*50	%ID*50	" "	(all DVT entries for associate
12006 Parallel Interface	ID*52	%ID*52	" "	(all DVT entries for associate
37222 Multiplexer Interface	IDM00	%DD*00 %IDM00	" "	(all DVT entries for associate

NOTE A: The RS232 MUX listed in the table above is the HP 12020A-C. The following gen records apply to Serial I/O devices using the 12040D MUX or the

SERIAL I/O CARD	IFT DVR	MODULES RELOCATED	TABLE GENERATION PHASE IFT ENTRIES	TABLE GENERATION PHASE DVT ENTRIES
Using 12040D MUX	ID800	DDC00.REL or DDC01.REL, ID800.REL or ID801.REL	IFT,name ,SC:sc	DVT,name,MHP_Term,LU:lu
Using 12100A OBIO	ID400	DDC00 DDC01 ID400.REL	IFT,name (see NOTE B)	DVT,name,MHP_Term,LU:lu

NOTE B: The A400 OBIO select code is always 77B and should NOT be supplied in the IFT specification line. It is defaulted.

NOTE C: The 700/41 Terminal cannot be used as a boot device and does not support CI stack and Edit/1000 screen mode.

Table E-4. Flexible Disc Default File Values

	7902 Disc	7902 Disc
Model Number	7902:0	7902:1
Entry Point	DD.30	DD.30
Device Type	30B	30B
Timeout	750	750
Buffer Limits	NONE	NONE
Table Extension	25	25
Priority	0	0
Queuing	NONE	NONE
Number of Driver Parameters	8	8
Driver Parameter	DP 1: 0 (HP-IB Addr.)	DP 1: 0 (HP-IB Addr.)
Area Defaults	DP 2: 0 DP 3: 0 DP 4: 0 DP 5: 3 DP 6: 134 DP 7: 30 DP 8: 2	DP 2: 1 DP 3: 0 DP 4: 0 DP 5: 3 DP 6: 134 DP 7: 30 DP 8: 2

Table E-5. 9133A/9134A Disc (Four-Volume Format) Default File Values

	9134 Disc			
	9134:0	9134:1	9134:2	9134:3
Model Number	9134:0	9134:1	9134:2	9134:3
Entry Point	DD.30	DD.30	DD.30	DD.30
Device Type	30B	30B	30B	30B
Timeout	750	750	750	750
Buffer Limits	NONE	NONE	NONE	NONE
Table Extension	25	25	25	25
Priority	0	0	0	0
Queuing	NONE	NONE	NONE	NONE
Number of Driver Parameters	8	8	8	8
Driver Parameter Area Defaults	DP 1: 0 (HP-IB Addr.) DP 2: 0 DP 3: 0 DP 4: 0 DP 5: 3 DP 6: 134 DP 7: 30 DP 8: 2	DP 1: 0 (HP-IB Addr.) DP 2: 1 DP 3: 0 DP 4: 0 DP 5: 3 DP 6: 134 DP 7: 30 DP 8: 2	DP 1: 0 (HP-IB Addr.) DP 2: 2 DP 3: 0 DP 4: 0 DP 5: 3 DP 6: 134 DP 7: 30 DP 8: 2	DP 1: 0 (HP-IB Addr.) DP 2: 3 DP 3: 0 DP 4: 0 DP 5: 3 DP 6: 134 DP 7: 30 DP 8: 2

Table E-6. 9133B/9134B Disc (Single-Volume Format) Default File Values

	9134 Disc		
Model Number	9134L:0	9134L:1	9134L:2
Entry Point	DD.30	DD.30	DD.30
Device Type	30B	30B	30B
Timeout	750	750	750
Buffer Limits	NONE	NONE	NONE
Table Extension	25	25	25
Priority	0	0	0
Queuing	NONE	NONE	NONE
Number of Driver Parameters	8	8	8
Driver Parameter Area Defaults	DP 1: 0 (HP-IB Addr.)	DP 1: 0 (HP-IB Addr.)	DP 1: 0 (HP-IB Addr.)
	DP 2: 0	DP 2: 0	DP 2: 0
	DP 3: 0	DP 3: 0	DP 3: 0
	DP 4: 0	DP 4: 118	DP 4: 152
	DP 5: 0	DP 5: 0	DP 5: 0
	DP 6: 472	DP 6: 136	DP 6: 612
	DP 7: 31	DP 7: 31	DP 7: 31
	DP 8: 4	DP 8: 4	DP 8: 4

Table E-7. 9133D/H/L 9134D/H/L Default File Values

9133/9134 Disc				
	M9133_CF:0	M9133_CF:1	M9133_CF:2	M9133_CF:3
Model Number	DD.33	DD.33	DD.33	DD.33
Entry Point	33B	33B	33B	33B
Device Type	500	500	500	500
Timeout	NONE	NONE	NONE	NONE
Buffer Limits	72	72	72	72
Table Extension	0	0	0	0
Priority	FIFO	FIFO	FIFO	FIFO
Queuing	8	8	8	8
Number of Driver Parameters	DP 1: 0	DP 1: 0	DP 1: 0	DP 1: 0
Driver Parameter Area Defaults	DP 2: 0	DP 2: 0	DP 2: 0	DP 2: 0
	DP 3: 0	DP 3: 0	DP 3: 0	DP 3: 0
	DP 4: 0	DP 4: 0	DP 4: 0	DP 4: 1
	DP 5: 0	DP 5: 28992	DP 5: 57984	DP 5: 12272
	DP 6: 604	DP 6: 604	DP 6: 413	DP 6: 1622
	DP 7: 48	DP 7: 48	DP 7: 48	DP 7: 48
	DP 8: 0	DP 8: 0	DP 8: 0	DP 8: 0

9133D/9134D

9133H/9134H or 9153/9154

9133L/9134L

Table E-8. 9133D/H/L Dual-Sided Double Density Microfloppy Default File Values

9133D/H/L Microfloppy	
1st LU	
Model Number	M_FLOPPY_CF:1
Entry Point	DD.33
Device Type	33B
Timeout	500
Buffer Limits	NONE
Table Extension	72
Priority	0
Queuing	FIFO
Number of Driver Parameters	8
Driver Parameter	DP 1: 0 (HP-IB Addr.)
Area Defaults	DP 2: 400B DP 3: 0 DP 4: 0 DP 5: 0 DP 6: 154 DP 7: 16 DP 8: 0

Table E-9. 9122D Dual-Sided Double Density Microfloppy Default File Values

	9122D	
	1st LU	2nd LU
Model Number	M_FLOPPY_CF:0	M_FLOPPY_CF:1
Entry Point	DD.33	DD.33
Device Type	33B	33B
Timeout	500	500
Buffer Limits	NONE	NONE
Table Extension	72	72
Priority	0	0
Queuing	NONE	NONE
Number of Driver Parameters	8	8
Driver Parameter	DP 1: 0 (HP-IB Addr.)	DP 1: 0 (HP-IB Addr.)
Area Defaults	DP 2: 0 DP 3: 0 DP 4: 0 DP 5: 0 DP 6: 154 DP 7: 16 DP 8: 0	DP 2: 400B DP 3: 0 DP 4: 0 DP 5: 0 DP 6: 154 DP 7: 16 DP 8: 0

Table E-10. 9121/9133A/B/XV Single-Sided Microfloppy File Default Values

	9121		9133A/B/XV
	Left Floppy	Right Floppy	Floppy
Model Number	M9121:0	M9121:1	M9121:0
Entry Point	DD.30	DD.30	DD.30
Device Type	30B	30B	30B
Timeout	3000	3000	3000
Buffer Limits	NONE	NONE	NONE
Table Extension	25	25	25
Number of Driver Parameters	8	8	8
Driver Parameter Area Defaults	DP 1: 0 (HP-IB Addr.) DP 2: 0 DP 3: 0 DP 4: 0 DP 5: 2 DP 6: 66 DP 7: 16 DP 8: 2	DP 1: 0 (HP-IB Addr.) DP 2: 1 DP 3: 0 DP 4: 0 DP 5: 2 DP 6: 66 DP 7: 16 DP 8: 2	DP 1: 0 (HP-IB Addr.) DP 2: 0 DP 3: 0 DP 4: 0 DP 5: 2 DP 6: 66 DP 7: 16 DP 8: 2

Table E-11. Mag Tape Default File Values

	7970E				7974A	7978A/B	9144
	Unit 0	Unit 1	Unit 2	Unit 3			
Model Number	7970E:0	7970E:1	7970E:2	7970E:3	7974	7978	M9144
Entry Point	DD.23	DD.23	DD.23	DD.23	DD.24	DD.24	DD.33
Device Type	23B	23B	23B	23B	24B*	24B*	26B
Timeout	1000	1000	1000	1000	500	500	1000
Buffer Limits	NONE	NONE	NONE	NONE	NONE	NONE	NONE
Table Extension	30	30	30	30	43	43	72
Priority	NONE	NONE	NONE	NONE	NONE	NONE	NONE
Number of Driver Parameters	3	3	3	3	5	5	8
Driver Parameter Area Defaults	DP2:400B	DP2:1000B	DPT:1400B	DP2:2000B	DP1:4 DP2:1600	DP1:4 DP2:1600	DP1:0 DP2:0 DP3:0 DP4-8: res.

* For utilities software date-coded 2401 or lower, generate with device type 23B instead of 24B.

Table E-12. 7906 Default File Values

	Cartridge 0	Cartridge 1	Cartridge 2	Cartridge 3
Model Number	M7906:0	M7906:1	M7906:2	M7906:3
Entry Point	DD.30	DD.30	DD.30	DD.30
Device Type	32B	32B	32B	32B
Timeout	3000	3000	3000	3000
Buffer Limits	NONE	NONE	NONE	NONE
Table Extension	25	25	25	25
Number of Driver Parameters	8	8	8	8
Driver Parameter Area Defaults	DP 1: 0 DP 2: 0 DP 3: 0 DP 4: 0 DP 5: 5 DP 6: 406 DP 7: 48 DP 8: 1	DP 1: 0 DP 2: 0 DP 3: 1 DP 4: 0 DP 5: 5 DP 6: 406 DP 7: 48 DP 8: 1	DP 1: 0 DP 2: 0 DP 3: 2 DP 4: 0 DP 5: 5 DP 6: 406 DP 7: 48 DP 8: 1	DP 1: 0 DP 2: 0 DP 3: 3 DP 4: 0 DP 5: 5 DP 6: 406 DP 7: 48 DP 8: 1
	↑ TOP	↑ BOTTOM	↑ TOP	↑ BOTTOM
	Removable Platter		Fixed Platter	

Table E-13. 7906M Default File Values

	Cartridge 0	Cartridge 1	Cartridge 2	Cartridge 3
Model Number	M7906:0	M7906:1	M7906:2	M7906:3
Entry Point	DDM30	DDM30	DDM30	DDM30
Device Type	32B	32B	32B	32B
Timeout	3000	3000	3000	3000
Buffer Limits	NONE	NONE	NONE	NONE
Table Extension	51	51	51	51
Number of Driver Parameters	8	8	8	8
Driver Parameter Area Defaults	DP 1: none DP 2: 0 DP 3: 0 DP 4: 0 DP 5: 5 DP 6: 406 DP 7: 48 DP 8: 1	DP 1: none DP 2: 0 DP 3: 1 DP 4: 0 DP 5: 5 DP 6: 406 DP 7: 48 DP 8: 1	DP 1: none DP 2: 0 DP 3: 2 DP 4: 0 DP 5: 5 DP 6: 406 DP 7: 48 DP 8: 1	DP 1: none DP 2: 0 DP 3: 3 DP 4: 0 DP 5: 5 DP 6: 406 DP 7: 48 DP 8: 1
	↑ TOP	↑ BOTTOM	↑ TOP	↑ BOTTOM
	Removable Platter		Fixed Platter	



Table E-14. 7920 Default File Values

Cartridge 4	Cartridge 5	Cartridge 6	Cartridge 7	Cartridge 8	Cartridge 9
M7920:4	M7920:5	M7920:6	M7920:7	M7920:8	M7920:9
DD.30	DD.30	DD.30	DD.30	DD.30	DD.30
32B	32B	32B	32B	32B	32B
3000	3000	3000	3000	3000	3000
NONE	NONE	NONE	NONE	NONE	NONE
25	25	25	25	25	25
8	8	8	8	8	8
DP 1: NONE DP 2: 0 DP 3: 0 DP 4: 332 DP 5: 9 DP 6: 406 DP 7: 48 DP 8: 5	DP 1: NONE DP 2: 0 DP 3: 0 DP 4: 415 DP 5: 9 DP 6: 406 DP 7: 48 DP 8: 5	DP 1: NONE DP 2: 0 DP 3: 0 DP 4: 498 DP 5: 9 DP 6: 406 DP 7: 48 DP 8: 5	DP 1: NONE DP 2: 0 DP 3: 0 DP 4: 581 DP 5: 9 DP 6: 406 DP 7: 48 DP 8: 5	DP 1: NONE DP 2: 0 DP 3: 0 DP 4: 664 DP 5: 9 DP 6: 406 DP 7: 48 DP 8: 5	DP 1: NONE DP 2: 0 DP 3: 0 DP 4: 747 DP 5: 10 DP 6: 370 DP 7: 48 DP 8: 5

	Cartridge 0	Cartridge 1	Cartridge 2	Cartridge 3
Model Number	M7920:0	M7920:1	M7920:2	M7920:3
Entry Point	DD.30	DD.30	DD.30	DD.30
Device Type	32B	32B	32B	32B
Timeout	3000	3000	3000	3000
Buffer Limits	NONE	NONE	NONE	NONE
Table Extension	25	25	25	25
Number of Driver Parameters	8	8	8	8
Driver Parameter Area Defaults	DP 1: NONE DP 2: 0 DP 3: 0 DP 4: 0 DP 5: 9 DP 6: 406 DP 7: 48 DP 8: 5	DP 1: NONE DP 2: 0 DP 3: 0 DP 4: 83 DP 5: 9 DP 6: 406 DP 7: 48 DP 8: 5	DP 1: NONE DP 2: 0 DP 3: 0 DP 4: 166 DP 5: 9 DP 6: 406 DP 7: 48 DP 8: 5	DP 1: NON DP 2: 0 DP 3: 0 DP 4: 249 DP 5: 9 DP 6: 406 DP 7: 48 DP 8: 5

Table E-15. 7920M Default File Values

Cartridge 4	Cartridge 5	Cartridge 6	Cartridge 7	Cartridge 8	Cartridge 9
M7920:4	M7920:5	M7920:6	M7920:7	M7920:8	M7920:9
DDM30	DDM30	DDM30	DDM30	DDM30	DDM30
32B	32B	32B	32B	32B	32B
3000	3000	3000	3000	3000	3000
NONE	NONE	NONE	NONE	NONE	NONE
51	51	51	51	51	51
8	8	8	8	8	8
DP 1: NONE	DP 1: NONE	DP 1: NONE	DP 1: NONE	DP 1: NONE	DP 1: NONE
DP 2: 0	DP 2: 0	DP 2: 0	DP 2: 0	DP 2: 0	DP 2: 0
DP 3: 0	DP 3: 0	DP 3: 0	DP 3: 0	DP 3: 0	DP 3: 0
DP 4: 332	DP 4: 415	DP 4: 498	DP 4: 581	DP 4: 664	DP 4: 747
DP 5: 9	DP 5: 9	DP 5: 9	DP 5: 9	DP 5: 9	DP 5: 10
DP 6: 406	DP 6: 406	DP 6: 406	DP 6: 406	DP 6: 406	DP 6: 370
DP 7: 48	DP 7: 48	DP 7: 48	DP 7: 48	DP 7: 48	DP 7: 48
DP 8: 5	DP 8: 5	DP 8: 5	DP 8: 5	DP 8: 5	DP 8: 5

	Cartridge 0	Cartridge 1	Cartridge 2	Cartridge 3
Model Number	M7920:0	M7920:1	M7920:2	M7920:3
Entry Point	DDM30	DDM30	DDM30	DDM30
Device Type	32B	32B	32B	32B
Timeout	3000	3000	3000	3000
Buffer Limits	NONE	NONE	NONE	NONE
Table Extension	51	51	51	51
Number of Driver Parameters	8	8	8	8
Driver Parameter Area Defaults	DP 1: NONE DP 2: 0 DP 3: 0 DP 4: 0 DP 5: 9 DP 6: 406 DP 7: 48 DP 8: 5	DP 1: NONE DP 2: 0 DP 3: 0 DP 4: 83 DP 5: 9 DP 6: 406 DP 7: 48 DP 8: 5	DP 1: NONE DP 2: 0 DP 3: 0 DP 4: 166 DP 5: 9 DP 6: 406 DP 7: 48 DP 8: 5	DP 1: NONE DP 2: 0 DP 3: 0 DP 4: 249 DP 5: 9 DP 6: 406 DP 7: 48 DP 8: 5

Table E-16. 7925 Default File Values

Cartridge 3	Cartridge 4	Cartridge 5	Cartridge 6	Cartridge 7	Cartridge 8
M7925:3	M7925:4	M7925:5	M7925:6	M7925:7	M7925:8
DD.30	DD.30	DD.30	DD.30	DD.30	DD.30
32B	32B	32B	32B	32B	32B
3000	3000	3000	3000	3000	3000
NONE	NONE	NONE	NONE	NONE	NONE
25	25	25	25	25	25
8	8	8	8	8	8
DP 1: NONE	DP 1: NONE	DP 1: NONE	DP 1: NONE	DP 1: NONE	DP 1: NONE
DP 2: 0	DP 2: 0	DP 2: 0	DP 2: 0	DP 2: 0	DP 2: 0
DP 3: 0	DP 3: 0	DP 3: 0	DP 3: 0	DP 3: 0	DP 3: 0
DP 4: 186	DP 4: 248	DP 4: 363	DP 4: 478	DP 4: 593	DP 4: 708
DP 5: 9	DP 5: 11	DP 5: 11	DP 5: 11	DP 5: 11	DP 5: 11
DP 6: 549	DP 6: 1024	DP 6: 1024	DP 6: 1024	DP 6: 1024	DP 6: 1024
DP 7: 64	DP 7: 64	DP 7: 64	DP 7: 64	DP 7: 64	DP 7: 64
DP 8: 9	DP 8: 9	DP 8: 9	DP 8: 9	DP 8: 9	DP 8: 9



	Cartridge 0	Cartridge 1	Cartridge 2
Model Number	M7925:0	M7925:1	M7925:2
Entry Point	DD.30	DD.30	DD.30
Device Type	32B	32B	32B
Timeout	3000	3000	3000
Buffer Limits	NONE	NONE	NONE
Table Extension	25	25	25
Number of Driver Parameters	8	8	8
Driver Parameter Area Defaults	DP 1: NONE DP 2: 0 DP 3: 0 DP 4: 0 DP 5: 9 DP 6: 549 DP 7: 64 DP 8: 9	DP 1: NONE DP 2: 0 DP 3: 0 DP 4: 62 DP 5: 9 DP 6: 549 DP 7: 64 DP 8: 9	DP 1: NONE DP 2: 0 DP 3: 0 DP 4: 124 DP 5: 9 DP 6: 549 DP 7: 64 DP 8: 9

Table E-17. 7925M Default File Values

File 2	Cartridge 3	Cartridge 4	Cartridge 5	Cartridge 6	Cartridge 7	Cartridge 8
	M7925:3	M7925:4	M7925:5	M7925:6	M7925:7	M7925:8
	DDM30	DDM30	DDM30	DDM30	DDM30	DDM30
	32B	32B	32B	32B	32B	32B
	3000	3000	3000	3000	3000	3000
	NONE	NONE	NONE	NONE	NONE	NONE
	51	51	51	51	51	51
	8	8	8	8	8	8
ONE	DP 1: NONE	DP 1: NONE	DP 1: NONE	DP 1: NONE	DP 1: NONE	DP 1: NONE
	DP 2: 0	DP 2: 0	DP 2: 0	DP 2: 0	DP 2: 0	DP 2: 0
	DP 3: 0	DP 3: 0	DP 3: 0	DP 3: 0	DP 3: 0	DP 3: 0
24	DP 4: 186	DP 4: 248	DP 4: 363	DP 4: 478	DP 4: 593	DP 4: 708
	DP 5: 9	DP 5: 11	DP 5: 11	DP 5: 11	DP 5: 11	DP 5: 11
49	DP 6: 549	DP 6: 1024	DP 6: 1024	DP 6: 1024	DP 6: 1024	DP 6: 1024
4	DP 7: 64	DP 7: 64	DP 7: 64	DP 7: 64	DP 7: 64	DP 7: 64
	DP 8: 9	DP 8: 9	DP 8: 9	DP 8: 9	DP 8: 9	DP 8: 9

	Cartridge 0	Cartridge 1	Cartridge 2
Model Number	M7925:0	M7925:1	M7925:2
Entry Point	DDM30	DDM30	DDM30
Device Type	32B	32B	32B
Timeout	3000	3000	3000
Buffer Limits	NONE	NONE	NONE
Table Extension	51	51	51
Number of Driver Parameters	8	8	8
Driver Parameter Area Defaults	DP 1: NONE DP 2: 0 DP 3: 0 DP 4: 0 DP 5: 9 DP 6: 549 DP 7: 64 DP 8: 9	DP 1: NONE DP 2: 0 DP 3: 0 DP 4: 62 DP 5: 9 DP 6: 549 DP 7: 64 DP 8: 9	DP 1: N DP 2: 0 DP 3: 0 DP 4: 1 DP 5: 9 DP 6: 5 DP 7: 6 DP 8: 9

Table E-18. Cartridge Tape Drive File Default Values

	CTD
Entry Point	DD.33
Device Type	26B
Timeout	500
Buffer Limits	NONE
Table Extension	72
Number of Driver Parameters	8
Driver Parameter Area Defaults	DP 1: 0 DP 2: 400B DP 3: 100000B DP 4: 0 DP 5: 64464 DP 6: 0 DP 7: 0 DP 8: 0



Note: Default values are identical for the CTD drive with all CS/80 discs.

Table E-20. 7907/08/11/12/14/33/41/42/45/46 Default File Values

Cartridge 4	Cartridge 5	Cartridge 6	Cartridge 7	Cartridge 8	Cartridge 9
M7911_CF:4 M7945_CF:4 M7912_CF:4 M7914_CF:4 M7933_CF:4	M7945_CF:5 M7912_CF:5 M7914_CF:5 M7933_CF:5	M7912_CF:6 M7914_CF:6 M7933_CF:6	M7914_CF:7 M7933_CF:7	M7933_CF:8	M7933_CF:9
DD.33	DD.33	DD.33	DD.33	DD.33	DD.33
33B	33B	33B	33B	33B	33B
NONE	NONE	NONE	NONE	NONE	NONE
72	72	72	72	72	72
0	0	0	0	0	0
FIFO	FIFO	FIFO	FIFO	FIFO	FIFO
8	8	8	8	8	8
DP 1: 0 DP 2: 0 DP 3: 0 DP 4: 1 DP 5: 27392 DP 6: 352 DP 7: 48 DP 8: 0	DP 1: 0 DP 2: 0 DP 3: 0 DP 4: 1 DP 5: 44288 DP 6: 2229 DP 7: 48 DP 8: 0	DP 1: 0 DP 2: 0 DP 3: 0 DP 4: 3 DP 5: 20223 DP 6: 821 DP 7: 48 DP 8: 0	DP 1: 0 DP 2: 0 DP 3: 0 DP 4: 3 DP 5: 59648 DP 6: 5413 DP 7: 48 DP 8: 0	DP 1: 0 DP 2: 0 DP 3: 0 DP 4: 7 DP 5: 57344 DP 6: 8862 DP 7: 48 DP 8: 0	DP 1: 0 DP 2: 0 DP 3: 0 DP 4: 14 DP 5: 23968 DP 6: 13300 DP 7: 48 DP 8: 0

	Cartridge 0	Cartridge 1	Cartridge 2	Cartridge 3
Model Number	M7908_CF:0 M7907FXD_CF:0 M7941_CF:0 M7911_CF:0 M7945_CF:0 M7912_CF:0 M7914_CF:0 M7933_CF:0	M7908_CF:1 M7907FXD_CF:1 M7941_CF:1 M7911_CF:1 M7945_CF:1 M7912_CF:1 M7914_CF:1 M7933_CF:1	M7907FXD_CF:2 M7941_CF:2 M7911_CF:2 M7945_CF:2 M7912_CF:2 M7914_CF:2 M7933_CF:2	M7941_CF:3 M7911_CF:3 M7945_CF:3 M7912_CF:3 M7914_CF:3 M7933_CF:3
Entry Point	DD.33	DD.33	DD.33	DD.33
Device Type	33B	33B	33B	33B
Buffer Limits	NONE	NONE	NONE	NONE
Table Extension	72	72	72	72
Priority	0	0	0	0
Queuing	FIFO	FIFO	FIFO	FIFO
Number of Driver Parameters	8	8	8	8
Driver Parameter Area Defaults	DP 1: 0 DP 2: 0 DP 3: 0 DP 4: 0 DP 5: 0 DP 6: 400 DP 7: 48 DP 8: 0	DP 1: 0 DP 2: 0 DP 3: 0 DP 4: 0 DP 5: 19200 DP 6: 943 DP 7: 48 DP 8: 0	DP 1: 0 DP 2: 0 DP 3: 0 DP 4: 0 DP 5: 64750 DP 6: 320 DP 7: 48 DP 8: 0	DP 1: 0 DP 2: 0 DP 3: 0 DP 4: 1 DP 5: 1459 DP 6: 266 DP 7: 48 DP 8: 0

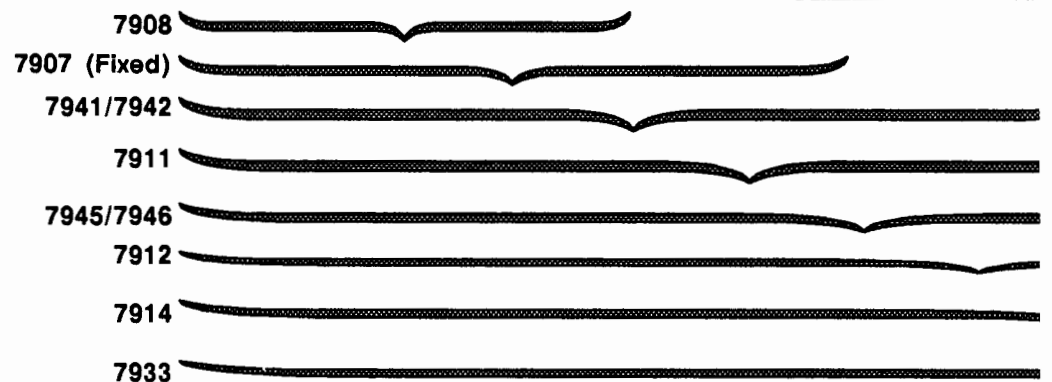


Table E-21. 248X Default File Values

15-Mbyte Disc				20-Mbyte Disc		Single Sided Microfloppy	Double Sided Microfloppy
	2480:6	2480:7	2480:13	2480:11	2480:12	2480:3	2480:14
	none	none	none	none	none	none	none
	30B	30B	30B	30B	30B	30B	30B
	5000	5000	5000	5000	5000	5000	5000
	none	none	none	none	none	none	none
	0	0	0	0	0	0	0
	0	0	0	0	0	0	0
	none	none	none	none	none	none	none
	9	9	9	9	9	9	9
0	DP 1: 0	DP 1: 0	DP 1: 0	DP 1: 0	DP 1: 0	DP 1: 3	DP 1: 3
0	DP 2: 0	DP 2: 0	DP 2: 0	DP 2: 0	DP 2: 0	DP 2: 0	DP 2: 0
0	DP 3: 0	DP 3: 0	DP 3: 0	DP 3: 0	DP 3: 0	DP 3: 0	DP 3: 0
104	DP 4: 159	DP 4: 236	DP 4: 104	DP 4: 0	DP 4: 489	DP 4: 0	DP 4: 0
0	DP 5: 0	DP 5: 0	DP 5: 0	DP 5: 0	DP 5: 0	DP 5: 0	DP 5: 0
330	DP 6: 462	DP 6: 420	DP 6: 1212	DP 6: 1956	DP 6: 476	DP 6: 66	DP 6: 154
31	DP 7: 31	DP 7: 31	DP 7: 31	DP 7: 31	DP 7: 31	DP 7: 16	DP 7: 16
6	DP 8: 6	DP 8: 6	DP 8: 6	DP 8: 4	DP 8: 4	DP 8: 1	DP 8: 2
0	DP 9: 0	DP 9: 0	DP 9: 0	DP 9: 0	DP 9: 0	DP 9: 0	DP 9: 0



	10-Mbyte Disc				
	2480:0	2480:1	2480:2	2480:4	2480:5
Model Number	2480:0	2480:1	2480:2	2480:4	2480:5
Entry Point	none	none	none	none	none
Device Type	30B	30B	30B	30B	30B
Timeout	5000	5000	5000	5000	5000
Buffer Limits	none	none	none	none	none
Table Extension	0	0	0	0	0
Priority	0	0	0	0	0
Queuing	none	none	none	none	none
Number of Driver Parameters	9	9	9	9	9
Driver Parameter Area Defaults	DP 1: 0 DP 2: 0 DP 3: 0 DP 4: 0 DP 5: 0 DP 6: 472 DP 7: 31 DP 8: 4 DP 9: 0	DP 1: 0 DP 2: 0 DP 3: 0 DP 4: 118 DP 5: 0 DP 6: 136 DP 7: 31 DP 8: 4 DP 9: 0	DP 1: 0 DP 2: 0 DP 3: 0 DP 4: 152 DP 5: 0 DP 6: 616 DP 7: 31 DP 8: 4 DP 9: 0	DP 1: 0 DP 2: 0 DP 3: 0 DP 4: 0 DP 5: 0 DP 6: 624 DP 7: 31 DP 8: 6 DP 9: 0	DP 1: DP 2: DP 3: DP 4: DP 5: DP 6: DP 7: DP 8: DP 9:

Note: Subchannel 13 is a merged version of subchannels 5 through 7. You can use subchannels 5 through 7 or subchannel 13, but not both.

Table E-22. IFT Default File Values

	12005A	12006A (PIC) (4)	12006A (ICOM)	12008A (1)	12009A (2)	12040x 37222A (3)	12153A	12205A
Entry Point	ID.00	ID.50	ID.52	ID.36	ID.37	IDM00	ID.41	ID.42
IFT Extension	33	9	27	6	122 + (4 * n)	20	4	5
Queuing	FIFO	NONE	NONE	FIFO	Priority	FIFO	User Defined	User Defined
DVT Parameters	NONE	3	NONE	NONE	NONE	NONE	NONE	NONE
DVT/IFT Device Type	00B	50B	52B	36B	37B	00B	41B	42B
timeout	5000			500				
DVT Extension	2			8		57		
Driver Parameter Area Defaults	DP1:140B DP2:100B DP3:11B			DP 1: 0 DP 2: 0 DP 3: 0 DP 4: 0 DP 5: 0 DP 6: 64 DP 7: 4				

- Notes: (1) PROM DVT default file is %DD*36 – not a device driver
(2) In calculating the extension, n = number of serial poll table entries
(3) IDM00 rev 2226 or earlier; IDM00 rev 2301 or later for 12040A
(4) These defaults must be explicitly set in the generation, since ID.50 requires no device driver.

Device and Interface Driver Tables E-33

Device and Interface Driver Tables E-33

Device and Interface Driver Tables E-33

Device and Interface Driver Tables E-33

Device and Interface Driver Tables E-33

Device and Interface Driver Tables E-33

Device and Interface Driver Tables E-33

Table E-23. 7936 Default File Values

Cartridge 2	Cartridge 3	Cartridge 4	Cartridge 5	Cartridge 6	Cartridge 7
M7936_CF:2	M7936_CF:3	M7936_CF:4	M7936_CF:5	M7936_CF:6	M7936_CF:7
DD.33	DD.33	DD.33	DD.33	DD.33	DD.33
33B	33B	33B	33B	33B	33B
NONE	NONE	NONE	NONE	NONE	NONE
72	72	72	72	72	72
8	8	8	8	8	8
DP 1: 0 DP 2: 0 DP 3: 0 DP 4: 0 DP 5: 64750 DP 6: 2560 DP 7: 64 DP 8: 0	DP 1: 0 DP 2: 0 DP 3: 0 DP 4: 3 DP 5: 31982 DP 6: 2560 DP 7: 64 DP 8: 0	DP 1: 0 DP 2: 0 DP 3: 0 DP 4: 5 DP 5: 64750 DP 6: 2560 DP 7: 64 DP 8: 0	DP 1: 0 DP 2: 0 DP 3: 0 DP 4: 8 DP 5: 31982 DP 6: 2560 DP 7: 64 DP 8: 0	DP 1: 0 DP 2: 0 DP 3: 0 DP 4: 10 DP 5: 64750 DP 6: 2560 DP 7: 64 DP 8: 0	DP 1: 0 DP 2: 0 DP 3: 0 DP 4: 13 DP 5: 31982 DP 6: 4968 DP 7: 64 DP 8: 0

	Cartridge 0	Cartridge 1
Model Number	M7936_CF:0	M7936_CF:1
Entry Point	DD.33	DD.33
Device Type	33B	33B
Buffer Limits	NONE	NONE
Table Extension	72	72
Number of Driver Area Defaults	8	8
Driver Parameter Area Defaults	DP 1: 0 DP 2: 0 DP 3: 0 DP 4: 0 DP 5: 0 DP 6: 400 DP 7: 48 DP 8: 0	DP 1: 0 DP 2: 0 DP 3: 0 DP 4: 0 DP 5: 19200 DP 6: 943 DP 7: 48 DP 8: 0

Table E-24. 7937 Default File Values

Cartridge 2	Cartridge 3	Cartridge 4	Cartridge 5	Cartridge 6	Cartridge 7	Cartridge 8	Cartridge 9
M7937_CF:2	M7937_CF:3	M7937_CF:4	M7937_CF:5	M7937_CF:6	M7937_CF:7	M7937_CF:8	M7937_CF:9
DD.33	DD.33	DD.33	DD.33	DD.33	DD.33	DD.33	DD.33
33B	33B	33B	33B	33B	33B	33B	33B
NONE	NONE	NONE	NONE	NONE	NONE	NONE	NONE
72	72	72	72	72	72	72	72
8	8	8	8	8	8	8	8
DP 1: 0 DP 2: 0 DP 3: 0 DP 4: 5 DP 5: 3310 DP 6: 4160 DP 7: 64 DP 8: 0	DP 1: 0 DP 2: 0 DP 3: 0 DP 4: 9 DP 5: 7406 DP 6: 4160 DP 7: 64 DP 8: 0	DP 1: 0 DP 2: 0 DP 3: 0 DP 4: 13 DP 5: 11502 DP 6: 4160 DP 7: 64 DP 8: 0	DP 1: 0 DP 2: 0 DP 3: 0 DP 4: 17 DP 5: 15598 DP 6: 4160 DP 7: 64 DP 8: 0	DP 1: 0 DP 2: 0 DP 3: 0 DP 4: 21 DP 5: 19694 DP 6: 4160 DP 7: 64 DP 8: 0	DP 1: 0 DP 2: 0 DP 3: 0 DP 4: 25 DP 5: 23790 DP 6: 4160 DP 7: 64 DP 8: 0	DP 1: 0 DP 2: 0 DP 3: 0 DP 4: 29 DP 5: 27886 DP 6: 4746 DP 7: 64 DP 8: 0	DP 1: 0 DP 2: 0 DP 3: 0 DP 4: 29 DP 5: 27886 DP 6: 4746 DP 7: 64 DP 8: 0

	Cartridge 0	Cartridge 1	Cartr
Model Number	M7937_CF:0	M7937_CF:1	M793
Entry Point	DD.33	DD.33	DD.3
Device Type	33B	33B	33B
Buffer Limits	NONE	NONE	NONE
Table Extension	72	72	72
Number of Driver Area Defaults	8	8	8
Driver Parameter Area Defaults	DP 1: 0 DP 2: 0 DP 3: 0 DP 4: 0 DP 5: 0 DP 6: 400 DP 7: 48 DP 8: 0	DP 1: 0 DP 2: 0 DP 3: 0 DP 4: 0 DP 5: 19200 DP 6: 943 DP 7: 48 DP 8: 0	DP 1 DP 2 DP 3 DP 4 DP 5 DP 6 DP 7 DP 8

Table E-24. 7937 Default File Values

Cartridge 2	Cartridge 3	Cartridge 4	Cartridge 5	Cartridge 6	Cartridge 7	Cartridge 8	Cartridge 9
M7937_CF:2	M7937_CF:3	M7937_CF:4	M7937_CF:5	M7937_CF:6	M7937_CF:7	M7937_CF:8	M7937_CF:9
DD.33	DD.33	DD.33	DD.33	DD.33	DD.33	DD.33	DD.33
33B	33B	33B	33B	33B	33B	33B	33B
NONE	NONE	NONE	NONE	NONE	NONE	NONE	NONE
72	72	72	72	72	72	72	72
8	8	8	8	8	8	8	8
DP 1: 0 DP 2: 0 DP 3: 0 DP 4: 5 DP 5: 3310 DP 6: 4160 DP 7: 64 DP 8: 0	DP 1: 0 DP 2: 0 DP 3: 0 DP 4: 9 DP 5: 7406 DP 6: 4160 DP 7: 64 DP 8: 0	DP 1: 0 DP 2: 0 DP 3: 0 DP 4: 13 DP 5: 11502 DP 6: 4160 DP 7: 64 DP 8: 0	DP 1: 0 DP 2: 0 DP 3: 0 DP 4: 17 DP 5: 15598 DP 6: 4160 DP 7: 64 DP 8: 0	DP 1: 0 DP 2: 0 DP 3: 0 DP 4: 21 DP 5: 19694 DP 6: 4160 DP 7: 64 DP 8: 0	DP 1: 0 DP 2: 0 DP 3: 0 DP 4: 25 DP 5: 23790 DP 6: 4160 DP 7: 64 DP 8: 0	DP 1: 0 DP 2: 0 DP 3: 0 DP 4: 29 DP 5: 27886 DP 6: 4746 DP 7: 64 DP 8: 0	DP 1: 0 DP 2: 0 DP 3: 0 DP 4: 29 DP 5: 27886 DP 6: 4746 DP 7: 64 DP 8: 0

Table E-25. 7957 Default File Values

	Cartridge 2	Cartridge 3	Cartridge 4	Cartridge 5	Cartridge 6	Cartridge 7
	M7957_CF:2	M7957_CF:3	M7957_CF:4	M7957_CF:5	M7957_CF:6	M7957_CF:7
	DD.33	DD.33	DD.33	DD.33	DD.33	DD.33
	33B	33B	33B	33B	33B	33B
	NONE	NONE	NONE	NONE	NONE	NONE
	72	72	72	72	72	72
	8	8	8	8	8	8
	DP 1: 0 DP 2: 0 DP 3: 0 DP 4: 0 DP 5: 64750 DP 6: 640 DP 7: 64 DP 8: 0	DP 1: 0 DP 2: 0 DP 3: 0 DP 4: 1 DP 5: 40174 DP 6: 640 DP 7: 64 DP 8: 0	DP 1: 0 DP 2: 0 DP 3: 0 DP 4: 2 DP 5: 15598 DP 6: 640 DP 7: 64 DP 8: 0	DP 1: 0 DP 2: 0 DP 3: 0 DP 4: 2 DP 5: 56558 DP 6: 640 DP 7: 64 DP 8: 0	DP 1: 0 DP 2: 0 DP 3: 0 DP 4: 3 DP 5: 31982 DP 6: 640 DP 7: 64 DP 8: 0	DP 1: 0 DP 2: 0 DP 3: 0 DP 4: 4 DP 5: 7406 DP 6: 774 DP 7: 64 DP 8: 0



	Cartridge 0	Cartridge 1
Model Number	M7957_CF:0	M7957_CF:1
Entry Point	DD.33	DD.33
Device Type	33B	33B
Buffer Limits	NONE	NONE
Table Extension	72	72
Number of Driver Area Defaults	8	8
Driver Parameter Area Defaults	DP 1: 0 DP 2: 0 DP 3: 0 DP 4: 0 DP 5: 0 DP 6: 400 DP 7: 48 DP 8: 0	DP 1: 0 DP 2: 0 DP 3: 0 DP 4: 0 DP 5: 19200 DP 6: 943 DP 7: 48 DP 8: 0

Table E-26. 7958 Default File Values

	Cartridge 0	Cartridge 1	Cartridge 2	Cartridge 3	Cartridge 4	Cartridge 5
	M7958_CF:0	M7958_CF:1	M7958_CF:2	M7958_CF:3	M7958_CF:4	M7958_CF:5
	DD.33	DD.33	DD.33	DD.33	DD.33	DD.33
	33B	33B	33B	33B	33B	33B
	NONE	NONE	NONE	NONE	NONE	NONE
	72	72	72	72	72	72
ar	8	8	8	8	8	8
ir	DP 1: 0 DP 2: 0 DP 3: 0 DP 4: 0 DP 5: 0 DP 6: 400 DP 7: 48 DP 8: 0	DP 1: 0 DP 2: 0 DP 3: 0 DP 4: 0 DP 5: 19200 DP 6: 943 DP 7: 48 DP 8: 0	DP 1: 0 DP 2: 0 DP 3: 0 DP 4: 0 DP 5: 64750 DP 6: 640 DP 7: 64 DP 8: 0	DP 1: 0 DP 2: 0 DP 3: 0 DP 4: 1 DP 5: 40174 DP 6: 640 DP 7: 64 DP 8: 0	DP 1: 0 DP 2: 0 DP 3: 0 DP 4: 2 DP 5: 15598 DP 6: 2560 DP 7: 64 DP 8: 0	DP 1: 0 DP 2: 0 DP 3: 0 DP 4: 2 DP 5: 48366 DP 6: 3125 DP 7: 64 DP 8: 0

Model Number
Entry Point
Device Type
Buffer Limits
Table Extension
Number of Drive Area Defaults
Driver Paramete Area Defaults

Default DVT Entries by Driver Name

This appendix summarizes DVT entries by driver name and model number. This information can be used as a quick cross-reference. You can easily find all models each driver supports, and specific information for each model number.

The tables are organized in the following manner: the driver name and information that is common for all model numbers is listed at the top of the table. Specific information for each model number is listed within each table. Refer to Chapter 6 of this manual for a description of the DVT command and parameters.

Driver	Eentry Point	TX	TO	QU	BL
DD*00	DD.00	32	0	FI	BU:100,400

Model	DT	DX	DP
2601	12B	12	1:2,0,10400B
2621	5	12	1:1,0,10400B,0,FM,GR,20040B,0,CO,MN,D,0
2631	12B	12	1:3,0,10400B
2635:0	0	12	1:3,0,10400B,0,FM,GR,20040B,0,CO,MN,D,0
2635:1	12B	12	1:3,0,10400B
2645	5	12	1:1,0,10400B,0,FM,GR,20040B,0,CO,MN,D,0
26xx	5	12	1:1,0,10400B,0,FM,GR,20040B,0,CO,MN,D,0
2675:0	5	12	1:3,0,10400B,0,FM,GR,20040B,0,CO,MN,D,0
2675:1	12B	12	1:3,0,10400B
256x	12B	12	1:3,0,10400B
2225	12B	12	1:3,0,10400B
2686	12B	12	1:3,0,10400B
755x	5	12	1:3,0,10400B
758x	5	12	1:3,0,10400B

Driver	Eentry Point	TX	TO
DD*12	DD.12	36	1000

Model	DT	DX	DP
2631	12B	2	2:0
2673A			2:0
2697B			2:1
2932A			2:1

Driver	Eentry Point	TX	TO	QU	DT	DX
DD*20	DD.20	45	3000	FI	20B	1

Model	DP
2645:1	1:1
2645:2	1:2
264X:1	1:1
264X:2	1:2

Driver	Eentry Point	TX	TO	DT	DX
DD*23	DD.23	30	1000	23B	3

Model	DP
7970E:0	2:400B,0
7970E:1	2:1000B,0
7970E:2	2:1400B,0
7970E:3	2:2000B,0

Driver	Eentry Point	TX	DX	QU	BL	PR	TO	DT
DD*24	DD.24	43	5	FI	UN,0,0	0	500	24B

Model	DP
7974	1:4,1600
7978	1:4,1600

In the following table, the recommended DVT entries for new systems are highlighted.

Driver	Entry Point	TX	DX	QU	BL	PR
DD*30	DD.30	25	8	-	-	-

Model	TO	DT	DP
9121:0	3000	30B	2:0,0,0,2,66,16,2
9121:1	3000	30B	2:1,0,0,2,66,16,2
7902:0	750	30B	2:0,0,0,3,134,30,2
7902:1	750	30B	2:1,0,0,3,134,30,2
9134:0	750	30B	2:0,0,0,3,134,30,2
9134:1	750	30B	2:1,0,0,3,134,30,2
9134:2	750	30B	2:2,0,0,3,134,30,2
9134:3	750	30B	2:3,0,0,3,134,30,2
9134L:0	750	30B	2:0,0,0,0,472,31,4
9134L:1	750	30B	2:0,0,118,0,136,31,4
9134L:2	750	30B	2:0,0,152,0,612,31,4
9134X:0	750	30B	2:0,0,0,0,624,31,6
9134X:1	750	30B	2:0,0,104,0,330,31,6
9134X:2	750	30B	2:0,0,159,0,462,31,6
9134X:3	750	30B	2:0,0,236,0,412,31,6
7906:0	3000	32B	2:0,0,0,5,406,48,1
7906:1	3000	32B	2:0,1,0,5,406,48,1
7906:2	3000	32B	2:0,2,0,5,406,48,1
7906:3	3000	32B	2:0,3,0,5,406,48,1
7910:0	1000	31B	2:0,0,0,4,370,32,2
7910:1	1000	31B	2:0,0,187,2,370,32,2
7910:2	1000	31B	2:0,0,373,2,370,32,2
7910:3	1000	31B	2:0,0,559,4,370,32,2
7920:0	3000	32B	2:0,0,0,9,506,48,5
7920:1	3000	32B	2:0,0,83,9,406,48,5
7920:2	3000	32B	2:0,0,166,9,406,48,5
7920:3	3000	32B	2:0,0,249,9,406,48,5
7920:4	3000	32B	2:0,0,332,9,406,48,5
7920:5	3000	32B	2:0,0,415,9,406,48,5
7920:6	3000	32B	2:0,0,498,9,406,48,5
7920:7	3000	32B	2:0,0,581,9,406,48,5
7920:8	3000	32B	2:0,0,664,9,406,48,5
7920:9	3000	32B	2:0,0,747,10,370,48,5
7925:0	3000	32B	2:0,0,0,9,549,64,9
7925:1	3000	32B	2:0,0,62,9,549,64,9
7925:2	3000	32B	2:0,0,124,9,549,64,9
7925:3	3000	32B	2:0,0,186,9,549,64,9
7925:4	3000	32B	2:0,0,248,11,1024,64,9
7925:5	3000	32B	2:0,0,363,11,1024,64,9
7925:6	3000	32B	2:0,0,478,11,1024,64,9
7925:7	3000	32B	2:0,0,593,11,1024,64,9
7925:8	3000	32B	2:0,0,708,11,1024,64,9

In the following table, the recommended DVT entries for new systems are highlighted.

Driver DD*33	Entry Point DD.33	TX 72	DX 8	QU FI	BL UN,0,0	PR 0	TO 500		
Model	DT	DP							
Tape	26B	2:400B	100000B	0	64464	0	0	0	
9144	26B	2:0	0	0	0	0	0	0	
7908:0	26B	2:0	0	0	0	400	48	0	
7908:1	33B	2:0	0	0	19200	214	48	0	
7908:2	33B	2:0	0	0	29472	300	48	0	
7908:3	33B	2:0	0	0	43872	214	48	0	
7908:4	33B	2:0	0	0	54144	215	48	0	
7908_LF:0	33B	2:0	0	0	0	400	48	0	
7908_LF:1	33B	2:0	0	0	19200	214	48	0	
7908_LF:2	33B	2:0	0	0	29472	300	48	0	
7908_LF:3	33B	2:0	0	0	43872	214	48	0	
7908_LF:4	33B	2:0	0	0	54144	215	48	0	
7908_CF:0	33B	2:0	0	0	0	400	48	0	
7908_CF:1	33B	2:0	0	0	19200	943	48	0	
7907FXD_CF:0	33B	2:0	0	0	0	400	48	0	
7907FXD_CF:1	33B	2:0	0	0	19200	943	48	0	
7907FXD_CF:2	33B	2:0	0	0	64750	320	48	0	
7907REM_CF:0	33B	2:400B	0	0	0	400	48	0	
7907REM_CF:1	33B	2:400B	0	0	19200	943	48	0	
7907REM_CF:2	33B	2:400B	0	0	64750	320	48	0	
7941:0	33B	2:0	0	0	0	400	48	0	
7941:1	33B	2:0	0	0	19200	214	48	0	
7941:2	33B	2:0	0	0	29472	300	48	0	
7941:3	33B	2:0	0	0	43872	214	48	0	
7941:4	33B	2:0	0	0	54144	215	48	0	
7941:5	33B	2:0	0	0	64750	400	48	0	
7941:6	33B	2:0	0	1	18414	539	48	0	
7941:7	33B	2:0	0	1	44288	424	48	0	
7941_LF:0	33B	2:0	0	0	0	400	48	0	
7941_LF:1	33B	2:0	0	0	19200	214	48	0	
7941_LF:2	33B	2:0	0	0	29472	300	48	0	
7941_LF:3	33B	2:0	0	0	43872	214	48	0	
7941_LF:4	33B	2:0	0	0	54144	215	48	0	
7941_LF:5	33B	2:0	0	0	64750	939	48	0	
7941_LF:6	33B	2:0	0	1	44288	424	48	0	
7941_CF:0	33B	2:0	0	0	0	400	48	0	
7941_CF:1	33B	2:0	0	0	19200	943	48	0	
7941_CF:2	33B	2:0	0	0	64750	320	48	0	
7941_CF:3	33B	2:0	0	1	14592	266	48	0	
7911:0	33B	2:0	0	0	0	400	48	0	
7911:1	33B	2:0	0	0	19200	214	48	0	
7911:2	33B	2:0	0	0	29472	300	48	0	
7911:3	33B	2:0	0	0	43872	214	48	0	
7911:4	33B	2:0	0	0	54144	215	48	0	
7911:5	33B	2:0	0	0	64750	400	48	0	
7911:6	33B	2:0	0	1	18414	539	48	0	
7911_LF:0	33B	2:0	0	0	0	400	48	0	
7911_LF:1	33B	2:0	0	0	19200	214	48	0	
7911_LF:2	33B	2:0	0	0	29472	300	48	0	
7911_LF:3	33B	2:0	0	0	43872	214	48	0	
7911_LF:4	33B	2:0	0	0	54144	215	48	0	
7911_LF:5	33B	2:0	0	0	64750	939	48	0	
7911_CF:0	33B	2:0	0	0	0	400	48	0	
7911_CF:1	33B	2:0	0	0	19200	943	48	0	
7911_CF:2	33B	2:0	0	0	64750	320	48	0	
7911_CF:3	33B	2:0	0	1	14592	266	48	0	
7911_CF:4	33B	2:0	0	1	27392	352	48	0	

F-4 Default DVT Entries by Driver Name

In the following table, the recommended DVT entries for new systems are highlighted.

Driver	Eentry Point	TX	DX	QU	BL	PR	TO
DD*33	DD.33	72	8	FI	UN,0,0	0	500



Model	DT	DP						
7945_CF:0	33B	2:0	0	0	0	400	48	0
7945_CF:1	33B	2:0	0	0	19200	943	48	0
7945_CF:2	33B	2:0	0	0	64750	320	48	0
7945_CF:3	33B	2:0	0	1	14592	266	48	0
7945_CF:4	33B	2:0	0	1	27392	352	48	0
7945_CF:5	33B	2:0	0	1	44288	2229	48	0
7912:0	33B	2:0	0	0	0	400	48	0
7912:1	33B	2:0	0	0	19200	214	48	0
7912:2	33B	2:0	0	0	29472	300	48	0
7912:3	33B	2:0	0	0	43872	214	48	0
7912:4	33B	2:0	0	0	54144	215	48	0
7912:5	33B	2:0	0	0	64750	400	48	0
7912:6	33B	2:0	0	1	18414	539	48	0
7912:7	33B	2:0	0	1	44288	924	48	0
7912:8	33B	2:0	0	1	64640	340	48	0
7912:9	33B	2:0	0	2	15424	764	48	0
7912:10	33B	2:0	0	2	52096	764	48	0
7912:11	33B	2:0	0	3	23232	758	48	0
7912_LF:0	33B	2:0	0	0	0	400	48	0
7912_LF:1	33B	2:0	0	0	19200	214	48	0
7912_LF:2	33B	2:0	0	0	29472	300	48	0
7912_LF:3	33B	2:0	0	0	43872	214	48	0
7912_LF:4	33B	2:0	0	0	54144	215	48	0
7912_LF:5	33B	2:0	0	0	64750	939	48	0
7912_LF:6	33B	2:0	0	1	44288	424	48	0
7912_LF:7	33B	2:0	0	1	64640	2626	48	0
7912_CF:0	33B	2:0	0	0	0	400	48	0
7912_CF:1	33B	2:0	0	0	19200	943	48	0
7912_CF:2	33B	2:0	0	0	64750	320	48	0
7912_CF:3	33B	2:0	0	1	14592	266	48	0
7912_CF:4	33B	2:0	0	1	27392	352	48	0
7912_CF:5	33B	2:0	0	1	44288	2229	48	0
7912_CF:6	33B	2:0	0	3	20223	821	48	0
7914:0	33B	2:0	0	0	0	400	48	0
7914:1	33B	2:0	0	0	19200	214	48	0
7914:2	33B	2:0	0	0	29472	300	48	0
7914:3	33B	2:0	0	0	43872	214	48	0
7914:4	33B	2:0	0	0	54144	215	48	0
7914:5	33B	2:0	0	0	64750	400	48	0
7914:6	33B	2:0	0	1	18414	539	48	0
7914:7	33B	2:0	0	1	44288	424	48	0
7914:8	33B	2:0	0	1	64640	340	48	0
7914:9	33B	2:0	0	2	15424	764	48	0
7914:10	33B	2:0	0	2	52096	764	48	0
7914:11	33B	2:0	0	3	23232	750	48	0
7914:12	33B	2:0	0	3	59616	902	48	0
7914:13	33B	2:0	0	4	37376	902	48	0
7914:14	33B	2:0	0	5	15136	902	48	0
7914:15	33B	2:0	0	5	58432	902	48	0
7914:16	33B	2:0	0	6	36192	902	48	0
7914:17	33B	2:0	0	7	13952	904	48	0

In the following table, the recommended DVT entries for new systems are highlighted.

Driver	Eentry Point	TX	DX	QU	BL	PR	TO
DD*33	DD.33	72	8	FI	UN,0,0	0	500

Model	DT	DP						
7914_LF:0	33B	2:0	0	0	0	400	48	0
7914_LF:1	33B	2:0	0	0	19200	214	48	0
7914_LF:2	33B	2:0	0	0	29472	300	48	0
7914_LF:3	33B	2:0	0	0	43872	214	48	0
7914_LF:4	33B	2:0	0	0	54144	215	48	0
7914_LF:5	33B	2:0	0	0	64750	939	48	0
7914_LF:6	33B	2:0	0	1	44288	424	48	0
7914_LF:7	33B	2:0	0	1	64640	2626	48	0
7914_LF:8	33B	2:0	0	3	59616	4510	48	0
7914_LF:9	33B	2:0	0	7	13952	904	48	0
7914_Cf:0	33B	2:0	0	0	0	400	48	0
7914_Cf:1	33B	2:0	0	0	19200	943	48	0
7914_Cf:2	33B	2:0	0	0	64750	320	48	0
7914_Cf:3	33B	2:0	0	1	14592	266	48	0
7914_Cf:4	33B	2:0	0	1	27392	352	48	0
7914_Cf:5	33B	2:0	0	1	44288	2229	48	0
7914_Cf:6	33B	2:0	0	3	20223	821	48	0
7914_Cf:7	33B	2:0	0	3	59648	5413	48	0
7933:0	33B	2:0	0	0	0	400	48	0
7933:1	33B	2:0	0	0	19200		48	0
7933:2	33B	2:0	0	0	29472		48	0
7933:3	33B	2:0	0	0	43872		48	0
7933:4	33B	2:0	0	0	54144		48	0
7933:5	33B	2:0	0	0	64750		48	0
7933:6	33B	2:0	0	1	18414		48	0
7933:7	33B	2:0	0	1	44288		48	0
7933:8	33B	2:0	0	1	64640		48	0
7933:9	33B	2:0	0	2	15424		48	0
7933:10	33B	2:0	0	2	52096		48	0
7933:11	33B	2:0	0	3	23232		48	0
7933:12	33B	2:0	0	3	58616		48	0
7933:13	33B	2:0	0	4	37376		48	0
7933:14	33B	2:0	0	5	15136	902	48	0
7933:15	33B	2:0	0	5	58432	902	48	0
7933:16	33B	2:0	0	6	36192	902	48	0
7933:17	33B	2:0	0	7	13952	904	48	0
7933:18	33B	2:0	0	7	57344	1477	48	0
7933:19	33B	2:0	0	8	62704	1477	48	0
7933:20	33B	2:0	0	10	2528	1477	48	0
7933:21	33B	2:0	0	11	7888	1477	48	0
7933:22	33B	2:0	0	12	13248	1477	48	0
7933:23	33B	2:0	0	13	18608	1477	48	0
7933:24	33B	2:0	0	14	23968	1477	48	0
7933:25	33B	2:0	0	15	29328	1477	48	0
7933:26	33B	2:0	0	16	34688	1477	48	0
7933:27	33B	2:0	0	18	40048	1477	48	0
7933:28	33B	2:0	0	19	45408	1477	48	0
7933:29	33B	2:0	0	19	50768	1477	48	0
7933:30	33B	2:0	0	20	56128	1477	48	0
7933:31	33B	2:0	0	21	61488	1477	48	0
7933:32	33B	2:0	0	23	1312	1477	48	0

In the following table, the recommended DVT entries for new systems are highlighted.

Driver	Eentry Point	TX	DX	QU	BL	PR	TO
DD*33	DD.33	72	8	FI	UN,0,0	0	500

Model	DT	DP						
7933_LF:0	33B	2:0	0	0	0	400	48	0
7933_LF:1	33B	2:0	0	0	19200	214	48	0
7933_LF:2	33B	2:0	0	0	29427	300	48	0
7933_LF:3	33B	2:0	0	0	43872	214	48	0
7933_LF:4	33B	2:0	0	0	54144	215	48	0
7933_LF:5	33B	2:0	0	0	64750	939	48	0
7933_LF:6	33B	2:0	0	1	44288	424	48	0
7933_LF:7	33B	2:0	0	1	64640	2626	48	0
7933_LF:8	33B	2:0	0	3	59616	4510	48	0
7933_LF:9	33B	2:0	0	7	13952	904	48	0
7933_LF:10	33B	2:0	0	7	57344	8862	48	0
7933_LF:11	33B	2:0	0	14	23968	13300	48	0
7933_CF:0	33B	2:0	0	0	0	400	48	0
7933_CF:1	33B	2:0	0	0	19200	943	48	0
7933_CF:2	33B	2:0	0	0	64750	320	48	0
7933_CF:3	33B	2:0	0	1	14592	266	48	0
7933_CF:4	33B	2:0	0	1	27392	352	48	0
7933_CF:5	33B	2:0	0	1	44288	2229	48	0
7933_CF:6	33B	2:0	0	3	20223	821	48	0
7933_CF:7	33B	2:0	0	3	59648	5413	48	0
7933_CF:8	33B	2:0	0	7	57344	8862	48	0
7933_CF:9	33B	2:0	0	14	23968	13300	48	0
7935:0	33B	2:0	0	0	0	400	48	0
7935:1	33B	2:0	0	0	19200	214	48	0
7935:2	33B	2:0	0	0	29472	300	48	0
7935:3	33B	2:0	0	0	43872	214	48	0
7935:4	33B	2:0	0	0	54144	215	48	0
7935:5	33B	2:0	0	0	64750	400	48	0
7935:6	33B	2:0	0	1	18414	539	48	0
7935:7	33B	2:0	0	1	44288	424	48	0
7935:8	33B	2:0	0	1	64640	340	48	0
7935:9	33B	2:0	0	2	1542	764	48	0
7935:10	33B	2:0	0	2	52096	764	48	0
7935:11	33B	2:0	0	3	23232	758	48	0
7935:12	33B	2:0	0	3	59616	902	48	0
7935:13	33B	2:0	0	4	37376	902	48	0
7935:14	33B	2:0	0	5	15136	902	48	0
7935:15	33B	2:0	0	5	58432	902	48	0
7935:16	33B	2:0	0	6	36192	902	48	0
7935:17	33B	2:0	0	7	13592	904	48	0
7935:18	33B	2:0	0	7	57344	1477	48	0
7935:19	33B	2:0	0	8	62704	1477	48	0
7935:20	33B	2:0	0	10	2528	1477	48	0
7935:21	33B	2:0	0	11	7888	1477	48	0
7935:22	33B	2:0	0	12	13248	1477	48	0
7935:23	33B	2:0	0	13	18608	1477	48	0
7935:24	33B	2:0	0	14	23968	1477	48	0
7935:25	33B	2:0	0	15	29328	1477	48	0
7935:26	33B	2:0	0	16	35688	1477	48	0
7935:27	33B	2:0	0	17	40048	1477	48	0
7935:28	33B	2:0	0	18	45408	1477	48	0
7935:29	33B	2:0	0	19	50768	1477	48	0
7935:30	33B	2:0	0	20	56128	1477	48	0
7935:31	33B	2:0	0	21	61488	1477	48	0
7935:32	33B	2:0	0	23	1312	1477	48	0

Driver	Eentry Point	TX	DX	QU	BL	PR	TO
DD*33	DD.33	72	8	FI	UN,0,0	0	500

Model	DT	DP						
7935_LF:0	33B	2:0	0	0	0	400	48	0
7935_LF:1	33B	2:0	0	0	19200	214	48	0
7935_LF:2	33B	2:0	0	0	29472	300	48	0
7935_LF:3	33B	2:0	0	0	43872	214	48	0
7935_LF:4	33B	2:0	0	0	54144	215	48	0
7935_LF:5	33B	2:0	0	0	64750	939	48	0
7935_LF:6	33B	2:0	0	1	44288	424	48	0
7935_LF:7	33B	2:0	0	1	64640	2626	48	0
7935_LF:8	33B	2:0	0	3	59616	5410	48	0
7935_LF:9	33B	2:0	0	7	13952	904	48	0
7935_LF:10	33B	2:0	0	7	47344	8862	48	0
7935_LF:11	33B	2:0	0	14	23968	13300	48	0
7936_CF:0	33B	2:0	0	0	0	400	48	0
7936_CF:1	33B	2:0	0	0	19200	943	48	0
7936_CF:2	33B	2:0	0	0	64750	2560	64	0
7936_CF:3	33B	2:0	0	3	31982	2560	64	0
7936_CF:4	33B	2:0	0	5	64750	2560	64	0
7936_CF:5	33B	2:0	0	8	31982	2560	64	0
7936_CF:6	33B	2:0	0	10	64750	2560	64	0
7936_CF:7	33B	2:0	0	13	31982	4968	64	0
7937_CF:0	33B	2:0	0	0	0	400	48	0
7937_CF:1	33B	2:0	0	0	19200	943	48	0
7937_CF:2	33B	2:0	0	0	64750	4160	64	0
7937_CF:3	33B	2:0	0	5	3310	4160	64	0
7937_CF:4	33B	2:0	0	9	7406	4160	64	0
7937_CF:5	33B	2:0	0	13	11502	4160	64	0
7937_CF:6	33B	2:0	0	17	15598	4160	64	0
7937_CF:7	33B	2:0	0	21	19694	4160	64	0
7937_CF:8	33B	2:0	0	25	23790	4160	64	0
7937_CF:9	33B	2:0	0	29	27886	4746	64	0
7940:0	33B	2:0	0	0	0	400	48	0
7940:1	33B	2:0	0	0	19200	214	48	0
7940:2	33B	2:0	0	0	29472	300	48	0
7940:3	33B	2:0	0	0	43872	214	48	0
7940:4	33B	2:0	0	0	54144	215	48	0
7940_LF:0	33B	2:0	0	0	0	400	48	0
7940_LF:1	33B	2:0	0	0	19200	214	48	0
7940_LF:2	33B	2:0	0	0	29472	300	48	0
7940_LF:3	33B	2:0	0	0	43872	214	48	0
7940_LF:4	33B	2:0	0	0	54144	215	48	0
7957_CF:0	33B	2:0	0	0	0	400	48	0
7957_CF:1	33B	2:0	0	0	19200	943	48	0
7957_CF:2	33B	2:0	0	0	64750	640	64	0
7957_CF:3	33B	2:0	0	1	40174	640	64	0
7957_CF:4	33B	2:0	0	2	15598	640	64	0
7957_CF:5	33B	2:0	0	2	56558	640	64	0
7957_CF:6	33B	2:0	0	3	31982	640	64	0
7957_CF:7	33B	2:0	0	4	7406	774	64	0
7958_CF:0	33B	2:0	0	0	0	400	48	0
7958_CF:1	33B	2:0	0	0	19200	943	48	0
7958_CF:2	33B	2:0	0	0	64750	640	64	0
7958_CF:3	33B	2:0	0	1	40174	640	64	0
7958_CF:4	33B	2:0	0	2	15598	2560	64	0
7958_CF:5	33B	2:0	0	4	48366	3125	64	0
9133_CF:0	33B	2:0	0	0	0	604	48	0
9133_CF:1	33B	2:0	0	0	28992	604	48	0
9133_CF:2	33B	2:0	0	0	57974	413	48	0

F-8 Default DVT Entries by Driver Name

Driver	Eentry Point	TX	DX	QU	BL	PR	TO
DD*33	DD.33	72	8	FI	UN,0,0	0	500

Model	DT	DP							
9153_CF:0	33B	2:0	0	0	0	604	48	0	
9153_CF:1	33B	2:0	0	0	28992	604	48	0	
9153_CF:2	33B	2:0	0	0	57974	413	48	0	

Model	DT	DP							
For 9122									
M_FLOPPY_CF:0	33B		0	0	0 0	154	16	0	
For 9122/9133									
M_FLOPPY_CF:1	33B		400B	0	0 0	154	16	0	
9133_CF:3	33B		0	0	1 12272	1622	48	0	

Driver	Eentry Point	TX	DX	QU	BL	PR
DD*36	DD.36	25	8	-	-	-

Model	DT	DP							
500	36B	2:0	0	0	0	64	4	1	

Driver	Eentry Point	TX	DX	QU	BL	PR
DDC12	DDC12	98	8	-	-	-

Model	DT	DP							
500	12B	2:1	0	0	0	0	0	0	500
2608S		8:500							
2563		8:500							
2566		8:500							

Driver	Eentry Point	TX	DX	QU	BL	PR
DDM30	DDM30	51	8	-	-	-

Model	TO	DT	DP						
7902:0	750	30B	2:0	0	0	3	134	30	2
7902:1	750	30B	2:1	0	0	3	134	30	2
9134:0	750	30B	2:0	0	0	3	134	30	2
9134:1	750	30B	2:1	0	0	3	134	30	2
9134:2	750	30B	2:2	0	0	3	134	30	2
9134:3	750	30B	2:3	0	0	3	134	30	2
9134L:0	750	30B	2:0	0	0	0	472	31	4
9134L:1	750	30B	2:0	0	118	136	0	31	4
9134L:2	750	30B	2:0	0	152	612	0	31	4
7906:0	3000	32B	2:0	0	0	5	406	48	1
7906:1	3000	32B	2:0	1	0	5	406	48	1
7906:2	3000	32B	2:0	2	0	5	406	48	1
7906:3	3000	32B	2:0	3	0	5	406	48	1
7910:0	1000	32B	2:0	0	0	4	370	32	2
7910:1	1000	32B	2:0	0	187	2	370	32	2
7910:2	1000	32B	2:0	0	373	2	370	32	2
7910:3	1000	32B	2:0	0	559	4	370	32	2
7920:0	3000	32B	2:0	0	0	9	406	48	5
7920:1	3000	32B	2:0	0	83	9	406	48	5
7920:2	3000	32B	2:0	0	166	9	406	48	5
7920:3	3000	32B	2:0	0	249	9	406	48	5
7920:4	3000	32B	2:0	0	332	9	406	48	5
7920:5	3000	32B	2:0	0	415	9	406	48	5
7920:6	3000	32B	2:0	0	498	9	406	48	5
7920:7	3000	32B	2:0	0	581	9	406	48	5
7920:8	3000	32B	2:0	0	664	9	406	48	5
7920:9	3000	32B	2:0	0	747	10	370	48	5
7925:0	3000	32B	2:0	0	0	9	549	64	9
7925:1	3000	32B	2:0	0	62	9	549	64	9
7925:2	3000	32B	2:0	0	124	9	549	64	9
7925:3	3000	32B	2:0	0	186	9	549	64	9
7925:4	3000	32B	2:0	0	248	11	1024	64	9
7925:5	3000	32B	2:0	0	363	11	1024	64	9
7925:6	3000	32B	2:0	0	478	11	1024	64	9
7925:7	3000	32B	2:0	0	593	11	1024	64	9
7925:8	3000	32B	2:0	0	708	11	1024	64	9

Driver	Eentry TX Point	DX	QU	BL	PR	TO	DT
GEN27	-	9	-	-	-	5000	30B

Model	DP								
10 Mbyte Fixed									
2480:0	1:0	0	0	0	0	472	31	4	0
2480:1	1:0	0	0	118	0	136	31	4	0
2480:2	1:0	0	0	152	0	616	31	4	0
Floppy									
2480:3	1:3	0	0	0	0	66	16	1	0
15 Mbyte Fixed									
2480:4	1:0	0	0	0	0	624	31	6	0
2480:5	1:0	0	0	104	0	330	31	6	0
2480:6	1:0	0	0	159	0	462	31	6	0
2480:7	1:0	0	0	236	0	420	31	6	0
20 Mbyte Fixed									
2480:11	1:0	0	0	0	0	1956	31	4	0
2480:12	1:0	0	0	489	0	476	31	4	0
15 Mbyte Fixed									
2480:13	1:0	0	0	104	0	1212	31	6	0
Floppy									
2480:14	1:3	0	0	0	0	154	16	2	0



Sample System Generation Answer File

The sample system generation answer file in this appendix contains commands for loading system utilities, configuring most supported I/O devices, and defining system parameters. Commands to include the optional products DS/1000 (91750A) and VC+ (92078A) are also specified.

The sample answer file can be used as the basis for a new system generation. The file is available on your Primary System under the name #ANS. Entries can be added or deleted as required to generate a new system. Before changing the answer file to create a new answer file, you should be thoroughly familiar with system generation concepts and the syntax and effects of the RTAGN commands given in Chapters 1 through 7 of this manual.

```
*
*
*   Source: 92077-17197  REV.5000 <870708.1754>
*
*   RTE-A Primary System Generation Answer File
*
*   #ANS:
*   Makes PRIMARY.SNP (#92077-16954), PRIMARY.SYS (#92077-16955)
*       and PRIMARY.LST (#92077-17278).
*
*
*   *****
*   *
*   *   This half of the primary answer file is used only for primary   *
*   *   system generation.  It is NOT to be used with a user system.   *
*   *   To use this answer file as a sample for generating another     *
*   *   system, delete the first half of this text, and continue with   *
*   *   the instructions on the second half.                             *
*   *
*   *****
*
*
*   MODIFICATIONS:
*   5.00: 4/10/87
*
*       Removed definition for LU1.  With 5.00 version of the
*       operating system, the generator links no LU1 to the bit
*       bucket.  The program LUCFG will now overwrite this area
*       with the correct mapping, but the RTE-A READY message will
*       not be sent due to no LU 1.  The ASIC card is now LU 100.
*       Change LU 85 to address 3 (was 2)
```

```

*
*
* This answer file is to be used in a PRIMARY memory based system. There
* is nothing generated into LU 1. It is required that the startup program
* be LUCFG.RUN, which maps the I/O card which has VCP enabled to the correct
* select code, and the correct interface driver to LU 1. This generation
* requires: LU1: ID100 at select code 20B (double-mapped if VCP port)
*           LU110: ID400 at select code 77B (double-mapped if VCP port)
*           LU120: ID800 at select code 30B (double-mapped if VCP port)
*           LU130: IDM00 at select code 23B (double-mapped if VCP port)
*
* Note that if mapping occurs, do not use the previous LU. (i.e., if
* 12100A is VCP port, LU 110 should not be used, since now there are
* two LU's pointing to the same hardware.
*

```

I / O D E F I N I T I O N S

LU#s	DEVICE	SELECT CODE	ADDRESS
HP-IB #1 =====			
15	HP-IB #1 controller	27B	36B
16-23	7907/08/11/12/14/		
25-26	33/35/41/42/45/46/	27B	0
29-31	and other CS/80 discs		
24	CS/80 compatible cartridge tape cache	27B	0
9	9144 stand-alone CS/80 cartridge tape	27B	1
50-52	9153 3.5" fixed disc	27B	2
53	9153 3.5" flexible disc single sided	27B	2
54	9153 3.5" flexible disc double sided	27B	2
7	7974A/7978A Streaming tape drive	27B	3
8	7970E HP-IB tape drive	27B	4
32-33	9121 3.5" flexible disc	27B	6
HP-IB #2 =====			
40	HP-IB #2 controller	26B	36B
44-47	9133XV/4XV 5.25" Winchester fixed disc	26B	0
12	9144 stand-alone CS/80 cartridge tape	26B	1
34-35	9121 3.5" flexible disc	26B	2
85	2608S,2563A,2566A line printer	26B	3
6	2932A line printer	26B	6
41-43	9133H/4H 5.25" Winchester fixed disc	26B	7
61	9133H 3.5" flexible disc	26B	7


```

re %id*43:92077,,      Power fail driver
re %signl:92077,,     Signals module
re %math:92077,.dmp, Required for A400/A600 without CDS (RPL40/RPL60)
re %math:92077,.ddi, Required for A400/A600 without CDS (RPL40/RPL60)
re %gen27:92077,,     Gen records only for 248X (show time stamp)
*
se $syslb:92077,,Search the system library
*
* define partitionable modules
*
pa perr,xcmdnd,stat,dsq,vema,lock,load,memry,iorq
pa time,class,abort,alarm
*
*
ms $sysa:92077,, Search the system dummy library
end,                Terminate system relocation phase
*
*
* OS partitions
*
re %load:92077,,      Program loading and swapping
end
*
re %memry:92077,,     Memory management module
end
*
re %iorq:92077,,      I/O request processing
end
*
re %lock:92077,,      LU locking and resource numbers
end
*
re %time:92077,,      Time scheduling of programs
re %class:92077,,     Class I/O module
end
*
re %xcmdnd:92077,,    Operator command extension module
re %stat:92077,,      Status command module
re %dsq:92077,,       DS/1000 module
end
*
re %perr:92077,,      Parity error handler
re %vema:92077,,      Virtual memory module
end
*
re %alarm:92077,,     Alarm signal processing module
end
*
re %abort:92077,,     Abort processing module
end
*
*
* Driver partitions
*
re %dd*33:92077,,     Device driver for CS/80 discs
end

```

```

*
re %id*66:91750,,      DS/1000: interface driver for DS/1000
end
*
re %id*37:92077,,      Interface driver for CS/80 discs
end
*
re id400.rel:92077,,    Interface driver for A400/OBIO
end
*
re %idm00:92077,,      Interface driver for 12040 rev A-C
re %dd*23:92077,,      Device driver for magnetic tape de
end
*
re %dd*00:92077,,      Device driver for %idm00:92077
re %dd*30:92077,,      Device driver for discs (ICD)
end
*
re %adv00:91750,,      DS/1000:device driver for DS/1000
end
*
re id800.rel:92077,,    Interface driver for 12040D 8 channel MUX
re %dd*12:92077,,      Device driver for HP-IB line printer
end
*
re ddc00.rel:92077,,    Device driver for 100 series interfaces
end
*
re %ddc12:92077,,      Device driver for HP-IB line printer
end
*
re %id*27:92077,,      Interface driver for 248X integrated disc
re %dd*24:92077,,      Device driver for 7974A/7978A streaming tape
end
*
re id100.rel:92077,,    Interface driver for 12005 ASIC
end
*
end ,,                  END DRIVER PARTITION PHASE
*
*
*
* Table Generation -- configure LU tables
*
*
*
* HP-IB #1 -- discs and magnetic tape          select code = 27b
*
*
ift,%id*37:92077,SC:27B
*
* Bus Controller LU                               HP-IB address 36b
*
dvt,,,LU:15,TO:2000,DT:77B,TX:0,DX:1,DP:1:36B,PR:0
*
* 7907/08/11/12/14/33/35/41/42/45/46 CS/80 discs
*
*                                               HP-IB address 0
*                                               LU 16-23,25-26,

```



```

*
dvt,%dd*33:92077,M7908_CF:0,LU:16,DP:1:0
dvt,%dd*33:92077,M7908_CF:1,LU:17,DP:1:0
dvt,%dd*33:92077,M7907FXD_CF:2,LU:18,DP:1:0
dvt,%dd*33:92077,M7941_CF:3,LU:19,DP:1:0
dvt,%dd*33:92077,M7911_CF:4,LU:20,DP:1:0
dvt,%dd*33:92077,M7945_CF:5,LU:21,DP:1:0
dvt,%dd*33:92077,M7912_CF:6,LU:22,DP:1:0
dvt,%dd*33:92077,M7914_CF:7,LU:23,DP:1:0
dvt,%dd*33:92077,M7933_CF:8,LU:25,DP:1:0
dvt,%dd*33:92077,M7933_CF:9,LU:26,DP:1:0
dvt,%dd*33:92077,M7907REM_CF:0,LU:29,DP:1:0
dvt,%dd*33:92077,M7907REM_CF:1,LU:30,DP:1:0
dvt,%dd*33:92077,M7907REM_CF:2,LU:31,DP:1:0
*
*
* CS/80 compatible cartridge tape cache                HP-IB address 0
*                                                         LU 24
*
dvt,%dd*33:92077,MTAPE,LU:24,DP:1:0
*
*
* 9144 stand-alone CS/80 cartridge tape                HP-IB address 1
*                                                         LU 9
*
dvt,%dd*33:92077,M9144:0,LU:9,DP:1:1
*
*
* 3.5" flexible disc single sided (9153/54)           HP-IB address 2
*                                                         LU 53
*
dvt,%dd*33:92077,M_FLOPPY_CF:1,LU:53,DP:1:2:400B:0:0:0,-
DP:6:66:16:0
*
*
* 3.5" flexible disc (9153/4)                         HP-IB address 2
*                                                         LU 54
*
dvt,%dd*33:92077,M_FLOPPY_CF:1,LU:54,DP:1:2
*
*
* 3.5" fixed disc (9153/54)                           HP-IB address 2
*                                                         LU 50-52
*
dvt,%dd*33:92077,M9133_CF:0,LU:50,DP:1:2
dvt,%dd*33:92077,M9133_CF:1,LU:51,DP:1:2
dvt,%dd*33:92077,M9133_CF:2,LU:52,DP:1:2,DP:5:57984
*
*
* 7974A/7978A streaming tape drive                   HP-IB address 3
*                                                         LU 7
*
dvt,%dd*24:92077,M7974:0,LU:7,DP:1:3
*
*
* HP-IB tape drive                                    HP-IB address 4
*                                                         LU 8
*

```



```
*
dvt,%dd*23:92077,M7970E:0,LU:8,DP:1:4,PR:1
*
*
*
* 5.25" and 3.5" flexible discs (9121)
*
*
*
dvt,%dd*30:92077,M9121:0,LU:32,DP:1:6,TO:3000
dvt,%dd*30:92077,M9121:1,LU:33,DP:1:6,TO:3000
*
*
*
* HP-IB #2 -- discs and magnetic tape
*
*
*
ift,%id*37:92077,SC:26B
*
* Bus Controller LU
*
*
dvt,, ,LU:40,TO:2000,DT:77B,TX:0,DX:1,DP:1:36B,PR:0
*
* 5.25" Winchester fixed disc (9133/4 XV)
*
*
dvt,%dd*30:92077,M9134X:0,LU:44,DP:1:0
dvt,%dd*30:92077,M9134X:1,LU:45,DP:1:0
dvt,%dd*30:92077,M9134X:2,LU:46,DP:1:0
dvt,%dd*30:92077,M9134X:3,LU:47,DP:1:0
*
* 9144 stand-alone CS/80 cartridge tape
*
*
dvt,%dd*33:92077,M9144:0,LU:12,DP:1:1
*
* 3.5" flexible discs (9121)
*
*
dvt,%dd*30:92077,M9121:0,LU:34,DP:1:2,TO:3000
dvt,%dd*30:92077,M9121:1,LU:35,DP:1:2,TO:3000
*
* 3.5" flexible disc (9133H)
*
*
dvt,%dd*33:92077,M_FLOPPY_CF:1,LU:61,DP:1:7
*
* 5.25" Winchester fixed disc (9133H/9134H)
*
*
dvt,%dd*33:92077,M9133_CF:0,LU:41,DP:1:7
dvt,%dd*33:92077,M9133_CF:1,LU:42,DP:1:7
dvt,%dd*33:92077,M9133_CF:2,LU:43,DP:1:7,DP:5:57984
*
* 2608S,2563A,2566A Line Printer
*
*
dvt,%ddc12:92077, ,LU:85,DP:1:3
*
* 2932A Line Printer
*
*
dvt,%dd*12:92077,M2932A,LU:6,DT:12B,DP:1:6
```

HP-IB address 6
LU 32-33

select code = 26b

HP-IB address 36b

HP-IB address 0

HP-IB address 1
LU 12

HP-IB address 2
LU 34-35

HP-IB address 7
LU 61

HP-IB address 7
LU 41-43

HP-IB address 3
LU 85

HP-IB address 6
LU 6

```

*
* 248x INTEGRATED DISC INTERFACE (MICRO-1000)
*
ift,%id*27:92077,SC:32B
*
* Hard disc (15mb)
*
dvt,%gen27:92077,M2480:15,LU:55
dvt,%gen27:92077,M2480:16,LU:62
*
* Hard disc (20mb)
*
dvt,%gen27:92077,m2480:11,lu:59
dvt,%gen27:92077,m2480:12,lu:60
*
* 3.5" flexible disc (singled sided)
*
dvt,%gen27:92077,M2480:3,LU:39
*
* 3.5" flexible disc (double sided)
*
dvt,%gen27:92077,M2480:14,LU:63
*
*
*
* D.S. Links -- two LU's for D.S., and two for LU mapping          LU 92-95
*
* Network Links                                                    select code = 24b
*                                                                    LU 79-80
*
ift%id*66:91750,EID.66,SC:24B,QU:FI
*
dvt,,,LU:79,DT:66B
dvt,,,LU:80,DT:66B
*
*
* LU mapping                                                       LU 81-82
*
ift,%adv00:91750,EIDV00,QU:FI, TX:2,AL:DY
*
dvt,,,LU:81,EddV00, TX:0
dvt,,,LU:82,EddV00, TX:5
*
*
*
* Terminal LU's
*
* NOTE: There is no LU defined to be LU 1. This points to the bit bucket
*
* ASIC #1                                                           Select Code 20B
*
ift,id100.rel:92077,sc:20B
dvt,ddc00.rel:92077,MHP_Term:A,lu:100
*

```

```

*
* ASIC #2 Terminal LU:101                               Select Code 21B
*
ift,id100.rel:92077,sc:21B
dvt,ddc00.rel:92077,MHP_Term:A,lu:101
*
*
*
* 12100A          port A terminal LU:110                Select Code 77B
* A400/OBIO      port B terminal LU:111
* 4-MUX          port C terminal LU:112
*
* port D terminal LU:113
*
*
ift,id400.rel:92077
dvt,ddc00.rel:92077,MHP_Term:0,lu:110
dvt,ddc00.rel:92077,MHP_Term:1,lu:111
dvt,ddc00.rel:92077,MHP_Term:2,lu:112
dvt,ddc00.rel:92077,MHP_Term:3,lu:113
*
*
*
*
* 12040D          port 0 terminal LU:120                Select Code 30B
* Revision D     port 1 terminal LU:121
* 8-MUX          port 2 terminal LU:122
*
* port 3 terminal LU:123
*
* port 4 terminal LU:124
*
* port 5 terminal LU:125
*
* port 6 terminal LU:126
*
* port 7 terminal LU:127
*
*
*
ift,id800.rel:92077,sc:30B
dvt,ddc00.rel:92077,MHP_Term:0,lu:120
dvt,ddc00.rel:92077,MHP_Term:1,lu:121
dvt,ddc00.rel:92077,MHP_Term:2,lu:122
dvt,ddc00.rel:92077,MHP_Term:3,lu:123
dvt,ddc00.rel:92077,MHP_Term:4,lu:124
dvt,ddc00.rel:92077,MHP_Term:5,lu:125
dvt,ddc00.rel:92077,MHP_Term:6,lu:126
dvt,ddc00.rel:92077,MHP_Term:7,lu:127
*
*
*
*
* 12040A-C       port 0 terminal LU:130                Select Code 23B
* Pre-          port 1 terminal LU:131
* Revision D     port 2 terminal LU:132
* 8-MUX          port 3 terminal LU:133
*
* port 4 terminal LU:134
*
* port 5 terminal LU:135
*
* port 6 terminal LU:136
*
* port 7 terminal LU:137
*
*

```

```

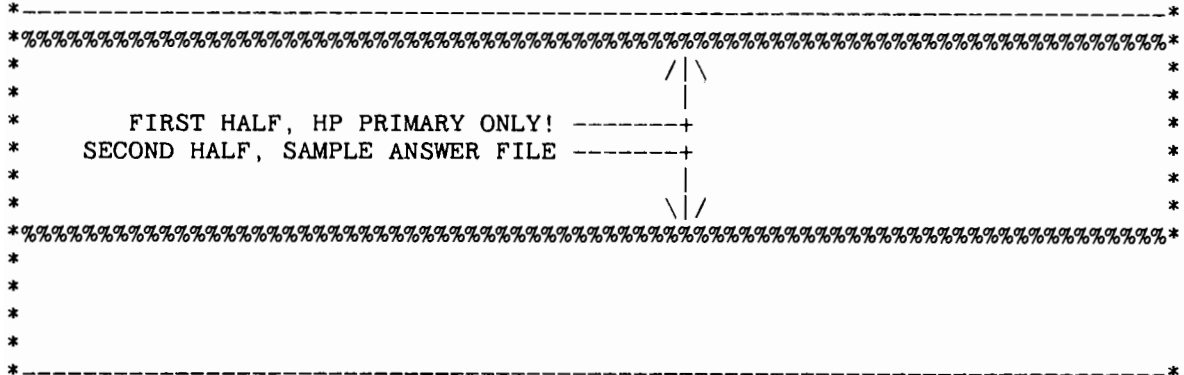
*
*
*
*
ift,%idm00:92077,sc:23B,tx:20
dvt,%dd*00:92077,M26XX,LU:130,DP:1:20004B,TX:57,-
    DP:5:CI:20040B:20040B:0,DP:9:CM:20040B:20040B:CM
dvt,%dd*00:92077,M26XX,LU:131,DP:1:20004B,TX:57,-
    DP:5:CI:20040B:20040B:0,DP:9:CM:20040B:20040B:CM
dvt,%dd*00:92077,M26XX,LU:132,DP:1:20004B,TX:57,-
    DP:5:CI:20040B:20040B:0,DP:9:CM:20040B:20040B:CM
dvt,%dd*00:92077,M26XX,LU:133,DP:1:20004B,TX:57,-
    DP:5:CI:20040B:20040B:0,DP:9:CM:20040B:20040B:CM
dvt,%dd*00:92077,M26XX,LU:134,DP:1:20004B,TX:57,-
    DP:5:CI:20040B:20040B:0,DP:9:CM:20040B:20040B:CM
dvt,%dd*00:92077,M26XX,LU:135,DP:1:20004B,TX:57,-
    DP:5:CI:20040B:20040B:0,DP:9:CM:20040B:20040B:CM
dvt,%dd*00:92077,M26XX,LU:136,DP:1:20004B,TX:57,-
    DP:5:CI:20040B:20040B:0,DP:9:CM:20040B:20040B:CM
dvt,%dd*00:92077,M26XX,LU:137,DP:1:20004B,TX:57,-
    DP:5:CI:20040B:20040B:0,DP:9:CM:20040B:20040B:CM
dvt,%dd*00:92077,M26XX,LU:138,DP:1:20004B,TX:57,-
    DP:5:CI:20040B:20040B:0,DP:9:CM:20040B:20040B:CM
end,,                                END OF dvt GENERATION PHASE
*
end,,                                END OF ift GENERATION PHASE
*
*
* Define Node Lists
*
* 7907/08/11/12/14/33/41/45, Cartridge Tape Drive, flexible 3.5" disc
node,16,17,18,19,20,21,22,23,24,25,26,29,30,31
*
* 3.5" 9153/54
node,50,51,52,53,54
*
* 3.5" Flexible Discs (9121)
node,32,33
node,34,35
*
* 5.25" Winchester fixed Disc LU's (9133/4 XV)
node,44,45,46,47
*
* 5.25" Fixed Disc LU's (9133/4 H) and 3.5" Floppy disc
node,41,42,43,61
*
* 248x Integrated Disc LU's (MICRO-1000)
node,55,62,59,60,39,63
*
*
end,                                END NODE LIST
*
*
end,                                END INTERRUPT TABLE
*
*
* Memory Allocation
*
*

```

```

clas 150,          Class number allocation
  resn 30,10      Resource number allocation/debug table spec.
id 150,           ID segment allocation
rs -150,         Memory descriptor allocation
sam 32767,4096   SAM allocation/XSAM
sl 200 1048,     Spool limits
bg 30,           Background swap priority allocation
qu 300 50,       Quantum time slice value
sp 30,10         Shared programs/extended sched table
*
mb 512,          Number of memory blocks for DS/1000
*
us 1,            Number of concurrent users (1 if no VC+)
lb,,             LOGOF buffer limit (use defaults)
*
*
* Labeled System Common Relocation
*
* Note: these DS1000 system common points are for DS testing
re %resa:91750
re $bigds:91750 #nrvs
re $bigds:91750 #levl
*
* Add the system common table for HpMdm.run
*
re hpmdm_table.rel:92077
*
ms $bigds:91750
*
end ,            END LABELED SYSTEM COMMON RELOCATION
*
*
* Unlabeled (Blank) Common
*
com 1024,        Number of words of memory to use
*
* System Messages
*
re %msgtb:92077,, Message table
end
re %$m000:92077,, Message module
end
*
end
*
* System Libraries
*
lib sec1000.lib
lib $biglb.lib
end
*
end
*
*
*

```



RTE-A Sample System Generation Answer File

* This half of the primary answer file is designed to be edited
 * and used for a system-specific generation. To use this file,
 * first delete the first half of this file, then uncomment the
 * required lines (marked with a "*!"), and any specific system
 * modules required by removing the appropriate comment character.

* For further information, consult the RTE-A System Generation
 * and Installation Manual (part no. 92077-90034).

- * 1. Uncomment all required modules by removing the string
 * ".*!REMOVE!". (See EDIT/1000 User's Guide and note
 * below for information on how to use EDIT/1000 to do this)
- * 2. If you want VC+ in your system, uncomment all VC+ modules
 * by removing the string ".*!YESVC+!"; otherwise uncomment
 * the non-VC+ modules by removing the string ".*!NO_VC+!".
 * VC+ is supplied by the HP 92078A Virtual Code+ package,
 * a package which allows a multiuser environment and spooling.
- * 3. If you have DS/1000 in your system, uncomment all DS/1000
 * modules by removing the string ".*!YES_DS!"; otherwise
 * uncomment the non-DS modules by removing the string
 * ".*!NO__DS!".
- * 4. If you have LAN/1000 in your system, uncomment all LAN/1000
 * modules by removing the string ".*!YESLAN!"; otherwise
 * do nothing. Note that the NS subsystems have their own
 * generation answer file, which you might prefer to use.
- * 5. If you do not use an ASIC card as LU 1 (as genned in
 * this system, comment it out and change one of the
 * terminal LU's generated into the system (such as the
 * 8 channel multiplexer) as LU 1.
- * 6. Uncomment the ".*" before the RPL file you will be using
 * according to your hardware configuration. If you are using
 * the A600 (not the A600+), i.e. %RPL60, you will need to
 * remove the string ".*!RPL60_!".
- * 7. If you want to use the DDC00/DDC01 modem handler, HpMdm.run,
 * uncomment the ".*" before hpmdm_table.rel. If you want to


```

*!REMOVE!re /rte_a/%exec,, EXEC request processing
*!REMOVE!re /rte_a/%rtioa,, Real-Time I/O control
*!REMOVE!re /rte_a/%iomod,, I/O module
*!REMOVE!re /rte_a/%maps,, Dynamic mapping system routines
*!REMOVE!re /rte_a/%progs,, Program state processing
*!REMOVE!re /rte_a/%util,, System variables and utilities
*!REMOVE!re /rte_a/%sam,, System available memory
*!REMOVE!re /rte_a/%sched,, Programmatic program scheduling
*!REMOVE!re /rte_a/%strng,, Runstring passing
*!REMOVE!re /rte_a/%erlog,, Error Logging
*!REMOVE!re /rte_a/%opmsg,, Operating system messages
*!REMOVE!re /rte_a/%sycom,, Operator commands
*!REMOVE!re /rte_a/%id*43,, Power fail driver
*!REMOVE!re /rte_a/%signl,, Signal processing module
*!YESVC+!re /vcplus/secos.rel,, Security/1000 module
*!YESVC+!re /vcplus/check.rel,, Security/1000 module
*!YESVC+!re /vcplus/%spool,, Spooling module
*
*
* #####
* #
* # If you are using an A600 ( not an A600+ ) or an A400 #
* # WITHOUT CDS, then the following paragraph applies. #
* # If you are using any other type of A-Series processor, #
* # ( A400 with CDS, A600+, A700 or A900 ) then the following #
* # following paragraph does NOT apply. #
* #
* # The two routines .dmp .ddi have to be relocated here #
* # because they are required by code that is in the O.S. #
* # partitions. You cannot do library searches for code in #
* # O.S. partitions. This is required ONLY when using an #
* # A600 or A400 without CDS (using %RPL40/%RPL60). If you #
* # are using any other RPL, then these modules will cause #
* # duplicate entry point errors if included in the generation. #
* #
* #####
*
*!RPL60!re /libraries/$math.lib,.dmp Required for A600 (not A600+) (RPL60)
*!RPL60!re /libraries/$math.lib,.ddi Required for A600 (not A600+) (RPL60)
*
*!REMOVE!se /libraries/$syslb.lib,,Search the system library
*
* define partitionable modules
*
*!REMOVE!pa perr,xcmd,stat,dsq,vema,lock,load,memry,iorq
*!REMOVE!pa time,class,abort,alarm
*!YESVC+!pa cdsfh
*
*
*
*!REMOVE!ms /libraries/$sysa.lib,, Search the system dummy library
*!REMOVE!end, Terminate system relocation phase
*
*
* OS partitions
*
*
*!REMOVE!re /rte_a/%load,, Program loading and swapping
*!REMOVE!end

```

```

*
*!REMOVE!re /rte_a/%memry,,      Memory management module
*!REMOVE!end
*
*!REMOVE!re /rte_a/%iorq,,      I/O request processing
*!REMOVE!end
*
*!REMOVE!re /rte_a/%lock,,      LU locking and resource numbers
*!REMOVE!end
*
*!REMOVE!re /rte_a/%time,,      Time scheduling of programs
*!REMOVE!re /rte_a/%class,,    Class I/O module
*!REMOVE!end
*
*!REMOVE!re /rte_a/%xcmd,,      Operator command extension module
*!REMOVE!re /rte_a/%stat,,     Status command module
*!REMOVE!re /rte_a/%dsq,,      DS/1000 module
*!REMOVE!end
*
*!REMOVE!re /vcplus/%cdfs,,    For CDS systems only
*!REMOVE!al,                    Align next module
*!REMOVE!re /rte_a/%perr,,     Parity error handler
*!REMOVE!re /rte_a/%vema,,     Virtual memory module
*!REMOVE!end
*
*!REMOVE!re /rte_a/%alarm,,    Alarm signal processing module
*!REMOVE!end
*
*!REMOVE!re /rte_a/%abort,,    Abort processing module
*!REMOVE!end
*
*
* Driver partitions
*
*
*!REMOVE!re /rte_a/%dd*33,,     Device driver for CS/80 discs
*!REMOVE!end
*
*!YES_DS!re /ds1000/%id*66,,    DS/1000: interface driver for DS/1000
*!YES_DS!end
*
*!YESLAN!re /rte_a/%id*67,,    LAN: interface driver for IEEE802.
*!YESLAN!end
*
*!REMOVE!re /rte_a/%id*37,,    Interface driver for CS/80 discs
*!REMOVE!end
*
*!REMOVE!re /rte_a/id400.rel,,  Interface driver for A400/OBIO
*!REMOVE!end
*
*!REMOVE!re /rte_a/%idm00,,    Interface driver for 12040 rev A-C
*!REMOVE!re /rte_a/%dd*23,,    Device driver for magnetic tape de
*!REMOVE!end
*
*!REMOVE!re /rte_a/%dd*00,,    Device driver for %idm00
*!REMOVE!re /rte_a/%dd*30,,    Device driver for discs (ICD)
*!REMOVE!end
*
*!YES_DS!re /ds1000/%adv00,,   DS/1000:device driver for DS/1000

```

```

*!YES_DS!end
*
*!REMOVE!re /rte_a/id800.rel,, Interface driver for 12040D 8 channel MUX
*!REMOVE!re /rte_a/%dd*12,, Device driver for HP-IB line printer
*!REMOVE!end
*
*
* NOTE: if you desire to use slaved devices (i.e., printers) or CTU's
* connected to your terminal, use DDC01.REL instead of DDC00.REL.
* Refer to the RTE-A generation manual for more details.
*
*!REMOVE!re /rte_a/ddc01.rel,, Device driver for 100 series interfaces
*!REMOVE!end
*
*!REMOVE!re /rte_a/%ddc12,, Device driver for HP-IB line printer
*!REMOVE!end
*
*!REMOVE!re /rte_a/%id*27,, Interface driver for 248X integrated disc
*!REMOVE!re /rte_a/%dd*24,, Device driver for 7974A/7978A streaming tape
*!REMOVE!end
*
*!REMOVE!re /rte_a/id100.rel,, Interface driver for 12005 ASIC
*!REMOVE!re /rte_a/%id*50,, Interface driver for Parallel card
*!REMOVE!end
*
*!REMOVE!end,, END DRIVER PARTITION PHASE
*
*
*
* Table Generation -- configure LU tables
*
*
*
* HP-IB #1 -- discs and magnetic tape select code = 27b
*
*
*!REMOVE!ift,/rte_a/%Id*37,SC:27B
*
* Bus Controller LU HP-IB address 36b
*
*!REMOVE!dvt,, ,LU:15,TO:2000,DT:77B,TX:0,DX:1,DP:1:36B,PR:0
*
* 7907/08/11/12/14/33/35/41/42/45/46 CS/80 discs
*
* HP-IB address 0
* LU 16-23,25-26,
* 29-31
*!REMOVE!dvt,/rte_a/%dd*33,M7908_CF:0,LU:16,DP:1:0
*!REMOVE!dvt,/rte_a/%dd*33,M7908_CF:1,LU:17,DP:1:0
*!REMOVE!dvt,/rte_a/%dd*33,M7907FXD_CF:2,LU:18,DP:1:0
*!REMOVE!dvt,/rte_a/%dd*33,M7941_CF:3,LU:19,DP:1:0
*!REMOVE!dvt,/rte_a/%dd*33,M7911_CF:4,LU:20,DP:1:0
*!REMOVE!dvt,/rte_a/%dd*33,M7945_CF:5,LU:21,DP:1:0
*!REMOVE!dvt,/rte_a/%dd*33,M7912_CF:6,LU:22,DP:1:0
*!REMOVE!dvt,/rte_a/%dd*33,M7914_CF:7,LU:23,DP:1:0
*!REMOVE!dvt,/rte_a/%dd*33,M7933_CF:8,LU:25,DP:1:0
*!REMOVE!dvt,/rte_a/%dd*33,M7933_CF:9,LU:26,DP:1:0
*!REMOVE!dvt,/rte_a/%dd*33,M7907REM_CF:0,LU:29,DP:1:0
*!REMOVE!dvt,/rte_a/%dd*33,M7907REM_CF:1,LU:30,DP:1:0

```

```

*!REMOVE!dvt,/rte_a/%dd*33,M7907REM_CF:2,LU:31,DP:1:0
*
*
* CS/80 compatible cartridge tape cache                HP-IB address 0
*                                                       LU 24
*
*!REMOVE!dvt,/rte_a/%dd*33,MTAPE,LU:24,DP:1:0
*
*
* 9144 stand-alone CS/80 cartridge tape                HP-IB address 1
*                                                       LU 9
*
*!REMOVE!dvt,/rte_a/%dd*33,M9144:0,LU:9,DP:1:1
*
*
* 3.5" single sided (9153/54)                          HP-IB address 2
*                                                       LU 53
*!REMOVE!dvt,/rte_a/%dd*33,M_FLOPPY_CF:1,LU:53,DP:1:2:400b:0:0:0,-
*!REMOVE!DP:6:66:16:0
*
*
* 3.5" double sided (9153/54)                          HP-IB address 2
*                                                       LU 54
*!REMOVE!dvt,/rte_a/%dd*33,M_FLOPPY_CF:1,LU:54,DP:1:2
*
* 3.5" fixed disc (9153/54)                            HP-IB address 2
*                                                       LU 50-52
*!REMOVE!dvt,/rte_a/%dd*33,M9133_CF:0,LU:50,DP:1:2
*!REMOVE!dvt,/rte_a/%dd*33,M9133_CF:1,LU:51,DP:1:2
*!REMOVE!dvt,/rte_a/%dd*33,M9133_CF:2,LU:52,DP:1:2,DP:5:57984
*
* 7974A/7978A streaming tape drive                    HP-IB address 3
*                                                       LU 7
*
*!REMOVE!dvt,/rte_a/%dd*24,M7974:0,LU:7,DP:1:3
*
*
* HP-IB tape drive                                     HP-IB address 4
*                                                       LU 8
*
*!REMOVE!dvt,/rte_a/%dd*23,M7970E:0,LU:8,DP:1:4,PR:1
*
*
* floppy disc 9122d                                    HP-IB address 5
*                                                       LU 36-37
*
*!REMOVE!dvt,/rte_a/%dd*33,m_floppy_cf:0,lu:36
*!REMOVE!dvt,/rte_a/%dd*33,m_floppy_cf:1,lu:37
*
*
* 5.25" and 3.5" flexible discs (9121)                HP-IB address 6
*                                                       LU 32-33
*
*!REMOVE!dvt,/rte_a/%dd*30,M9121:0,LU:32,DP:1:6,TO:3000
*!REMOVE!dvt,/rte_a/%dd*30,M9121:1,LU:33,DP:1:6,TO:3000
*
*
*
*

```

```

* HP-IB #2 -- discs and magnetic tape                select code = 26b
*
*
*!REMOVE!ift,/rte_a/%Id*37,SC:26B
*
* Bus Controller LU                                HP-IB address 36b
*
*!REMOVE!dvt,, ,LU:40,TO:2000,DT:77B,TX:0,DX:1,DP:1:36B,PR:0
*
* 5.25" Winchester fixed disc (9133/4 XV)          HP-IB address 0
*                                                    LU 44-47
*
*!REMOVE!dvt,/rte_a/%dd*30,M9134X:0,LU:44,DP:1:0
*!REMOVE!dvt,/rte_a/%dd*30,M9134X:1,LU:45,DP:1:0
*!REMOVE!dvt,/rte_a/%dd*30,M9134X:2,LU:46,DP:1:0
*!REMOVE!dvt,/rte_a/%dd*30,M9134X:3,LU:47,DP:1:0
*
* 9144 stand-alone CS/80 cartridge tape            HP-IB address 1
*                                                    LU 12
*!REMOVE!dvt,/rte_a/%dd*33,M9144:0,LU:12,DP:1:1
*
* 5.25" and 3.5" flexible discs (9121)             HP-IB address 2
*                                                    LU 34-35
*!REMOVE!dvt,/rte_a/%dd*30,M9121:0,LU:34,DP:1:2,TO:3000
*!REMOVE!dvt,/rte_a/%dd*30,M9121:1,LU:35,DP:1:2,TO:3000
*
* 3.5" flexible disc (9133H)                      HP-IB address 7
*                                                    LU 61
*!REMOVE!dvt,/rte_a/%dd*33,M_FLOPPY_CF:1,LU:61,DP:1:7
*
* 5.25" Winchester fixed disc (9133H/9134H)       HP-IB address 7
*                                                    LU 41-43
*!REMOVE!dvt,/rte_a/%dd*33,M9133_CF:0,LU:41,DP:1:7
*!REMOVE!dvt,/rte_a/%dd*33,M9133_CF:1,LU:42,DP:1:7
*!REMOVE!dvt,/rte_a/%dd*33,M9133_CF:2,LU:43,DP:1:7,DP:5:57984
*
* 2608S,2563A,2566A Line Printer                  HP-IB address 3
*                                                    LU 85
*!REMOVE!dvt,/rte_a/%ddC12,,LU:85,DP:1:3
*
* 2932A Line Printer                              HP-IB address 6
*                                                    LU 6
*!REMOVE!dvt,/rte_a/%dd*12,M2932A,LU:6,DT:12B,DP:1:6
*
* 248x INTEGRATED DISC INTERFACE (MICRO-1000)
*
*!REMOVE!ift,/rte_a/%Id*27,SC:32B
*
* Hard disc (15mb)
*
*!REMOVE!dvt,/rte_a/%GEN27,M2480:15,LU:55
*!REMOVE!dvt,/rte_a/%GEN27,M2480:16,LU:62
*
* Hard disc (20mb)
*
*!REMOVE!dvt,/rte_a/%gen27,m2480:11,lu:59
*!REMOVE!dvt,/rte_a/%gen27,m2480:12,lu:60
*
* 3.5" flexible disc ( singled sided )

```

```

*
*!REMOVE!dvt,/rte_a/%GEN27,M2480:3,LU:39
*
* 3.5" flexible disc ( double sided )
*
*!REMOVE!dvt,/rte_a/%GEN27,M2480:14,LU:63
*
*
*
* HP-IB #3 -- Instruments                select code = 25b
*
*!REMOVE!ift,/rte_a/%Id*37,SC:25B
*
* Bus Controller LU                      HP-IB address 36b
*                                         LU 91
*!REMOVE!dvt,,,LU:91,TO:2000,DT:77B,DX:1,DP:1:36B
*
* FOUR DEVICES                          HP-IB address 1,5-7
*                                         LU 92-95
*!REMOVE!dvt,,,LU:92,TO:500,DT:77B,DX:1,DP:1:1
*!REMOVE!dvt,,,LU:93,TO:500,DT:77B,DX:1,DP:1:5
*!REMOVE!dvt,,,LU:94,TO:500,DT:77B,DX:1,DP:1:6
*!REMOVE!dvt,,,LU:95,TO:500,DT:77B,DX:1,DP:1:7
*
*
*
* D.S. Links -- two LU's for D.S., and two for LU mapping
*
*
* Network Links                          select code = 24b
*                                         LU 79-80
*!YES_DS!ift,/ds1000/%ID*66,EID.66,SC:24B,QU:FI
*
*!YES_DS!dvt,,,LU:79,DT:66B
*!YES_DS!dvt,,,LU:80,DT:66B
*
*
* LU mapping                              LU 81-82
*!YES_DS!ift,/ds1000/%ADV00,EIDV00,QU:FI,TX:2,AL:DY
*
*!YES_DS!dvt,,,LU:81,EddV00,TX:0
*!YES_DS!dvt,,,LU:82,EddV00,TX:5
*
* IEEE 802.3 Local Area Network          LU 96,97
*
* Table entries for id*67 the LAN/1000 driver for 12076A
* The ift extension area defaults to 200 words, allowing
* 4 Multicast addresses to be used by the card. The formula
* to change this is: tx=188+(3* max # of multicast addresses
* used by the card.)
*!YESLAN!ift,/rte_a/%id*67,tx:200,sc:37
*
* dvt table entries for id*67 (12076A 802.3 LAN card)
* first dvt entry MUST have an even LU

```



```

*
*!YESLAN!dvt,/rte_a/%gen67,m67:1,lu:96
*!YESLAN!dvt,/rte_a/%gen67,m67:0,lu:97
*
*
*
* Parallel Interface Card                               select code = 35b
*                                                         LU 84
*
*!REMOVE!ift,/rte_a/%Id*50,SC:35B
*
*!REMOVE!dvt,,LU:84,TO:5000,DX:2,DP:1:0:0,DT:55B
*
*
* Terminal LU's
*
* #####
* #
* # For systems with non-12005 ASIC system consoles: #
* #   change the 'lu:1' below to 'lu:yy', where yy is an #
* #   change the 'lu:1' to 'lu:yy', where yy is an unused LU, #
* #   is an unused LU, and change the desired LU:XX to LU:1. #
* #
* #####
*
* ASIC #1 (default system console)                       Select Code 20B
*
*!REMOVE!ift,/rte_a/id100.rel,sc:20B
*!REMOVE!dvt,/rte_a/ddc01.rel,MHP_Term:A,lu:1
*
*
*
* ASIC #2 Terminal LU:100                               Select Code 21B
*   slaved device (printer) LU:200
*   left CTU                                             LU:201
*   right CTU                                           LU:202
*
*!REMOVE!ift,/rte_a/id100.rel,sc:21B
*!REMOVE!dvt,/rte_a/ddc01.rel,MHP_Term:A,lu:100
*!REMOVE!dvt,/rte_a/ddc01.rel,MHP_Slaved_Serial,lu:200
*!REMOVE!dvt,/rte_a/ddc01.rel,MHP_CTU:L,lu:201
*!REMOVE!dvt,/rte_a/ddc01.rel,MHP_CTU:R,lu:202
*
*
*
* 12100A        port A terminal LU:110                   Select Code 77B
* A400/OBIO     port B terminal LU:111
* 4-MUX         slaved device (printer) LU:211
*               left CTU                               LU:212
*               right CTU                             LU:213
*
*               port C terminal LU:112
*               port D terminal LU:113
*
*
*!NOTE: Do not specify a select code for id400.rel.  It is forced to
*   select code 77B.
*!REMOVE!ift,/rte_a/id400.rel
*!REMOVE!dvt,/rte_a/ddc01.rel,MHP_Term:O,lu:110

```

```

*!REMOVE!dvt,/rte_a/ddc01.rel,MHP_Term:1,lu:111
*!REMOVE!dvt,/rte_a/ddc01.rel,MHP_Slaved_Serial,lu:211
*!REMOVE!dvt,/rte_a/ddc01.rel,MHP_CTU:L,lu:212
*!REMOVE!dvt,/rte_a/ddc01.rel,MHP_CTU:R,lu:213
*!REMOVE!dvt,/rte_a/ddc01.rel,MHP_Term:2,lu:112
*!REMOVE!dvt,/rte_a/ddc01.rel,MHP_Term:3,lu:113
*
*
*
*   12040D      port 0 terminal LU:120      Select Code 30B
*   Revision D  port 1 terminal LU:121
*   8-MUX       slaved device (printer) LU:221
*               left CTU                  LU:222
*               right CTU                 LU:223
*
*   port 2 terminal LU:122
*   port 3 terminal LU:123
*   port 4 terminal LU:124
*   port 5 terminal LU:125
*   port 6 terminal LU:126
*   port 7 terminal LU:127
*
*!REMOVE!ift,/rte_a/id800.rel,sc:30B
*!REMOVE!dvt,/rte_a/ddc01.rel,MHP_Term:0,lu:120
*!REMOVE!dvt,/rte_a/ddc01.rel,MHP_Term:1,lu:121
*!REMOVE!dvt,/rte_a/ddc01.rel,MHP_Slaved_Serial,lu:221
*!REMOVE!dvt,/rte_a/ddc01.rel,MHP_CTU:L,lu:222
*!REMOVE!dvt,/rte_a/ddc01.rel,MHP_CTU:R,lu:223
*!REMOVE!dvt,/rte_a/ddc01.rel,MHP_Term:2,lu:122
*!REMOVE!dvt,/rte_a/ddc01.rel,MHP_Term:3,lu:123
*!REMOVE!dvt,/rte_a/ddc01.rel,MHP_Term:4,lu:124
*!REMOVE!dvt,/rte_a/ddc01.rel,MHP_Term:5,lu:125
*!REMOVE!dvt,/rte_a/ddc01.rel,MHP_Term:6,lu:126
*!REMOVE!dvt,/rte_a/ddc01.rel,MHP_Term:7,lu:127
*
*
*
*   12040A-C    port 0 terminal LU:130      Select Code 23B
*   Pre-        port 1 terminal LU:131
*   Revision D  port 2 terminal LU:132
*   8-MUX       port 3 terminal LU:133
*
*   port 4 terminal LU:134
*   port 5 terminal LU:135
*   port 6 terminal LU:136
*   port 7 terminal LU:137
*
*!REMOVE!ift,/rte_a/%idm00,sc:23B,tx:20
*!REMOVE!dvt,/rte_a/%dd*00,M26XX,LU:130,DP:1:20004B,TX:57,-
*!REMOVE!      DP:5:CI:20040B:20040B:0,DP:9:CM:20040B:20040B:CM
*!REMOVE!dvt,/rte_a/%dd*00,M26XX,LU:131,DP:1:20004B,TX:57,-
*!REMOVE!      DP:5:CI:20040B:20040B:0,DP:9:CM:20040B:20040B:CM
*!REMOVE!dvt,/rte_a/%dd*00,M26XX,LU:132,DP:1:20004B,TX:57,-
*!REMOVE!      DP:5:CI:20040B:20040B:0,DP:9:CM:20040B:20040B:CM
*!REMOVE!dvt,/rte_a/%dd*00,M26XX,LU:133,DP:1:20004B,TX:57,-
*!REMOVE!      DP:5:CI:20040B:20040B:0,DP:9:CM:20040B:20040B:CM
*!REMOVE!dvt,/rte_a/%dd*00,M26XX,LU:134,DP:1:20004B,TX:57,-
*!REMOVE!      DP:5:CI:20040B:20040B:0,DP:9:CM:20040B:20040B:CM
*!REMOVE!dvt,/rte_a/%dd*00,M26XX,LU:135,DP:1:20004B,TX:57,-
*!REMOVE!      DP:5:CI:20040B:20040B:0,DP:9:CM:20040B:20040B:CM
*!REMOVE!dvt,/rte_a/%dd*00,M26XX,LU:136,DP:1:20004B,TX:57,-

```

```

*!REMOVE!      DP:5:CI:20040B:20040B:0,DP:9:CM:20040B:20040B:CM
*!REMOVE!dvt,/rte_a/%dd*00,M26XX,LU:137,DP:1:20004B,TX:57,-
*!REMOVE!      DP:5:CI:20040B:20040B:0,DP:9:CM:20040B:20040B:CM
*!REMOVE!dvt,/rte_a/%dd*00,M26XX,LU:138,DP:1:20004B,TX:57,-
*!REMOVE!      DP:5:CI:20040B:20040B:0,DP:9:CM:20040B:20040B:CM
*!REMOVE!end,,          END OF dvt GENERATION PHASE
*
*!REMOVE!end,,          END OF ift GENERATION PHASE
*
*
* Define Node Lists
*
* 7907/08/11/12/14/33/41/45 and a Compatible Cartridge Tape Drive
*!REMOVE!node,16,17,18,19,20,21,22,23,24,25,26,29,30,31
*
* 5.25" and 3.5" Flexible Discs (9121)
*!REMOVE!node,32,33
*!REMOVE!node,34,35
*
* 9122D 3.5" flexible discs
*!REMOVE!node,36,37
*
* 3.5" flexible disc and fixed disc (9153/54)
*!REMOVE!node,50,51,52,53,54
*
* 5.25" Winchester fixed Disc LU's (9133/4 XV)
*!REMOVE!node,44,45,46,47
*
* 5.25" Winchester Fixed Disc LU's (9133/4 H) and 9133H flexible disc
*!REMOVE!node,41,42,43,61
*
* 248x Integrated Disc LU's
*!REMOVE!node,55,62,59,60,39,63
*
* 4-Channel Mux port B/Rev. D 8-MUX CTU's and slaved printer
*!REMOVE!node,111,211,212,213
*!REMOVE!node,121,221,222,223
*
*!REMOVE!end,,          END NODE LIST
*
*
*!REMOVE!end,,          END INTERRUPT TABLE
*
*
* Memory Allocation
*
*
*!REMOVE!clas 150,      Class number allocation
*!REMOVE!resn 30,0     Resource number allocation/debug table spec
*!REMOVE!id 50,       ID segment allocation
*!REMOVE!rs -150,     Memory descriptor allocation
*!REMOVE!sam 32767,4096 SAM allocation/XSAM
*!REMOVE!sl 200 1048, Spool limits
*!REMOVE!bg 30,       Background swap priority allocation
*!REMOVE!qu 300 50,   Quantum time slice value
*!REMOVE!sp 30,0     Shared programs/extended sched table
*
*!YES_DS!mb 512,      Number of memory blocks for DS/1000
*!NO_DS!mb 0,        Number of memory blocks for DS/1000

```

```

*
*!YESVC+!us 15,           Number of concurrent users (1 if no VC+)
*!NO_VC+!us 1,           Number of concurrent users (1 if no VC+)
*!REMOVE!lb,,           LOGOF buffer limit (use defaults)
*
*
* Labeled System Common Relocation
*
*
*!YES_DS!re /ds1000/%resa
*!YES_DS!re /libraries/$bigds.lib #nrvs
*!YES_DS!re /libraries/$bigds.lib #levl
*
* If you are using the serial modem handler HpMdm, uncomment the
* following line
*re /rte_a/hpmdm_table.rel
*
*!YES_DS!ms /libraries/$bigds.lib
*
*!REMOVE!end,           END LABELED SYSTEM COMMON RELOCATION
*
*
* Unlabeled (Blank) Common
*
*!REMOVE!com 1024,       Number of words of memory to use
*
*
* System Messages
*
*!REMOVE!re /rte_a/%msgtb,, Message table
*!REMOVE!end
*!REMOVE!re /rte_a/%$m000,, Message module
*!REMOVE!end
*!YESVC+!re /vcplus/security.rel
*!YESVC+!end
*
*!REMOVE!end
*
* System Libraries
*
*!REMOVE!lib sec1000.lib,, Security/1000 library
*!YES_DS!lib $bigds.lib,, DS/1000 library
*!REMOVE!lib $biglb.lib,,
*
* add here any other library routines which you often use
*
*!REMOVE!end
*
* CDS Libraries
*
*!YESVC+!lib sec1000cds.lib,, Security/1000 CDS library
*!YESVC+!lib $bgcds.lib,,
*!YESVC+!lib sec1000.lib,, Security/1000 library
*!YESVC+!lib $biglb.lib,,
*
*!REMOVE!end

```





VCP Boot Information

VCP Errors

If you input a command the VCP does not recognize, the VCP terminal will respond with:

```
!?  
VCP>
```

Other errors during initial loading are contained in Appendix A of this manual.

General Boot Procedure

From the virtual control panel (VCP) terminal:

1. Press the BREAK key to pass control to the VCP program if not already in VCP mode. VCP commands will now be used to boot the system.
2. Enter `%Bdevice[ffbusc][file][,SS]`
or
`%Ldevice[ffbusc]`

where:

`%B` Execute a ROM loader routine specified by the two character device code and start execution as soon as the system is loaded.

`%L` Execute a ROM loader routine specified by the two character device code, but do not start execution of the system after it has been loaded into memory. Mainly used for loading diagnostics and memory patches.

`device` Specifies the device from which to boot the system as defined by one of the following two-character mnemonic codes:

RM PROM card
DC Disc drive or Cartridge Tape Drive (CTD) via HP-IB
DS Disc drive via DS/1000-IV link
MT Magnetic tape drive
DI Integrated discs

- ffbusc Defines the device parameters to be passed to the ROM loaders. Octal digits, leading zeros can be omitted.
- ff = File number. For moving head discs, this value is an offset of 256-block increments. The default is 0, which indicates that the head is to be positioned at the beginning of the disc (cylinder 0) when the boot extension file is loaded. If any other value is given, the disc head moves 256 blocks times this value before reading from the disc.
- If you are booting over DS, the file number can be no greater than 77777.
- b = Bus address (or disc address). The disc ROM loader uses this value to specify the HP-IB address (0-7) of the disc unit from which to load the system file. For the 248x integrated disc, the disc ROM loader uses this value to specify the integrated disc address from which to load the system file. The unit number must be 0 for both the fixed discs and the micro-floppy disc driver.
- u = Unit or head number from which to load the file (0 or 1). For the 7902 disc, this parameter is 0 for the left drive or 1 for the right. In the case of the 7906 disc, this parameter is the head number. In the case of the CS/80 disc drive with CTD, this parameter is 0 for the disc drive or 1 for the cartridge tape.
- sc = Select code of the card that is interfaced to the device containing the system file.
- file Name of file to be loaded into memory (memory based systems) or name of the boot command file to be used by the boot extension (disc based system). The file for a disc volume must be in directory SYSTEM. The default name for both of these files is BOOT.CMD::SYSTEM for a file disc volume or SYSTEM for a FMGR cartridge. Do not use this parameter when you are loading via DS.
- ,SS By appending SS to the end of the file name, the Boot extension will load the system and then display the VCP prompt (VCP>) instead of starting the system. A %R or %E will start the system. The SS parameter must be separated from the file name with a space or comma.

The VCP program will use default values, depending on where the system is to be booted from, if these optional parameters are not supplied. See Table H-1 for the default values.

Table H-1. VCP Default Parameters for Boot

device	ffbusc	file
Disc via HP-IB	2027	/SYSTEM/BOOT.COMD or SYSTEM
PROM Module	22	none
DS Link	24	none
Magnetic Tape	4027	none
248x Integrated Disc Controller	32	/SYSTEM/BOOT.COMD or SYSTEM

Parity Errors

Whenever the computer is running and a parity error occurs, the number of the page that contains the parity error is reported on the system console. It is a good idea to keep a list of bad pages, as you will want to specify them during boot.

Parity errors may be either hard (permanent) or soft (temporary). The next time you boot after a parity error has occurred, enter a %C to the VCP before you enter the %B command. The %C command writes zeros in all locations in memory. If your parity error was a soft parity error it will not recur; if it was a hard parity error it will recur and the error will be reported. You can then specify the page on which it occurred as a bad page in the boot command file.

The %C command may take a few seconds to execute, particularly if you have a large amount of memory.

Loading Your System from Disc

Before loading your new disc based system into memory from disc, make sure you do the following:

- Ensure that the boot extension (BOOTEX) is physically stored beginning at cylinder zero of the disc drive from which the system is to be loaded. Refer to the INSTL chapter in the Utilities Manual and Chapter 9 of this manual for details on installation of the boot extension (BOOTEX).
- Be sure the boot command file, the memory image system file, and the snap file are stored on the same LU that contains BOOTEX.
- Obtain the HP-IB address, unit number, head number, and the interface card select code of the disc (or the select code of the 248x integrated disc controller) from which your system will be loaded.

Now the following should be entered from the VCP:

1. Press the BREAK key to pass control to the VCP program if not already in VCP mode.
2. For disc on the HP-IB bus, enter:

`%BDCbusfile[,SS]`

Default is BOOT.CMD::SYSTEM (or system)

Default is 2027 octal

where:

- %BDC** Execute the disc bootstrap loader program for the disc device, including the CTD. The loader will load the boot extension (BOOTEX) into low memory. BOOTEX will set up the swap area, the memory allocation table, and relink and RP program files specified in the command file. BOOTEX will then copy part of itself to the top of physical memory and will load in the system file. When loading is complete the system starts execution immediately (unless the suspend 'SS' was added).
- b** HP-IB address of the disc drive where your target system file resides. Default address is 2.
- u** Unit number or head number of the disc drive (0-7) where the system file resides. Default value is 0.
- sc** Select code of the HP-IB card to which the disc is interfaced. The select code is switch-selectable on the card. The default select code is 27 octal.
- file** Name of the boot command file to be used by BOOTEX. The file for a disc volume must be in directory SYSTEM. As shown above, the default file name that BOOTEX will use is BOOT.CMD::SYSTEM (or SYSTEM for a FMGR disc cartridge). However, if you have named your boot command file anything other than the default file name, you will have to supply the name to the VCP program to be passed to the boot extension. This allows you to store multiple systems on one disc, and specify to the VCP program to be booted. Further, if you want to execute any of the diagnostic programs, you may pass the name of the diagnostic to be placed into memory by BOOTEX.
- ,SS** When SS is appended to the file name, a "cold" load will result. That is, the system will be loaded into memory by BOOTEX, and BOOTEX will halt. You can then alter the system with VCP commands, and start up by using the %E VCP command.

Note that SS must be separated from the file name with a space or a comma.

If you wish to load a memory image system file directly (the output file from BUILD for a memory based system, for example) the process is very similar to that described above. The file name is the name of the memory image system file instead of the boot control file. BOOTEX detects that the file is a system file instead of a boot control file and simply loads the system into memory.

Examples

The following examples show the entries that should be made at the VCP when booting several different systems with the system file at the location described in the example definition. Note that all user entries are italicized and that values displayed by the VCP program will be different.

1. To load and execute from a disc a system with a boot command file called `BOOT.CMD::SYSTEM` and with a hierarchical file system:

```
Bus Address = 2
Unit Number = 0
Select Code = 27 (hierarchical file disc volume)
```

The following sequence of commands should be entered at the VCP:

```
CI> (Strike the BREAK key)
P 002675 A 077774 B 000000 RW 100003 M 002674 T 026655
```

```
VCP> %BDC or lowercase %bdc
BOOT PROCESS COMPLETE
```

```
* RTE READY *
```

```
:
```

Because the default parameters are used, only `%BDC` needs to be entered to specify that the default system is to be booted from disc.

2. To load the diagnostic program called `HPIB::SYSTEM` from the disc with:

```
Bus Address = 7
Unit Number = 1
Select Code = 27
```

The following sequence of commands must be entered from the VCP:

```
CI> (Strike the BREAK key)
P 002675 A 077774 B 000000 RW 100003 M 002674 T 026655
```

```
VCP> %BDC7127HPIB or %bdc7127hpib
```

```
12009A HP-IB INTERFACE DIAGNOSTIC
```

```
READY
```

In the above example, all parameters must be entered and the diagnostic program specified, since none of them are the default values.

3. To cold load a system with a boot command file called MUTIGR::SYSTEM from the disc with:

Bus Address = 2
Unit Number = 0
Select Code = 27

The following sequence of commands should be entered from the VCP:

```
CI> (Strike the BREAK key)
P 002675 A 077774 B 000000 RW 100003 M 002674 T 026655
```

```
VCP> %bdcMUTIGR,ss
BOOT PROCESS COMPLETE
```

```
VCP> %E
```

```
* RTE READY *
:
:
```

The device parameters did not have to be supplied here since all are default values. The first character following BDC is non-numeric, indicating that MUTIGR is text and VCP should use the default parameters. The characters ,SS were appended to the file name to indicate a cold load. This allows you to make changes or load patches to the system prior to entering the %E.

4. To boot a system from disc with a boot command file called BOOT.CMD::SYSTEM, using a hierarchical file disc volume:

Bus Address = 7
Unit Number = 0
Select Code = 27

The following sequence of commands should be entered from the VCP:

```
CI> (Strike the BREAK key)
P 002675 A 077774 B 000000 RW 100003 M 002674 T 026655
```

```
VCP> %bdc7027
BOOT PROCESS COMPLETE
```

```
* RTE READY *
:
:
```

This example shows that the boot command file name does not have to be supplied since the default file name is used. This command also could be given as

```
VCP> bdc7
```

since the unit number and select code variables are the default values.

5. To boot a system with a boot command file called LSTIGR.CMD::SYSTEM from the 7906 disc with:

Bus Address = 2
Unit Number = 0
Head Number = 3
Select Code = 27

The following sequence of commands should be entered from the VCP:

```
FMGR: (Strike the BREAK key)  
P 002675 A 077774 B 000000 RW 100003 M 002674 T 026655
```

```
VCP> %bdc2327LSTIGR.CMD::SYSTEM  
BOOT PROCESS COMPLETE
```

```
* RTE READY *  
:  
:
```

If the 7906 disc is structured in surface mode, the head number where the system file resides may be supplied to the VCP program.

6. To boot a memory image system file output from BUILD called BOOT.CMD::SYSTEM from the disc with:

Bus Address = 7
Unit Number = 0
Select Code = 22 (hierarchical file disc volume)

The following sequence of commands should be entered from the VCP:

```
CI> (Strike the BREAK key)  
P 002675 A 077774 B 000000 RW 100003 M 002674 T 026655
```

```
VCP> %bdc7022  
BOOT PROCESS COMPLETE
```

```
* RTE READY *  
:  
:
```

This example shows that the system file name does not have to be supplied since the default file name is used.

7. To load and execute a system from disc, entering the boot commands interactively, with BOOTEX and the system and snap files described by

Bus Address = 2
Unit Number = 0
Select Code = 27

The following sequence of commands should be entered at the VCP:

```
CI> (Strike the BREAK key)
P 002675 A 077774 B 000000 RW 100003 M 002674 T 026655

VCP> %bdcLU1

BOOTEX: SY,SYSTMTC          system file on directory SYSTEM
BOOTEX: SN,SNAPC            snap file name
BOOTEX: MC,13              mount disc LU 13
BOOTEX: RP,D.RTR,D.RTR    RP D.RTR, keep the D.RTR name
BOOTEX: RP,CI              RP CI
BOOTEX: ST                 CI is the startup program
BOOTEX: EN                 terminate RP sequence
BOOTEX: RV,10              partition 10 pages long
BOOTEX: RV,12              partition 12 pages long
BOOTEX: RV,20              partition 20 pages long
BOOTEX: SW,/SYSTEM/SWAP:1024 create 1024-block swap file
BOOTEX: EN                 terminate command entry
BOOT PROCESS COMPLETE

* RTE READY *
:
:
```

This example shows that LU1, a type-zero file for the terminal, can be used as the file name and boot commands can be entered interactively.

If the boot program cannot find the file specified as the boot control file, it will switch to the interactive mode.

The default directory for command file SYSTEMFILE and the snap file is directory SYSTEM on a hierarchical file system disc volume. If the LU from which the system booted is a FMGR LU, the default is cartridge 0.

Program files in the RP commands are searched as specified. If unsuccessful and there are no file type extensions, file type extension RUN is used. If the directory was omitted, PROGRAMS is used. Then another search is made for the program files.

Loading Your System from CTD

Before loading your system from Cartridge Tape Drive (CTD), do the following:

- Use BUILD to create your system,
- Use CSYS to copy your system onto the tape. The command string for CSYS is:
RU,CSYS,systemfilenamr, CTD lu,file#
- Legal file numbers start at 0, the first file on the CTD.

On the VCP terminal keyboard:

1. If you are not in VCP mode, press the BREAK key to pass control to the VCP program.
2. Enter: %BDCffb1sc

where:

- | | |
|------|---|
| %BDC | Executes the bootstrap loader program for disc or CTD. |
| ff | File number of system file on CTD. Default file number is 0. |
| b | HP-IB bus address of the unit to load system file from. For the 7908, 7911, 7912, and 7914 discs, the bus address is preset to 0. You can change this address to a value between 0 and 7. |
| 1 | Unit number of CTD. |
| sc | Select code of HP-IB interface card. The select code is switch-selectable on the card. The default value is octal 27. |

Examples

The following examples show the entries to the VCP when booting different systems from Cartridge Tape Drive as described in each example. Note that all keyboard entries are italicized.

1. The system file is located at the first file position on the tape. The device parameters are as follows:

```
HP-IB Bus Address = 0
Unit Number      = 1
Select Code      = 27
```

The following sequence of commands should be entered at the virtual control panel:

```
CI> (Strike the BREAK key)
P 002645 A 077774 B 000000 RW 100003 M 002674 T 02674 T 026655

VCP> %bdc127
BOOT PROCESS COMPLETE

* RTE READY *
:
:
```

Since 0 is the default value for the file number, it does not need to be specified.

2. To load and execute a system located at the fifth file (file number 4) on the tape, and the device parameters as follows:

```
HP-IB Bus Address = 0
Unit Number      = 1
Select Code      = 37
```

The following sequence of commands should be entered from the virtual control panel:

```
CI> (Strike the BREAK key)
P 002645 A 077774 B 000000 RW 100003 M 002674 T 026655

VCP> %bdc40137
BOOT PROCESS COMPLETE

* RTE READY *
:
:
```

The tape will advance to the fifth file before the actual load takes place.

Loading Your System from PROM

Before loading your new system from a PROM module, do the following:

- Burn the system file into the PROM module. If the system image takes more than one PROM module (32k words), the PROM modules must be burned sequentially and the select codes must be set sequentially.
- Determine the absolute file number where the system file resides on the PROM.
- Obtain the select code of the PROM.

Now enter the following from the VCP:

1. Press the BREAK key to pass control to the VCP program if not already in VCP mode.
2. Enter %BRM00sc

where:

- | | |
|------|--|
| %BRM | Execute the PROM loader program for the PROM, and begin execution of the system when the entire system file has been loaded into memory. |
| 00 | Zero, used as placeholders for the bu parameters. |
| sc | Select code of the interface card from which your system will be loading. The select code is switch-selectable on the card. The default select code is 22 octal. |

Loading Your System over a DS Link

Before loading your new system over a DS link, perform the following steps. Refer to the DS/1000-IV Network Manager's Manual for details:

1. Create the merged system file, using BUILD, or use the absolute binary version created from this memory-image file by MI2AB. DS program PROGL will accept either one.
2. Select a source computer system. The source must be disc based, and must be running DS. The source computer and the target computer must be connected by a DS/1000-IV link. If the source computer is directly connected to the destination computer (neighbors), the source computer system must be running DS program PROGL, either version.
3. If the source computer is not directly connected to the destination computer, select an intermediate DS node that is directly connected to the destination system. This intermediate node need not be directly connected to the source computer system, but it must be running DS, including the store-and-forward version of PROGL.

4. Put the merged system file on disc at the source. Give the system file one of the default DS file descriptors Pnnnnn:0:0, where nnnnn is the file number you will use in the boot string at the destination system. If this file descriptor can be P00000:0:, you will be able to default the file number in the boot string, and no special translation subroutine will be required at the source computer.

Alternatively, provide your own translation subroutine named #DNFL from the number you intend to use in the boot string to the actual file descriptor describing the merged system file.

5. Note the select code of the DS link in your target system that leads directly to the neighbor computer running PROGL. If this can be select code 24, you will be able to default the select code in the boot string.
6. If you are using the store-and-forward version of PROGL on an intermediate computer connected to the destination computer, set up the entry points #RMT1 and #RMT2 to indicate first and second choice (if applicable) of source nodes.

With the above steps completed, you can boot the target system at the destination computer VCP terminal. The boot procedure is the same if the VCP terminal is itself remotely located at a directly-connected DS node. When you are at the keyboard of the VCP terminal, you are effectively at the destination system.

Boot the target system:

1. If the VCP terminal is at the destination system, press the BREAK key to get the VCP prompt.
2. Enter %BDSff00sc

where:

- | | |
|------|---|
| %BDS | Executes the DS boot loader program and begins execution of the system when the entire merged system file has been loaded into memory. |
| ff | is a file number not greater than 77777 octal, translatable to the file descriptor of the merged system file to be booted via DS. |
| 00 | Use 00 as placeholders for the bu parameters in the boot string if the ff parameter is not zero. |
| sc | is the select code of the target system DS interface card that is directly connected to the source computer system or to the directly-connected (neighbor) intermediate computer if the source computer is further out on the DS network. |

Loading Your System from Magnetic Tape

Before loading your system from magnetic tape, do the following:

- Use BUILD to create your system,
- Use the CO command from CI to copy your system onto magnetic tape.

On the VCP terminal keyboard:

1. If you are not in VCP mode, press the BREAK key to pass control to the VCP program.
2. Enter: `%BMTffbusc`

where:

- | | |
|-------------------|---|
| <code>%BMT</code> | Executes the bootstrap loader program for magnetic tape. |
| <code>ff</code> | File number of system file on magnetic tape. Default file number is 0, which causes the loader ROM to load the file at the current position of the tape. Any value greater than 0 causes the loader ROM to rewind the tape and then find the file. The first file on the tape is file number 1. |
| <code>b</code> | HP-IB bus address of the unit from which to load system file. The magnetic tape is preset to HP-IB address 4 (the default value) in the Primary System. You can change this address to a value between 0 and 7. |
| <code>u</code> | Unit number of the magnetic tape, normally 0 (the default value). |
| <code>sc</code> | Select code of HP-IB interface card. The select code is switch-selectable on the card. The default value is octal 27. |

Examples

The following examples show the entries to the VCP when booting different systems from magnetic tape as described in each example. Note that all keyboard entries are italicized.

1. The system file is located at the first file position on the tape and the tape has been rewound. The device parameters are as follows:

```
HP-IB Bus Address = 4  
Unit Number      = 0  
Select Code      = 27
```

The following sequence of commands should be entered at the virtual control panel:

```
CI> (Strike the BREAK key)  
P 002645 A 077774 B 000000 RW 100003 M 002674 T 02674 T 026655  
  
VCP> %bmt4027  
BOOT PROCESS COMPLETE  
  
* RTE READY *  
:  
:
```

Since 0 is the default value for the file number, it does not need to be specified.

2. To load and execute a system located at the fifth file (file number 5) on the tape, and the device parameters as follows:

```
HP-IB Bus Address = 4  
Unit Number      = 0  
Select Code      = 37
```

The following sequence of commands should be entered from the virtual control panel:

```
CI> (Strike the BREAK key)  
P 002645 A 077774 B 000000 RW 100003 M 002674 T 026655  
  
VCP> %bmt54037  
BOOT PROCESS COMPLETE  
  
* RTE READY *  
:  
:
```

Because the file number is specified, the tape will rewind and then advance to the fifth file before the actual load takes place.

Loading Your System from 248X Integrated Discs

Before loading your new disc based system into memory from disc, do the following:

- Ensure that the boot extension (BOOTEX) is physically stored beginning at cylinder zero of the disc drive from which the system is to be loaded. Refer to the INSTL chapter in the Utilities Manual and Chapter 9 of this manual for details on installation of the boot extension (BOOTEX).
- Be sure the boot command file, the memory image system file, and the snap file are stored on the same LU that contains BOOTEX.
- Obtain the address, unit number, head number, and the select code of the 248x integrated disc controller from which your system will be loaded.

Now enter the following from the VCP:

1. Press the BREAK key to pass control to the VCP program if not already in VCP mode.
2. Enter %bdibuscfile[,SS]

where:

- %bdi** Execute the disc bootstrap loader program for the 248x Integrated disc. The loader will load the boot extension (BOOTEX) into low memory. BOOTEX will set up the swap area, the memory allocation table, and relink and RP program files specified in the command file. BOOTEX will then copy part of itself to the top of physical memory and will load the system file. When loading is complete, the system starts execution immediately if the suspend parameter SS is omitted.
- b** Disc address of the integrated disc where the target system file resides. The default address is 0.
- b = 0** First fixed disc
 b = 3 Micro floppy
- u** Unit number of the micro floppy drive. The default is 0.
- u = 0** First micro floppy
- sc** Select code of the integrated disc controller. The select code is set via a switch on the card. Refer to the Computer Service and Installation Manual for the switch settings. The default is 32 octal.
- file** Name of the boot command file to be used by BOOTEX. The default file name that BOOTEX will use is /SYSTEM/BOOT.COMD (or SYSTEM for a FMGR disc cartridge).

If you have specifically named your boot command file, you must supply the name to the VCP program to be passed to the boot extension. This allows you to store multiple systems on one disc, and specify to the VCP program which is to be booted.

If you want to execute any of the diagnostic programs, you may pass the name of the diagnostic to be placed into memory by BOOTEX.

,SS When SS is appended to the file name, a “cold” load will result. That is, the system will be loaded into memory by BOOTEX, then BOOTEX will halt. You can then alter the system with VCP commands, and start up by using the %E VCP command.

Note that SS must be separated from the file name with a space or a comma.

If you wish to load a memory image system file directly (the output file from BUILD for a memory based system, for example) the process is similar to that described above, using the name of the memory image system file instead of the boot control file. BOOTEX detects that the file is a system file instead of a boot control file and simply loads the system into memory.

Examples

The following examples show the entries that should be made at the VCP when booting several different systems with the system file at the location described in the example definitions. All user entries are italicized. Both uppercase or lowercase letters can be used.

1. To load and execute a system with a hierarchical file system and a boot command file called /SYSTEM/BOOT.CMD:

Disc Address = 0
Unit Number = 0
Select Code = 27 (disc volume)

Enter the following sequence of commands from the VCP:

```
CI> (Strike the BREAK key)  
P 002675 A 077774 B 000000 RW 100003 M 002674 T 026655
```

```
VCP> %bdi  
BOOT PROCESS COMPLETE
```

```
* RTE READY *  
:  
:
```

Because the default parameters are used, only %bdi needs to be entered to specify that the default system is to be booted from disc.

2. To load the diagnostic program called /SYSTEM/HPIB from the disc with:

Disc Address = 3
Unit Number = 0
Select Code = 32



Enter the following sequence of commands from the VCP:

```
CI> (Strike the BREAK key)
P 002675 A 077774 B 000000 RW 100003 M 002674 T 026655
```

```
VCP> %bdi3032hpib
```

```
12009A HP-IB INTERFACE DIAGNOSTIC
```

```
READY
```

In this example all parameters must be entered, and the diagnostic program specified, since neither is the default value.

3. To cold load a system with a boot command file called /SYSTEM/MUTIGR from the disc with:

Disc Address = 0
Unit Number = 0
Select Code = 32

Enter the following sequence of commands from the VCP:

```
CI> (Strike the BREAK key)
P 002675 A 077774 B 000000 RW 100003 M 002674 T 026655
```

```
VCP> %bdimutigr,ss
BOOT PROCESS COMPLETE
```

```
VCP> %E
```

```
* RTE READY *
:
:
```

The device parameters did not have to be supplied here since all were default values. The first character following BDI is non-numeric, indicating that MUTIGR is text and VCP should use the default parameters. The characters ,SS appended to the file name indicate a cold load. If necessary, prior to entering 'E', you could make changes or load patches to the system.

4. To boot a system from disc with a boot command file called /SYSTEM/BOOT.CMD:

Disc Address = 3
Unit Number = 0
Select Code = 32

Enter the following sequence of commands from the VCP:

```
CI> (Strike the BREAK key)
P 002675 A 077774 B 000000 RW 100003 M 002674 T 026655
```

```
VCP> %bdi3032
BOOT PROCESS COMPLETE
```

```
* RTE READY *
:
:
```

This example shows that the boot command file name does not have to be supplied since the default file name is used.

5. To boot a memory image system file output from BUILD called /SYSTEM/BOOT.CMD with:

Disc Address = 0
Unit Number = 0
Select Code = 22 (disc volume)

Enter the following sequence of commands from the VCP:

```
CI> (Strike the BREAK key)
P 002675 A 077774 B 000000 RW 100003 M 002674 T 026655
```

```
VCP> %bdi0022
BOOT PROCESS COMPLETE
```

```
* RTE READY *
:
:
```

This example shows that the system file name does not have to be supplied since the default file name is used.

6. Load and execute a system from disc, entering the boot commands interactively. BOOTEX and the system and snap files are on the disc described by:

Disc Address = 0
 Unit Number = 0
 Select Code = 32

Enter the following sequence of commands from the VCP:

```

CI> (Strike the BREAK key)
P 002675 A 077774 B 000000 RW 100003 M 002674 T 026655

VCP> %bdiLU1

BOOTEX: SY,SYSTMC           system file on directory SYSTEM
BOOTEX: SN,SNAPC           snap file name
BOOTEX: MC,36             mount disc LU 13
BOOTEX: RP,D.RTR,D.RTR    RP D.RTR
BOOTEX: RP,C             RP CI
BOOTEX: ST              CI is the startup program
BOOTEX: EN             terminate RP sequence
BOOTEX: RV,10          partition 10 pages long
BOOTEX: RV,12          partition 12 pages long
BOOTEX: RV,20          partition 20 pages long
BOOTEX: SW,/SYSTEM/SWAP::::1024 create 1024-block swap file
BOOTEX: EN             terminate command entry
BOOT PROCESS COMPLETE

* RTE READY *
:
:

```

This example shows that LU1, a type-zero file for the terminal, can be used as the file name and boot commands can be entered interactively. If the boot program cannot find the file specified as the boot control file, it will also switch to the interactive mode.

The default directory for command file SYSTEMFILE and the snap file is directory SYSTEM on a hierarchical file system disc volume. If the LU from which the system booted is a FMGR LU, the default is cartridge 0.

Program files in the RP commands are searched as specified. If unsuccessful and there are no file type extensions, file type extension RUN is used. If the directory was omitted, PROGRAMS is used. Then another search is made for the program files.

BUILD Program

Introduction

The program BUILD takes an RTE-A operating system file, its snapshot file, and programs loaded for the system and creates one memory-image file that contains a bootable system image. BUILD will load single-segment and multi-segment CDS programs, shared CDS programs, EMA and SHEMA programs into a memory based system. (All references to CDS apply only to systems with VC+.) It also will load non-CDS programs, however non-CDS programs with overlays cannot be loaded into the memory based system.

Because the original operating system is left unchanged, you can create different memory layouts of the same basic system using BUILD. The term “original system” is used here to describe a system output by the generator, and “bootable system” to describe the output from BUILD.

You can use an output file from BUILD as an input file on another pass through BUILD. When you use a BUILD output file as the original system image input file to BUILD, the previously defined ID segments and memory descriptors are zeroed. This feature eliminates the need for keeping both the original generated system and the BUILD output system (although most users will want to keep the original generated system).

BUILD Program Operation

The BUILD program accepts commands interactively or from a command file. The file names can be specified in the BUILD runstring or interactively in response to the BUILD prompts.

The BUILD Runstring

The BUILD program runstring syntax is shown below.

RU,BUILD,command,list,output,snapshot,system,op

where:

- | | |
|----------|---|
| command | The file or logical unit from which commands are received. The default command device is the terminal from which BUILD is scheduled. |
| list | The file or logical unit to which the listing is sent. The default is to the terminal from which BUILD is scheduled. |
| output | The bootable system file output of BUILD. If this file is not specified in the runstring, BUILD will read the command file for the name or, if interactive, will prompt for the file name. BUILD forces the output file to type 1, with a default size of 2048 blocks (512K byte system). On completion, BUILD truncates the file to the smallest size possible. You can specify the output file size in the file descriptor. |
| snapshot | The snapshot file of the original system. If this file is not specified, BUILD will read the command file or, if interactive, prompt for it. |
| system | The original system image created by the generator, or the output file of a previous run of BUILD. If not specified, BUILD will read the command file or prompt for the file descriptor. (BUILD does not modify this file.) |
| op | An optional parameter defining the exit path on errors when input is from a command file. The default path is the abort (/A) exit. Because this default is in effect until the runstring is sequentially parsed, errors in the runstring will result in the abort exit. This prevents a cascade of errors which would otherwise follow. Unexpected EOF errors on the command file override the default, and cause the end (/E) exit. The options are:

ABORT Abort and purge the BUILD system file created. (/A)
END Exit, saving the system file created. (/E)
CONTINUE Continue processing subsequent commands on all but unexpected command file EOF errors, or errors associated with runstring files. (/C) |

BUILD Commands

The BUILD program has three distinct phases. In each of these phases, different user commands are valid.

BUILD Phase One

During this phase, BUILD parses the runstring and prompts for any vital information that is not specified. Once this information has been supplied, BUILD automatically proceeds to the second phase.

BUILD Phase Two

The second phase of BUILD allows you to define the memory layout for the final bootable system. During this phase, BUILD will prompt for the memory size of the system and the reserved partition layout (note that only reserved partitions are available on memory-based systems).

If you request the automatic partitioning feature (YES response to the prompt), BUILD will create all partitions, sizing them as required for the programs. If you do not desire this feature (NO response), you must specify all of the partitions explicitly. You cannot mix the automatic and explicit partitioning features.

This phase is completed by requesting automatic partitioning, by completely defining the memory partition layout and entering the /C command, or by running out of memory descriptors (as defined at system generation).

BUILD Phase Three

During phase three of BUILD, you can redefine the system security code and load programs into memory. This phase of BUILD recognizes the following commands:

SS,newcode	Change the system security code from the value in the system image to the new value, a number or one or two uppercase characters (FMGR changes lowercase to uppercase).
PT	Display a partition table which describes the occupants of the partitions and the sizes of all previously RP'd programs.
RP,prog[,newname]	Construct an ID segment for the program and load the program into the memory image file. If the program is shareable, its code partition will be shared with any other copies of the program (92078 only).

Following each RP command, subcommands can be entered to modify the way in which the program's ID segment is constructed. These commands refer to the program last loaded using the RP command. If these commands are used before an RP command has been given, an error message will occur.

- PR[IO]RITY],n Set the program priority to n (a number between 0 and 32767). The default value of n is that supplied by LINK.
- PA[RTITION],nn[,C/D] Place the program into partition nn. If this command is not specified, BUILD loads the program into the smallest unused partition that can hold the program. The optional C/D parameter specifies whether code or data is placed in partition nn. Specify 'C' for code, 'D' for data. This command is invalid when the automatic partitioning feature has been specified.
- ST[ART][,p1]..[,p5] Make this program the startup program with the specified RMPAR parameters. If this option is used for more than one program, the last program specified as the start program will be the actual startup program.
- SZ,nn Size the program, including system common (if used), to nn pages. This has the same function, but overrides, the LINK size (SZ) command for non-CDS programs. For CDS programs the SZ command overrides the LINK heap (HE) command. BUILD does not change the stack size (the LINK stack (ST) command). If you want this changed, use LINK before running BUILD. Allows creation of a dynamic buffer which can be managed by the program with the aid of LIMEM. Note that sizing of EMA or VMA programs is not allowed in BUILD. Such programs must be correctly sized when loaded.

The following five commands provide control and instructional functions.

- /A Abort the run of BUILD. The bootable file thus far constructed is purged.
- /E End the run of BUILD. The bootable file constructed is saved.
- /R Restart at the partition definition phase of BUILD. This can be supplied during phases two and three. It results in a complete redefinition of the memory and program layout, voiding all previous ID and partition definitions.
- /C Complete the partition definition phase of BUILD in the non-automatic partition construction case. This phase is normally completed automatically when all partitions and memory are defined.
- ?? This command causes BUILD to output to the list file a description of all commands. The ?? command is only acceptable following the 'BUILD:' prompt.

Shareable EMA

During the RP phase, BUILD saves all information about the size of each shareable EMA area needed by the programs. After receiving the /E command, BUILD creates memory descriptors for the shareable EMA areas.

Each time an EMA area is set up, BUILD prints the message:

Creating shareable EMA partition with nn pages

If there is not enough memory left in the system, or if there are not enough memory descriptors available, a fatal error results; BUILD must be run again. BUILD ignores the specification to put shareable EMA in a specific reserved partition (LINK command SH), because there may be a program occupying the specified reserved partition. As in a disc-based system, all programs that access a given shareable EMA area will access the same area.

There is no specific mechanism in BUILD that allows you to specify bad memory pages and then to have those bad pages automatically skipped. (BOOTEX, the disc-based system boot program, has this capability.) If your system contains bad memory pages, you can use manual partition mode to define the bad page (or area of bad pages) to be a partition; then, during the RP phase, do not assign any programs to the bad partition.

BUILD Examples

In the following example, BUILD is run interactively, specifying only a list file in the runstring. BUILD prompts for the missing parameters (phase one). In this and the second example, user entries are italicized for clarity.

This example includes an attempt to load a duplicate program. As shown, BUILD reported the error, but continued operations. All of the files used in this example are on a remote system, accessed by the DS transparency features of the file system (>5003 indicates that the file is on node 5003). Note that the application program 'APCDS' is a shared CDS program, and that AP.A and AP.B are both using the same code partition.

RU,BUILD,,/HOME/EXAMPLE.LIST>5003

```
*****
*
*   BUILD SYSTEM IMAGE
*   Thu Jan 23, 1986  1:42 pm
*
*   BUILD constructs a bootable system image file given
*   a generated system file, its snapshot, and programs to be
*   loaded into memory. Type /A to abort the build at any
*   time, /E to end.
*
```

Bootable system file ? */SYSTEM/MEMBASED.SYS>5003*

Current file size constrains the system to be under 256 pages.

Snapshot file ? */SYSTEM/SNAP.SNP>5003*

Original system file ? */SYSTEM/SYSTEM.SYS>5003*

```
*****
*
*   The partition layout of the RTE-A system will now be
*   defined. The system requires the first 56 pages of memory.
*   This phase will be completed automatically when all memory
*   or memory descriptors have been defined. A /C also completes
*   this phase and a /R will restart this phase of the build.
*
```

Do you want automatic partition construction (YES/NO) ? *YES*

Physical memory size in K words (nnn) ? *256*

200 pages of memory remaining.

427 memory descriptors remain undefined.

```
*****
*
*   Programs may now be loaded into memory. Type ?? for help.
*
```

BUILD: *RP,/CDSAPPLIC/APCDS.RUN>5003,AP.A*

123 pages of memory remaining.

BUILD: *PT*

prtn num	low page	length	occupant	
1	56	16	AP.A	(data)
2	72	61	AP.A	(data)

BUILD: *RP,FVERI.RUN::PROGRAMS>5003*

91 pages of memory remaining.

BUILD: *PT*

prtn num	low page	length	occupant	
1	56	16	AP.A	(data)
2	72	61	AP.B	(shared code)
3	133	32	FVERI	(data)

BUILD: *RP,/CDSAPPLIC/APCDS.RUN>5003,AP.A*

*error - Program already loaded.

BUILD: *RP,/CDSAPPLIC/APCDS.RUN>5003,AP.B*

75 pages of memory remaining.

BUILD: *PT*

prtn num	low page	length	occupant	
1	56	16	AP.A	(data)
2	72	61	AP.B	AP.A (shared code)
3	133	32	FVERI	(data)
4	165	17	AP.B	(data)

BUILD: */E*

BUILD completed.

Bootable system image in file MEMBASED.SYS::SYSTEM:1:1448:128>5003.

In the following example, partition sizes are specifically assigned when running BUILD interactively. Note that the list, system, and snapshot files are specified in the runstring. In the example, BUILD issued a warning that the RPL checksum values on a type 6 file (DRTR) do not match the system, but proceeded to load the program without altering the type 6 file. An error was issued when an attempt was made to load a program with overlays (EDIT.RUN). This is not a fatal error; BUILD ignores the RP command and continues. The file CI.RUN was specified as the startup program using the ST command.

Note that BUILD loads programs into the reserved partitions on a best-fit basis; that is, programs are RP'd into the smallest partitions that will accommodate them.

RU BUILD,,JANET.LST,JANET.SYS,SNAP,SYSTEM

```
*****
*
*   BUILD SYSTEM IMAGE
*   Thu Jan 23, 1986  1:55 pm
*
*   BUILD constructs a bootable system image file given
*   a generated system file, its snapshot, and programs to be
*   loaded into memory.  Type /A to abort the build at any
*   time, /E to end.
*
Current file size constrains the system to be under 256 pages.
*****
*
*   The partition layout of the RTE-A system will now be
*   defined.  The system requires the first 56 pages of memory.
*   This phase will be completed automatically when all memory
*   or memory descriptors have been defined.  A /C also completes
*   this phase and a /R will restart this phase of the build.
*
```

Do you want automatic partition construction (YES/NO) ? *NO*

Physical memory size in K words (nnn) ? *256*

202 pages of memory remaining

Partition 1 length in pages (nn) ? *32*

170 pages of memory remaining

Partition 2 length in pages (nn) ? *32*

138 pages of memory remaining

Partition 3 length in pages (nn) ? *7*

131 pages of memory remaining

Partition 4 length in pages (nn) ? *13*

118 pages of memory remaining

Partition 5 length in pages (nn) ? *32*

86 pages of memory remaining

Partition 6 length in pages (nn) ? *12*

74 pages of memory remaining

Partition 7 length in pages (nn) ? *14*

60 pages of memory remaining

Partition 8 length in pages (nn) ? 32

28 pages of memory remaining

Partition 9 length in pages (nn) ? /c

28 pages of memory remaining

483 memory descriptors remain undefined

BUILD: *RP IO.RUN::PROGRAMS*

The minimum partition size required for this program is 7 pages.

Its current size is 7 pages.

The program is currently assigned to partition 3.

BUILD: *RP DRTR.RUN::PROGRAMS*

*warning - RPL checksum does not match

The minimum partition size required for this program is 29 pages.

Its current size is 32 pages.

The program is currently assigned to partition 1.

BUILD: *RP DL.RUN::PROGRAMS*

The minimum partition size required for this program is 18 pages.

Its current size is 32 pages.

The program is currently assigned to partition 2.

BUILD: *RP EDIT.RUN::PROGRAMS*

*error - Cannot RP a program with overlays.

BUILD: *RP CI.RUN::PROGRAMS*

The minimum partition size required for this program is 30 pages.

Its current size is 32 pages.

The program is currently assigned to partition 5.

BUILD: *ST*

BUILD: *RP USERS.RUN::PROGRAMS*

The minimum partition size required for this program is 14 pages.

Its current size is 14 pages.

The program is currently assigned to partition 7.

BUILD: *RP WH.RUN::PROGRAMS*

The minimum partition size required for this program is 13 pages.

Its current size is 13 pages.

The program is currently assigned to partition 4.

BUILD: *RP PROMT.RUN::PROGRAMS*



The minimum partition size required for this program is 8 pages.
Its current size is 12 pages.
The program is currently assigned to partition 6.
BUILD: *PT*

prtn num	low page	length	occupant
1	54	32	DRTR (data)
2	86	32	DL (data)
3	118	7	IO (data)
4	125	13	WH (data)
5	138	32	CI (data)
6	170	12	PROMT (data)
7	182	14	USERS (data)
8	196	32	(none)

BUILD: */E:*

BUILD completed. Bootable system image in file /PUBS/BUILD/
JANET.SYS:::1:1568:128

BUILD Command File

The following is a sample command file that could be constructed to supply the necessary inputs. Comments may be included by using a comma to separate the comments from the expected parameter list. If a command has optional parameters (e.g., ST and RP), enough commas must be included as placeholders to be sure all optional parameters are parsed as null strings. Comments will be echoed to the list file, but lines longer than 80 characters will be truncated.

```
:RU,BUILD,CMAND::18,EXMPL,MBSYS,'SNP.B::18,'GEN.B::18
```

where

file CMAND::18 contains the following:

NO,	no automatic partitioning
256,	memory size in pages
5,	size of partition 1
32,	size of partition 2
10,	.
13,	.
28,	.
32,	size of partition 6
/C,	end partition definition
RP,RUNL,,	load RUNL into the system.
PA,4,	place RUNL in partition 4
RP,COMND,,	load COMND into the system.
PA,1,,	place COMND in partition 1
RP,DRTR,D.RTR,	load D.RTR into the system
SZ,32,	size D.RTR to 32 pages
PA,2,	place D.RTR in partition 2
RP,WH,,	load WH into the system
SZ,15,	size WH to 15 pages
PA,5,	place WH into partition 5
RP,APPLN,,	load APPLN into the system
PA,6,	place APPLN in partition 6
ST,ST,FILE,,	make APPLN the startup program; pass parameters
/E,	complete the BUILD process



Boot Extension BOOTEX

The boot process for RTE-A consists of initially loading the boot extension, modifying the system file, loading the system file, and starting up the system.

In the initial load, the boot ROM loader brings the boot extension, BOOTEX, into memory and then starts the system. BOOTEX, which resides on the disc starting at cylinder 0, sector 0, contains the boot program. The boot loader passes a string to BOOTEX which contains the name of either a boot command file (type 4) or an actual system file to be booted into memory (type 1). BOOTEX begins by opening the file passed to it by the boot ROMs. If the file is a type 1, BOOTEX simply loads the file from the disc into memory and starts it.

If the file is a type 4, BOOTEX assumes it is a boot command file. The command file contains the names of the system file to be booted and its associated snapshot file, the names of any type 6 program files to be restored, and the reserved partition definitions for the system. BOOTEX restores each program file, then modifies the system file by setting up the memory descriptors and creating ID segments for those programs to be RP'd, and sets up the swap file and the swap area descriptors. Once this has been done, the system file remains permanently altered and may be booted directly into memory, bypassing the boot command file. There are, however, dangers in a direct boot. These are covered in the section Skipping the Boot Program at the end of this appendix.

The modified system file is now loaded into memory, and if a program has been specified as the startup program it begins to execute.

Boot Program Operation

BOOTEX can be run either interactively or through a command file specified in the boot string (e.g., %BDCffbuscFILENAME). The boot command file must be on the disc LU from which you are booting. If the boot command file is not found on that disc, or if the file is a type 0 file (specifying the terminal), BOOTEX goes into interactive mode and accepts commands from your terminal. In most cases you can run BOOTEX interactively using the command: %BDCffbusc--

If any errors other than "file not found" occur when BOOTEX tries to open the boot command file, it executes a HALT 1 with the FMP error code in the A-Register. Other errors encountered while the boot program is running will cause it to print an error message to the terminal. If an error is not fatal, BOOTEX will continue to read from the boot command file or continue to prompt if running interactively. If the error is fatal, BOOTEX halts the system. Be aware that BOOTEX cannot report any error messages for items such as driver faults or write-protected discs; these conditions must be guarded against before you run BOOTEX.

The following commands are recognized by BOOTEX:

- EC - Echo commands
- MI - Specify mirrored configuration information file (Datapair/1000 only)
- S2 - Identify named file as copy of the system file
- SY - Identify named file as system file
- SN - Identify named file as snap file
- MS - Size memory to specified number of pages
- SA - Size SAM to specified number of words
- MC - Mount specified cartridge
- SC - Designate specified LU as scratch cartridge
- BP - Specify bad memory pages
- RP - Restore specified program
- ST - Specify previously RP'd program as startup
- SZ - Size previously RP'd program to specified number of pages
- PR - Assign named priority to previously RP'd program
- AS - Assign previously RP'd program to named partition
- RV - Specify reserved partition size (in pages)
- SW - Identify named file as swap file
- QU - Specify timeslice quantum and priority
- BG - Set timeslice boundary
- SS - Specify system security code
- AC - Enable, disable session accounting (VC+ only)
- EN - End RP phase, end BOOTEX

ECHO Command

The EC command can be used to echo all commands to the terminal. This command must not be given while in the RP specification phase.

EC Echoes all commands to the terminal.

MI Command

For Datapair/1000 configuration only. This command specifies a mirrored configuration information file. The syntax is:

[MI,filedescriptor]

where the file descriptor is the configuration information file name. Refer to the System Generation and Boot chapter of the Datapair/1000 Reference Manual (part no. 92050-90001).

SYSTEM/SNAP Commands

The first phase of BOOTEX is the system and snap file definition. In this section of BOOTEX, you specify both the system file to be modified and loaded into memory and its associated snap file. The commands to identify these files are:

[S2,filedescriptor]

SY,filedescriptor

SN,filedescriptor

The S2 command is optional, and if specified, must be entered before both the SY and SN commands. S2 specifies a copy of the system that will overwrite the SY file before booting. This will prevent your system from being corrupted if power should fail during system boot-up.

These two commands must be entered before any other commands, and the files **MUST** be found on the boot cartridge. The file descriptor can include directory and subdirectories. If omitted, the default is directory /SYSTEM for a file system volume, or 0 for a FMGR cartridge.

Memory Commands

The system memory size and system available memory (SAM) can be specified with the following commands:

MS,<number of pages>

SA,<number of words>

The MS command sets the memory size to the number of pages specified; this cannot exceed the amount of physical memory present in the processor. The SA command sets SAM to the number of words specified, rounded up to the next page boundary. If you specify a size greater than the physical memory, the error message "Memory size too big" is issued and the MS command is ignored.

Mounting Cartridges

The mount command for a cartridge must be entered before you can enter a command to RP a file on that cartridge or a command to set up a swap file on that cartridge. MC commands cannot appear after the swap file has been declared (by the SW command). The cartridge you are booting from is called the boot cartridge, and is automatically mounted for you by BOOTEX.

MC,+/-LU

The LU may be specified as a negative LU number or a positive alphanumeric crn designator.

Scratch LU

Using the SC command you can specify the disc LU the operating system will use for VMA backing store scratch files.

SC,+/-LU

The LU may have been previously mounted by an MC command, but it need not have been. The same error messages apply to SC as to MC. If a scratch LU is not defined at boot, the first cartridge mounted will be used. The LU must be a FMGR LU; otherwise, an error VM209 (no such directory) will occur when creating a VMA scratch file.

Bad Pages

NOTE

Bad pages cannot be specified in the system or system common area. If there are bad pages in either of these areas, the physical memory must be repaired, replaced, or reordered.

The BP command allows you to specify any bad pages of memory. All bad pages must be specified before any partitions are defined, and must be identified in increasing order. BOOTEX will skip any bad pages as it is defining partitions. The BP command can take either of the following forms:

BP,page number identify a bad page of memory by its page number. This page number is printed in the parity error message.

BP,pnum1,pnum2 identify a block of bad pages from pnum1 through pnum2.

The parity error message, issued when a system parity error is detected, is:

Parity error occurred at physical page nn page address mmmmm

The message identifies the page (nn) and address within the page (mmmmm) at which the error was detected. Often a second message:

Pages actually marked bad (downed) =xx

is issued to specify the number of pages marked as bad.

It is strongly recommended that you maintain a written log of bad pages, and keep this log physically adjacent to the system console for use with the BP command at boot.

Restore Program Commands RP, ST, SZ, PR

You can identify those program files to be relinked and to optionally have ID segments set up for them during the restore-program/link phase. During this phase, you can relink or restore programs (RP), and change their size (SZ) and priority (PR). You can also specify one startup program (ST) and supply RMPAR parameters for that program.

Use the RP command to enter the RP phase. RP commands are the only legal commands during this phase, which must be terminated by an EN command.

RP,name,[newname] Construct an ID segment for the type 6 program in the system. The program is renamed if the newname parameter is entered. If omitted, the program is RP'd under the original name. If the file cannot be opened, the default file type extension .RUN and directory PROGRAMS are appended to the name (i.e., <name>.run::programs) and a second try is made.

ST,[p1,...,p5] Designates the program given in the previous RP command as the startup program with the specified RMPAR parameters. If this command is given for more than one program, the last program so specified will be the actual startup program.

For program CI, the default is to search the file WELCOME<xx>.CMD::SYSTEM and to execute the commands in that file. (In the filename, <xx> is parameter P2 in the ST command.

SZ,nn Size the program given in the previous RP command (including any system common) to nn pages. This command overrides the LINK size command. Note that sizing EMA and VMA programs is not allowed at boot time. You must specify the correct sizes at load time.

PR,n Change the priority of the program given in the previous RP command to priority n (a value from 0 to 32767). The default priority is that supplied by the loader.

The cartridge on which a program file is located must be one that was previously mounted using the BOOTEX MC command. If the CRN/LU parameter is omitted, the first disc mounted is used as the default.

Reserved Partition Assignment

There are two ways to define reserved partitions: by assigning programs or by simply giving the partition size. A previously RP'd program is assigned to a reserved partition by the AS command:

```
AS,program[program[,...]]
```

Up to twenty programs may be assigned to the same partition. Each AS command creates one reserved partition and assigns programs to it. The entry "program:c" assigns the code segment, and the entry "program:d" assigns the data segment of a CDS program. If neither :c or :d is assigned, the default is to :d.

If a bad page is encountered when creating a reserved partition of the necessary size, a warning message is printed and the partition is created in memory following the bad pages:

```
BAD PAGE mm ENCOUNTERED DURING ALLOCATION OF PARTITION num  
nn PAGES WASTED
```

where mm is the number of the bad page that caused the problem, num is the number of the reserved partition being allocated, and nn is the number of pages being skipped due to the bad page. All pages from the current page through the bad page are skipped. (Enter the bad page numbers in the parity error log.)

Reserved Partition Definition

This phase defines reserved partitions without assigning programs to them. The command takes the form:

```
RV,k
```

where k is the number of pages to allocate for the partition. If a bad page is encountered while trying to create the partition, the warning message described in the AS command, above, is printed and the partition is created in memory after the bad page. (Enter the bad page number in the parity error log.)

SWAP File Definition

The swap file command allows you to specify a file to which programs will be swapped.

```
SW,filedescriptor
```

If BOOTEX finds the file given in the command, that file will be used, ignoring any file size parameter in the descriptor. If that file does not already exist, it is created, using all of the parameters in the filedescriptor. The minimum size of the swap file is 3 blocks; the maximum is 4M blocks. (Refer to the Programmer's Reference Manual for instructions for specifying a swap file larger than 14K blocks.)

BOOTEX calculates the maximum size for the swap file size as:

(dynamic memory size) * (#ID segments in system)

This is converted into the file size by creating eight blocks for every 32k words of memory required according to the above algorithm. This will create a huge swap file, so it is often advisable to specify the number of blocks for the swap file size.

BOOTEX searches for the file on the indicated disc (or the first disc mounted if CRN/LU not specified.) If not found, it is created (SWAP::0 on a FMGR disc, /SYSTEM/SWAP.SWP on a disc volume). If this command is not given, a default swap file, SWAP, is created on the system disc. If the disc is a FMGR disc, a security code of SW is used. The message "xxxx BLOCKS IN SWAP FILE" is printed on successful swap file initialization.

Note that the list of mounted disc LUs is maintained in the first two blocks of the swap file. If an error occurs during creation of the swap file, no disc LUs will be available when the system is booted. In this case, they must be explicitly mounted.

Timeslicing, Swapping, Security Code, Session Accounting

Commands to set the timeslicing quantum, the priority swapping boundary, system security code and session accounting can be entered at any point in the boot process (except in the middle of an RP sequence).

QU,timeslice,priority	Set the timeslice quantum to the given value in milliseconds. Set the priority boundary where timeslicing is to begin. If this command is not given, the values will be those set at generation.
BG,boundary	Set the boundary where priority swapping is to begin. If this command is not given, the value will be that set at generation.
SS,newcode	Change the system security code from the value in the system file to the newcode.
AC[,ON/OFF]	For systems with VC+ only. Turns session accounting on or off. AC with no parameters turns accounting on. If this command is not given, accounting defaults to on.

(Note that the system utility METER uses information gathered by the session accounting system; session accounting must be enabled for METER to function properly.)

END Command

The command input to BOOTEX is terminated with an EN command. The EN command also must be used to terminate the RP phase. Because an end of file or CRLF is interpreted as an EN command, a CRLF can terminate the RP phase as well as command input to BOOTEX. When command input to BOOTEX is terminated, BOOTEX loads the modified system into memory and starts it.

Sample Boot Command File

The following is an example of a command file constructed to pass commands to BOOTEX.

```
EC                ,echo command input
SY,SYSTMC        ,system file name
SN,SNAPC         ,snap file name
SS,XX            ,system security code of XX
RP,D.RTR         ,create ID segment for D.RTR
PR,2             ,change D.RTR's priority to 2
RP,CI            ,create ID segment for CI
ST               ,CI is startup program; pass parameters
RP,APPLN         ,restore user program APPLN
EN               ,end RP phase
AS,D.RTR         ,assign D.RTR to reserved partition 1
RV,28           ,create reserved partition 2 with 28 pages
RV,7             ,create reserved partition 3 with 7 pages
RV,12           ,create reserved partition 4 with 12 pages
SW,SWP1:SW:::32 ,initialize file SWP1 with security code SW
                 ,and size as 32 pages
EN               ,terminate command input
```

Skipping the Boot Program

BOOTEX modifies on the disc the system file that is being booted. After the first time through the boot process, the system file will have been modified. You may be tempted to give the system file name in the boot string instead of the boot command file name. This would cause the system file to be loaded into memory without going through the full boot process. This is NOT recommended. Unpredictable results can occur if you do this.

The boot program checks to make sure there is as much physical memory as the system expects, and it makes sure the swap file exists. If there is any change to these items, it will not be caught at boot time unless you go through the whole boot process. Another dangerous condition could occur if a program file is not in the same place on the disc as expected in the present ID segment that was created in the system file by BOOTEX. If the program is being removed under this condition, unpredictable results can occur. If you suspect any discrepancies between the actual program file positions and their positions as reflected in the system file, you should boot from the boot command file.

It is strongly recommended that you always go through the commands in the boot command file.



CS/80 Disc Configuration

This appendix contains the step-by-step procedure for converting your CS/80 disc system to the new standard HP CS/80 disc configuration. Perform each step in the order given, and do not omit any of the procedures. Note that the target LU must be the same physical size as the host LU.

1. Using the CS/80 pushbutton save utility PBSAVE, back up your operating system.
2. Copy the answer file #ANS::16 to a new file using the CI CO command, as

```
CI> CO #ANS::16 #NWANS::16
```

The file name #NWANS is used here as an example; any distinctive file name may be used.

3. Edit the new answer file as follows:
 - a. Change the table configuration and node declaration sections to read:

```
*
* DECLARE DVTs FOR CS/80 DISCS WITH COMPATIBLE CTD -
* LU 16-26,29-31: HP-IB ADDRESS 0
*
DVT,%DD*33::17,M7908_CF:0,LU:16,DP:1:0
DVT,%DD*33::17,M7908_CF:1,LU:17,DP:1:0
DVT,%DD*33::17,M7907FXD_CF:2,LU:18,DP:1:0
DVT,%DD*33::17,M7941_CF:3,LU:19,DP:1:0
DVT,%DD*33::17,M7911_CF:4,LU:20,DP:1:0
DVT,%DD*33::17,M7945_CF:5,LU:21,DP:1:0
DVT,%DD*33::17,M7912_CF:6,LU:22,DP:1:0
DVT,%DD*33::17,M7914_CF:7,LU:23,DP:1:0
DVT,%DD*33::17,M7933_CF:8,LU:25,DP:1:0
DVT,%DD*33::17,M7933_CF:9,LU:26,DP:1:0
*
DVT,%DD*33::17,M7907REM_CF:0,LU:29,DP:1:0
DVT,%DD*33::17,M7907REM_CF:1,LU:30,DP:1:0
DVT,%DD*33::17,M7907REM_CF:2,LU:31,DP:1:0
*
* DECLARE NODES FOR CS/80 DISCS AND COMPATIBLE CTD
*
NODE,16,17,18,19,20,21,22,23,25,26,29,30,31,24
*
```

- b. Change all references from LU MS to LU 17. This can be done using the Edit/1000 global exchange command X, as

```
1$X/:MS/:17//
```

- c. In the section headed "FOUR DEVICES", change the LU references from 25 through 28 to LU 42 through 45.

4. Save the existing device driver %DD*33 to a new name. Using the tape filer utility TF, install modified device driver %DD*33 (Part No. 92077-16668, Revision 2440) as follows:

```
CI> RN %DD*33 %DDH.33          * rename existing driver
CI> TF                          * call tape filer utility TF
TF: CO 24{%DD*33} ::17 V      * copy modified device driver to LU 17
(copy verify message)
TF: EX                          * exit TF
CI> __
```

Note that TF will issue an appropriate message while the file is being copied to disc and verified (V option). Refer to the RTE-A Utilities Reference Manual for details on running TF.

5. Again using the tape filer utility TF, save all files on LUs 17 through the last disc LU number onto a fresh tape (these LUs will be reconfigured as LU 17):

```
CI> TF
TF: DE,,24,VA                  *default destination to CTD LU
TF: GR                          *execute following commands as a group
TF: CO ::17                     *copy command for each LU
.
.
.
TF: CO ::nn
TF: EG                          *end group execute command
(copy verify messages)
TF: EX                          *exit TF
CI> __
```

The V option will cause the files to be verified as they are copied; the A option directs TF to append each file to the preceding file on the tape.

6. Generate the new target system using the edited answer file:

```
RU,RTAGN,#NWANS::16,'NWLST::16,'NWSYS::16,'NWSNP::16
```

You may want to create a transfer file for the generation.

7. Using the CI CO command, move the following files to target LU 16:

```
DL.RUN::PROGRAMS
IO.RUN::PROGRAMS
CI.RUN::PROGRAMS
LI.RUN::PROGRAMS
TF.RUN::PROGRAMS
FC.RUN::PROGRAMS
COMND.RUN::PROGRAMS
DRTR.RUN::PROGRAMS
SWAP.SWP::SYSTEM
PROMPT.RUN::PROGRAMS *VC+ systems only
```

for example, use the CO command as:

```
CI> CO DL.RUN::PROGRAMS DL::16
```

If the target LU is a new file system disc volume, create new /SYSTEM, /PROGRAMS, /LIBRARIES directories using the CI CRDIR command before moving the files to the LU 16 directories.

The new system and snap files ('NWSYS and 'NWSNP created in step 6, above) also must be moved to the newly-created directory /SYSTEM.

8. Create the temporary and permanent boot command files by editing the existing boot command file SYSTEM::16.

To create the temporary boot command file:

- a. Copy boot command file SYSTEM to a new file and edit the copied file:

```
CI> CO SYSTEM::16 NWBOOT::16
CI> EDIT NWBOOT::16
```

The file name NWBOOT is used here for illustration; any distinctive file name may be used.

b. Edit the file NWBOOT to read:

```
*
EC
*
SY,'NWSYS::16
SN,'NWSNP::16
*
RP,DRTR::16,D.RTR
RP,CI::16,CI
RP,CI::16,CM
RP,CI::16,START
ST
*
END
*
SW,SWP::16::3000
* AS,D.RTR
*
END
```

To create the new permanent boot command file, edit the file SYSTEM to read:

```
*
EC
*
SY,'NWSYS
SN,'NWSNP
*
MC,-17
*MC,-18
*MC,-19
*
RP,DRTR.RUN::PROGRAMS,D.RTR
*
RP,CI.RUN::PROGRAMS,CI
*RP,CI.RUN::PROGRAMS (VC+ only)
RP,CI.RUN::PROGRAMS,CM
RP,CI.RUN::PROGRAMS,START
ST,,2
*
END
*
SW,SWP::16::3000
*
AS,D.RTR
*
END
```

9. Install BOOTEX on the target system LU 16, using the INSTL utility, as
CI> INSTL,'NWSNP::16,'NWSYS::16,BOOTEX:-32767:16,16,BOOTEX:-32767:17

If target system LU 16 is a CI volume, you must also run the utility FPUT:

```
CI> FPUT,BOOTEX::17,16,0
```

Specifying an offset of zero (last parameter in the runstring) will install BOOTEX starting at block 0 on the disc. Refer to the RTE-A Utilities Reference Manual for details of both INSTL and FPUT.

10. Now boot the new target system on LU 16, specifying the new boot command file (here NWBOOT) in the boot string. Press the BREAK key to get the VCP prompt, then enter:

```
VCP> %BDC0027NWBOOT
```

As the NWBOOT commands are executed, they are echoed on your screen. In response to the system START prompt following the "Boot process complete" message, enter:

```
START CI  
CI>
```

The target system is now booted and you are running under CI.

11. Install the CTD containing the LU 17 (and on) files, and restore all files to LU 17. If target system LU 16 is a CI volume, rename or purge any directories on LU 16 that are have the same name as the directories to be restored. The command sequence is:

```
CI> IN -17  
CI> MC -17  
CI> TF  
TF: CO 24 17 V  
(copy verify messages)  
TF: EX  
CI>
```

12. Edit the welcome file WELCOME2.CMD::SYSTEM to include the following:

```
*      TARGET SYSTEM WELCOME FILE FOR
*      NEW LU CONFIGURATION, REV 2440
*
RP,FTN7X.RUN::PROGRAMS
RP,LINK.RUN::PROGRAMS
RP,EDIT.RUN::PROGRAMS
SZ,EDIT,32
RP,WH.RUN::PROGRAMS
*RP,PROMT.RUN::PROGRAMS      (VC+ only)
*
*CN,1,20B,PROMT              (VC+ only)
*
```

Refer to the related chapter of this manual, and to the RTE-A Primary System Installation Manual for more information on the Welcome file.

13. Now reboot the system using the edited permanent boot command file SYSTEM. Press the BREAK key to get the VCP prompt, then boot the system with the boot string:

```
VCP> %BDC0027
```

The normal boot sequence commands and prompts will be seen.

14. After successful boot of the system, the following housekeeping procedures are recommended:

- a. Purge all programs copied to LU 16 in Step 7.
- b. Pack LU 16 using either the FMGR PK command or the CI FPACK utility (described in the RTE-A Utilities Reference Manual).
- c. Back up the new system.

Installing Localizable Programs

This appendix discusses the system generation and installation considerations for systems or applications designed to be localizable, such as QDM/1000. Localized applications are programs for which messages and prompts can be translated into a local native language and for which the format for time, date, and other similar data can be customized. The main program code, usually written in a version of English called Native/1000, need not be translated. Such applications use RTE-A's native language support capability whether they are localized or not.

Localized Programs

A program using RTE's native language support contains no hard coded language or custom dependent messages or formatting. A localized program must access messages and prompts located within external binary files or catalogs and use routines driven by formatting tables. These binary message catalog files and formatting files must be installed during system generation and installation.

A localized program can access binary message catalogs using three different methods. Each method requires different amounts of main memory and exhibits different access rates.

Linking the binary message catalogs to the executable image of the program permits the most rapid message access, but requires large amounts of memory. This method requires a separate program image file, or .RUN file, for each native language supported. Different .RUN files are created by modifying the program's LINK command file to call out a different relocatable binary message catalog. User defined search paths (UDSPs) enable selection of the different native language versions of the program.

The other two methods use disc-based message files. In one method, the program accesses the disc-based message catalog file directly. Access to the disc-based binary messages is slow, but the memory requirement is reduced. The other disc-based method uses the message monitor (MSG.M) system utility program to access the disc-based catalogs and pass them to the localized programs that actually report the messages. The message monitor uses the system resources EMA and SAM for string passing to reduce a program's main memory requirements and provide faster message access time than direct disc-based access. For each supported native language, there is a different disc-based message catalog in the /CATALOGS directory.

The custom formatting tables are included in the LANGDEF files, the disc-based catalogs located in the /CATALOGS/ directory. The message monitor uses these files to provide the language dependent table information to a calling program.

File Naming Conventions

An application that has been localized into several languages will have associated with it several binary message catalogs and possibly other language-dependent formatting files for each language. File naming conventions include the language numbering used by RTE-A's native language support capability. For example, the original Native/1000 binary message catalog could be called FILE.C000; the German version of it would be called FILE.C012.

Language ID Numbers

All native language message of LANGDEF files should have a Language ID number (LANGID) as the last three characters of the file name. This number describes the native language that is used in the file. Language ID 000 corresponds to Native/1000, a version of English that the messages are originally written in. Complete information regarding supported languages is available from an HP representative.

Hierarchical File System (CI) File Names

When hierarchical (CI) files are used, the language number becomes the last three characters of the four character type extension. The following file type extensions are used (where nnn is replaced by the language ID number):

- .Snnn - source message catalogs
- .Rnnn - relocatable binary message catalogs
- .Cnnn - disc-based binary message catalogs

FMGR File Names

The names of FMGR files are limited to six characters. The last three characters are the language number. Normally, the first character of the FMGR file name identifies the type of the file. The other two characters describe the product. Because all products may have their message files in the same directory, message catalog names must be unique across the system.

The message catalog naming conventions for FMGR file are (where "xx" is any two alpha-numeric characters and "nnn" is the language ID number):

- <xxnnn - for source message catalogs.
- %xxnnn - for relocatable binary catalog files, both single and multi-module files.
- >xxnnn - for disc-based binary catalog files.

System Generation and Installation

This section discusses the generation and installation of disc-based systems with the VC+ option using RTE-A's native language support capabilities.

System Design Considerations

Before generating your system, review the following design considerations.

Terminals

In session systems (that is, systems with the VC+ option), terminals may be different types. For example, they may be ASCII, Roman8, Kana8, and/or Kanji. Furthermore, the lingual configuration can vary for individual terminals within the system. The lingual configuration can be switched at logon.

Memory Requirements

Systems using RTE-A native language support require extra memory for the following:

1. Localized RTE-A system messages, in addition to the Native/1000 messages from the %\$M000 relocatable. The Native/1000 system messages must be relocated for all systems, and have a size of approximately 3K words.
2. Space for the partition for the message monitor MSG.M, which has a standard size of 36K words. Although MSG.M is swappable, better system performance is achieved if MSG.M has a reserved partition.

System Generation

This section describes those specific steps required at system generation to use RTE-A's native language support features.

The three steps to be taken at system generation are as follows:

1. In the driver partition phase, relocate the appropriate driver.
2. In the table generation phase, set the serial driver for each terminal using a lingual configuration other than Native/1000 for 8 data bits.
3. In the memory allocation phase, relocate system message relocatables.

System Installation

The following is a list of steps for installing session systems. In the event an application (such as QDM/1000) has been made localizable but is to be used only in the Native/1000 lingual configuration, some of the steps involving the placement of localized files are not applicable.

1. Place all binary message catalogs into directory /CATALOGS.
2. Install the message monitor, if required.
3. Place xxxxxx.Hnnn into /SYSTEM directory (where nnn is the language ID number).
4. Set up a directory /HELPxxx and place localized help messages into it (where xxx is the language ID number).
5. Set up a directory /PROGRAMSxxx and put applicable localized programs into it (where xxx is the language ID number).
6. Place unlocalized programs into directory /PROGRAMS.
7. Set up a text file NLTERMCAP.TXT (to be used by NLSID) and put it into the directory /SYSTEM.
8. For each user, create a CI command file that will run NLSID and PATH to set a language ID and program search path.
9. Run program GRUMP to set the session startup command as scheduling CI to execute the command files created in step 7.
10. Modify the welcome file to RP the message monitor, MSG.M.
11. Bootup the new system.

Placing Binary Message Catalogs into /CATALOGS

Place the binary message catalogs with the format >xxnnn or xxxxxx.Cnnn into a directory /CATALOGS.

The LANGDEF files (>LGxxx) are the binary message catalogs that define table-driven language-dependent information read by a library routine NInfo. The binary message catalog >MSxxx is an error message catalog for the message monitor MSG.M.

Both the >LGxxx and >MSxxx files localized to a native language as well as the >LG000 and >MS000 of Native/1000 must be in the directory /CATALOGS. For example, if French is the native language, then >LG007, >MS007, >LG000, and >MS000 must be in /CATALOGS. When the application uses only Native/1000, /CATALOGS requires only >LG000 and >MS000.

Installing the Message Monitor (If Required)

When an application (such as QDM/1000) requires the message monitor, it must be linked and installed using the MSGM.LOD command file. The command file produces MSGM.RUN which must later be RP'd as MSG.M.

Placing Help Files xxxxx.Hnnn in the /SYSTEM Directory

Localized help files with the format xxxxx.Hnnn, where nnn is the native language number of the language used in the file, must be placed in the directory /SYSTEM. Files with the form xxxxx.Hnnn are localized versions of Native/1000 files with the form xxxxx.HLP.

Creating Directory /HELPxxx for Localized Help Files

Create a directory /HELPxxx, where xxx corresponds to the language number of your native language, and place localized ASCII (type 4) help files for programs such as CI and DL into it.

Placing Localized Programs into /PROGRAMSxxx

Every program's search path can be specified for every user with the program PATH. As a result, you can place the localized programs into a directory such as /PROGRAMSxxx rather than in the directory /PROGRAMS, with some exceptions. The following programs must be in the directory, /PROGRAMS:

1. Programs PMxxx.RUN which are localized versions of PROMT.RUN.
2. A program LOGON.RUN which does not have an independent local version.
3. Programs which use only the catalog reading accessing method, such as MSG.M.

Placing Unlocalized Programs in Directory /PROGRAMS

Along with the programs described above, place all unlocalized programs into the directory /PROGRAMS. MSGM.RUN must be in this directory.

Setting Up NLTERMCAP.TXT

NLSID is a utility program that sets the native language ID to a terminal LU. See the section Using NLSID to Set Terminal Language ID for details. NLSID requires a file NLTERMCAP.TXT to provide information about the languages supported by the terminal LUs. This section describes setting up the NLTERMCAP.TXT file.

The format of the file is as follows:

(The first column of the NLTERMCAP.TXT file)

NL:<lu number> <language number>,<language number>...

The line must start with NL in the first two columns or else it will be ignored by NLSID. Immediately follow NL by a colon and an LU number. Separate languages by commas or spaces. A non-digit starts a comment which continues until the end of the line.

There can be more than one line with the same LU number. For example, an HP 2628A (which has the ROMAN8 character set) at LU 78 could have the lines:

(The first column of the NLTERMCAP.TXT file)

NL:78 0
NL:78 1, 2, 3

Once created, the NLTERMCAP.TXT file should be in the /SYSTEM directory.

Using NLSID to Set Terminal Language ID

NLSID allows you to set the language number in the terminal LUs device table and to display the current language ID or a terminal LU.

The runstring is:

[RU] NLSID [[LU:n] [language ID]]

parameters:

n: LU number

language ID: Either language number or language name

Examples:

```
CI> NLSID          ..display current language name on your terminal
CI> NLSID LU:5     ..display current language name on LU 5
CI> NLSID 291      ..set RTE-Japanese to your terminal
CI> NLSID RTE-JAPANESE ..equivalent to above
CI> NLSID LU:3 FRENCH ..set language number 7 (FRENCH) to LU 3
CI> NLSID LU:5 3    ..set language number 3 (DANISH) to LU 5
```

In a VC+ system, the NLSID program is usually run from the CI command file when each user logs on. The command file sets the language number for the terminal. However, in addition to setting the language number, the command file must specify the search path using the utility program PATH so that the directory /PROGRAMSxxx (where xxx represents the language ID) is searched before the directory /PROGRAMS. The directory /PROGRAMSxxx contains corresponding lingual versions of the programs. Refer to the RTE-A User's Manual (part no. 92077-90002) for details on the utility PATH.

The following example shows the use of the utilities PATH and NLSID in the CI command file:

```
If
NLSID RTE-Japanese
Then
PATH 1 . /PROGRAMS291 /PROGRAMS ....change search path for LangID 291
Else
PATH 1 . /PROGRAMS .....set ordinary search path when NLSID fails
Fi
```

When NLSID successfully sets the requested Language ID in the terminal's device table, it returns zero (true status) to the calling program. Otherwise, it returns non-zero (false status). If NLSID finds an error, such as an unsupported language name, it will set the terminal's device table to language number 0 (Native/1000). Refer to the Error Message section for a description of NLSID errors.

Preparing the Welcome File

WELCOME_{xx}.CMD::SYSTEM is the command file for the first scheduled CI program at system bootup. The message monitor must be RP'd in the welcome file. Make sure that MSGM.RUN is in the /PROGRAMS directory when the welcome file is executed.

Booting Up the New System

Refer to the VCP Boot Information appendix of this manual for details concerning booting the system.

Localizable System and Subsystem Files

Table L-1 lists the localizable system and subsystem files, the names of their source and binary catalogs, and their proper locations in various directories.

Table L-1. Localizable System and Subsystem Files

Program	Access Type	Source Catalog	Binary Catalog	Where to place on session system	Remarks
GENCAT	cat	<GC000	>GC000	/programs	0)
MSG.M	cat	<MS000	>MS000	/programs	
NLSID	rel	<ID000	%ID000	/programsxxx	
NLTM	rel	<TM000	%TM000	/programsxxx	
OS ERLOG	sys	<\$M000	;%M000	(OS message map)	
CI	rel	<CI000 <CA000 <CR000	%CI000 %CA000 %CR000	/programsxxx	for CISUB for CI LIB
CIX	rel	<CX000 <CR000	%CX000 %CR000	/programsxxx	
WH	rel	<WH000	%WH000	/programsxxx	SPOOL SPOOL SPOOL
SP	rel	<SP000	%SP000	/programsxxx	
SPRT	rel	<RT000	%RT000	/programsxxx	
OUTPT	rel	<PT000	%PT000	/programsxxx	
IO	rel	<IO000	%IO000	/programsxxx	
LI	rel	<CR000	%CR000	/programsxxx	
RS	rel	<RS000	%RS000	/programsxxx	
DL	rel	<DL000	%DL000	/programsxxx	
LOGON	rel	<LG000	%LG000	/programs	1)
PROMT	rel	<PM000	%PM000	/programs	2)
PMxxx	rel	<PMxxx	%PMxxx	/programs	2)
D.ERR	rel	<DE000	%DE000	/programs	1)
CLSDDS	rel	<CL000	%CL000	/programs	
LINK	cat			/programs	
EDIT	rel&hlp	<ED000	%ED000	/programs	
FTN7x	rel	<FX000	%FX000	/programsxxx	
DEBUG	cat	<DB000	>DB000	/programs	
<p>0) This program has a system attribute and multiple lingual versions of catalogs should be implemented.</p> <p>1) This program has a system attribute and multiple lingual versions of relocatable catalogs must be relocated.</p> <p>2) Localized version of PROMT should be PMxxx which should be placed into directory/programs.</p>					

Table L-1. Localizable System and Subsystem Files (cont.)

Program	Access Type	Source Catalog	Binary Catalog	Where to place on session system	Remarks
FORMT	rel	<FT000	%FT000	/programsxxx	
FORMC	rel	<FC000	%FC000	/programsxxx	
FOWN	rel	<FW000	%FW000	/programsxxx	
FPACK	rel	<FP000	%FP000	/programsxxx	
FREES	rel	<FR000	%FR000	/programsxxx	
FVERI	rel	<FV000	%FV000	/programsxxx	
PRINT	rel	<PR000	%PR000	/programsxxx	PRINT
PR NO	rel	<P0000	%P0000	/programsxxx	PRINT
TF	cat	<TF000	>TF000	/programs	
IMAGE/1000-II					
QUERY	rel & cat	<QY000	%QY000 & >QY000	/programsxxx	5)
DBDS	rel	<DB000	%DB000	/programsxxx	
DBUTL	rel&hlp	<UT000	%UT000	/programsxxx	
DBL0D	rel	<L0000	%L0000	/programsxxx	
DBSTR	rel	<ST000	%ST000	/programsxxx	
DBRBR	rel	<RB000	%RB000	/programsxxx	6)
DBRFR	rel	<RF000	%RF000	/programsxxx	6)
DBARC	rel	<AR000	%AR000	/programsxxx	6)
DBSPA	rel	<SA000	%SA000	/programsxxx	6)
DBBLD	rel	<BL000	%BL000	/programsxxx	
DBULD	rel			/programsxxx	4)
DBRST	rel			/programsxxx	4)
DBMS.LIB	rel	<LB000	%LB000		4)
DBMON				/programs	3)
DEMON				/programs	3)
<p>3) Not localizable.</p> <p>4) Necessary message modules will be relocated from DBMS.LIB which includes %LB000.</p> <p>5) The sources message catalog contains two target types, RELOCATABLE and CATALOG.</p> <p>6) IMAGE library DBMXn.LIB should be made for every language.</p>					

Error Messages

NLSID: NIInfo error number <error number>.

This message is issued when no language ID parameter is given, but NIInfo returned an error. The program terminates after issuing this message.

<language ID> is the current language for LU <user's lu>.

This message is issued when no language ID parameter is given in the runstring. It tells you what the current language is. The program terminates after issuing this message.

NLSID: Bad lu number.

This message is issued when the LU number in the runstring is not a number or out of range. The program terminates after issuing this message.

NLSID: Lu <lu number> out of range; -1<lu>256.

This message is issued when the LU number in the runstring is out of range, -1<lu>256. The program terminates after issuing this message.

NLSID: Lu <lu number> is not terminal lu.

This message is issued when the LU number in the runstring is not the terminal LU. The program terminates after issuing this message.

NLSID: For session system, LangID cannot be changed other than your log lu.

This message is issued when you try to change langid other than your log on LU for session system. For a session system, only langid on your log LU can be changed. The program terminates after issuing this message.

NLSID: NIInfo error <error number>.

This message is issued when NIInfo returns an error. The next two messages explain the common errors, others are just reported by number.

Native Language Support not installed.

This message is issued when NIInfo returns error 1 (native language support installed) or error -407 (MSG.M schedule error) when the user requested something other than NATIVE/1000.

Language <language ID> is not configured on this system.

This message is issued when NIInfo returns error 2 (language not configured).

NLSID: Bad /system/NLTERMCAP.TXT file
Expected 'NL:<lu number>' but found:
<bad lu number>

This message is issued when the NITermCap file has a line that starts with NL but does not have an ':<lu>' after it.

NLSID: Language number <lang#> not supported on LU <lu #>.

This messages is issued then the NITermCap file has been read but the requested language number is not found. The first parameter is the language number and the second is the LU number.

NLSID: Language number <lang#> is not supported on this system.

This message is issued when the requested langid is not in the system language table.

NLSID: Program <prog name> does not have an ID segment!

This message is issued when the program that should be the primary program does not have an ID segment.

Native/1000 set as the Native language for this terminal.

This message tells you that an error occurred while the language was being set and Native/1000 is being used instead of the language that you requested. The message is issued after one of above error messages is issued.

Index

Symbols

/HELP Directory Creation, L-6
/PROGRAMS Directory, L-6

Numbers

248X Integrated Disc Loading, H-15
7902 Disc Drive Special Considerations, C-19
7908M Disc Drives, C-25

A

AC, J-7
ALIGN, 4-6, 5-7
Allocating an Extended Schedule Table, 7-6
Allocating Class Numbers, 7-1
Allocating Debug Table Entries, 7-2
Allocating Extended System Available Memory (XSAM), 7-4
Allocating ID Segments, 7-2
Allocating Memory Descriptors, 7-3
Allocating Resource Numbers, 7-2
Allocating System Available Memory (SAM), 7-4
Allocating System Message Block, 7-8
Answer File
 Building, 6-10
 Primary System, G-1
 System Generation, G-1
Arrangement of Disc Cylinders, C-5
AS, 9-7
Autoboot Switch Settings, 9-24
Automatic Boot Switch Settings, 10-13

B

Background Priority, 7-5
Background Priority Command BG, 7-13
Backing Up the Target System, 9-23, 10-12
Bad Pages, 9-3
BG, 7-13, 9-6, J-7

Binary Message Catalogs for Localizable Programs, L-5
BLOCC Command, 4-5
BOOT Command File, 9-1, J-8
 Sample, 9-9
BOOTEX, J-1
BOOTEX for New Discs, 9-11
Boot Extension BOOTEX, J-1
Boot Information, H-1
Boot Preparation, 9-13
Boot Procedure, H-1
Bootable Medium
 CTD, 10-7
 Disc Volume, 10-7
 Magnetic Tape, 10-7
 Merged System File, 10-7
 PROM, 10-7
Booting from Disc, Using BOOTEX, 10-8
Booting from FMGR Cartridge, 10-8
Booting the Target System, 9-19, 10-9
BP, 9-3
BUILD, 10-2
BUILD Command File, I-11
BUILD Program, I-1

C

CATALOGS, 9-16
CDS Library, 11-11
Checking Operation of Target System, 9-21
CI Commands Implemented in Separate Programs, 9-17
CI Directories & Programs, 9-15
CLAS, 7-10
Class Number Command CLAS, 7-10
Class Numbers, 7-1
CM as a Secondary Program, 9-22
CMDFILES, 9-17
COM, 7-16
Command File, 2-2, 10-5
Command File Installation, 9-10
Commands
 ALIGN, 4-6
 Background Priority BG, 7-13
 BLOCC, 4-5
 BUILD, I-3
 Class Number, 7-10
 Concurrent User US, 7-15
 DISPLAY, 4-5
 DVT, 6-14

- END, 4-6, 6-19, 6-21
- ID Segment ID, 7-11
- IFT, 6-11
- INT, 6-21
- Labeled Common, 7-15
- LENTRIES, 4-6
- Library LIB, 7-17
- LOCC, 4-4
- LOGOF Buffer Limit LB, 7-15
- Memory Allocation Phase, 7-10
- Memory Descriptor RS, 7-12
- MSEARCH, 4-4
- NODE, 6-20
- Partitioned, 4-9
- PR, 10-4
- PT, 10-3
- Quantum Timeslice QU, 7-14
- RELOCATE, 4-3, 4-4
- Resource Number RESN, 7-11
- RP, 10-4
- Shared Program SP, 7-14
- Spooling Limits SL, 7-13
- SS, 10-3
- ST, 10-4
- System Available Memory SAM, 7-12
- System Memory Block MB, 7-14
- System Relocation Phase, 4-3
- Table Generation Phase, 6-10
- TG, 4-7
- Unlabeled (Blank) Common COM, 7-16
- Concurrent User Command US, 7-15
- Concurrent Users, 7-6
- Configuring a Disc Subsystem, C-1
- Configuring CS/80 Disc, C-10
- Configuring Non-CS/80 Discs, C-4
- Conserving Space, 3-2
- Creating a Merged System File, 10-2
- CS/80 Disc Configuration, C-10, K-1
- CTD System Loading, H-9

D

- Debug Table Entries, 7-2
- Default DVT Entries by Driver Name, F-1
- Default File, D-2
- Default GEN Record, D-2
- Default Libraries, 7-9
- Defaults, 6-8
 - IFT and DVT Parameter, 6-8
- Device Driver Tables, E-1
- Device Tables, 6-1
- DISPLAY, 4-5
- Directories
 - CATALOGS, 9-16
 - CMDFILES, 9-17
 - PROGRAMS, 9-16
 - SCRATCH, 9-17

- SYSTEM, 9-16
- Disc Configuration, C-3
- Disc Cylinder Arrangement, C-5
- Disc Drives Supported, E-1
- Disc Subsystem Categories, C-2
- Disc Subsystem Configuration, C-1
- Disc Volume Mounting, 9-4
- Disc-Based System Installation Procedure, 9-1
- Driver Modules, 4-2
- DS Link Loading, H-11

E

- DS Transparency Software Installation, 9-25
- DVT, 6-14
- DVT Entries by Driver Name, F-1
- EC, 9-2
- Echoing Commands, 9-2
- EN, 9-9
- END, 4-6, J-7
- END Command, 6-19, 6-21
- Entry Records, D-4
- Error Information, VCP, H-1
- Error Messages, A-1
 - BUILD Program, A-9
 - Cartridge Tape Loader, A-12
 - Disc Loader, A-13
 - DS/1000 Loader, A-12
 - HP 12022A Disc Interface Loader, A-14
 - Localizable Applications, L-11
 - Mag Tape Loader, A-13
 - Other Loader Errors, A-14
 - PROM Card Loader, A-12
 - RTAGN Errors, A-1
 - VCP Loader, A-12
- Establishing New CI Directories & Programs, 9-15
- Extended Schedule Table, 7-6
- Extended System Available Memory (XSAM), 7-4

F

- File Naming Conventions, Localizable Programs, L-2
- Files Required for Memory-Based System, 10-1
- Format System, 9-18
- FPUT, 9-11
- FPUT and BOOTEX for New Disc Config., 9-11
- Generator Running, 2-1
- GRUMP, 11-1

G

GEN Records, D-1
Generation File Entries, E-5
 HELP, 9-16
 LIBRARIES, 9-16
Group and User Management Prog (GRUMP),
 11-1

H

HELP, 9-16
HELP Directory Creation, L-6
Help Files for Localizable Programs, L-6

I - J - K

ID, 7-11
ID Segment Command ID, 7-11
ID Segments, 7-2
IFT, 6-11
IFT and DVT Parameter Defaults, 6-8
IFT/DVT, Worksheet, 6-5
INLTERMCAP.TXT, L-6
INT Command, 6-21
Initialization Phase, 3-1
Installing a Disc-Based System, 9-1
Installing a Memory-Based System, 10-1
Installing DS Transparency Software, 9-25
Installing Localizable Programs, L-1
Installing Optional Subsystems, 9-24
Installing Session Systems, Localizable
 Programs, L-5
Installing System, Snapshot, and Command
 Files, 9-10
Interface Driver Tables, E-1
Interface Tables, 6-1

L

LANGID, L-2
Labeled Common Commands, 7-15
Language ID Numbers, Localizable Programs,
 L-2
LB, 7-15
LENTRIES, 4-6
LIB, 7-17
LIBRARIES, 9-16

Libraries, 4-2
Library Command LIB, 7-17
Links, 3-3
List File, 2-2
LOCC, 4-4
LOGOF Buffer Limit Command LB, 7-15
LOGOF buffer limits, 7-7
LOGON, 11-6
Loading Your System from 248X Integrated
 Discs, H-15
Loading Your System from CTD, H-9
Loading Your System from Disc, H-3
Loading Your System from Magnetic Tape,
 H-13
Loading Your System from PROM, H-11
Loading Your System over a DS Link, H-11
Localizable Program Installation, L-1
Localizable Programs, L-8
 Binary Message Catalogs, L-5
 Error Messages, L-11
 File Naming Conventions, L-2
 Help Files, L-6
 Language ID Numbers, L-2
 Memory Requirements, L-3
 System and Subsystem Files, L-9
 System Design Considerations, L-3
 System Generation, L-3
 System Generation and Installation, L-3
 Welcom File, L-8
Localizable System and Subsystem Files, L-9,
 L-10

M

MAC Disc Drives, C-28
Magnetic Tape System Loading, H-13
Master Security Code, 9-6
MB, 7-14
Memory Allocation Phase, 7-1
Memory Allocation Phase Commands, 7-10
Memory Descriptor Command RS, 7-12
Memory Descriptors, 7-3
Memory Map Sample, 5-6
Memory Requirements, Localizable Programs,
 L-3
Memory Size, 9-3
Memory-Based System installation, 10-1
Merged System File Creation, 10-2
Merged System File on Bootable Medium, 10-7
Message Catalog Relocation, 7-16
Message Monitor MSG.M, L-1
Microsystems with Integrated Peripherals
 (248x), C-23
Mirrored Disc Configuration, 9-2
Model 6 Integrated Disc Drives, C-21
Module Flag Records, D-4

Mounting Disc Volumes, 9-4
MSEARCH, 4-4
MSG.M, L-1
Multiuser Accounts, 11-5
Multiuser and Spooling Setup, 11-1

N

Native/1000, L-1, L-2
NODE Command, 6-20
Node Lists, 6-2
 Configuration, 6-3

O

OS Module/Driver Partitioning, 5-1
OUTPT, 11-10
Overlaying Files, 2-4

P

PA, 4-9, 10-4
Parameter Default Records, D-1
Parameters, System, 9-6
Partition Considerations, 5-8
Partition Samples, 5-8
Partitionable System Modules, Tag Area
 Requirements, 5-4
Partitioned Command, 4-9
Partitioning Concerns, 4-7
Partitioning GEN Records, D-5
PR, 10-4
PROGRAMS, 9-16
PROM Loading, H-11
PROMT, 11-6
Preparation for Boot, 9-13
Primary System Generation Answer File, G-1
Processor Card Switch Settings, 9-24, 10-13
Program Restoration, 9-6
Programs
 PA, 10-4
 SZ, 10-4
PT, 10-3

Q

QU, 7-14, 9-6, J-7
Quantum Timeslice Command QU, 7-14

R

RELOCATE Command, 4-3
RESN, 7-11
Relocating Modules, 4-1
Reserved Partition, 9-7, J-6
Resource Number Command RESN, 7-11
Resource Numbers, 7-2
Restoring Programs, 9-6
RP, 10-4
RS, 7-12
RTAGN Runstring, 2-1
Running the Generator, 2-1
RV, 9-8, J-6

S

SAM, 7-4, 7-12
SCRATCH, 9-17
Scratch LU, 9-5
SEARCH, 4-4
Security Code, 9-6, J-7
Security Tables, 7-8
Session Accounting, 9-5, J-7
Session System Installation, Localizable
 Programs, L-5
Setting Master Security Code, 9-6
Setting Switches for Autoboot, 9-24
Setting Switches for Automatic Boot, 10-13
Setting System Parameters, 9-6
Setting Up System Common, 7-7
Setting Up the Startup Program & WELCOME
 File, 9-22
Setting Up the Swap File, 9-4
Shared CI, 11-11
Shared Program Command SP, 7-14
Shared Programs, 7-5
Skipping the Boot Program, J-8
SL, 7-13
SMP, 11-10
SN, 9-2
Snapshot File, 2-3
Snapshot File Installation, 9-10
SP, 7-14, 11-10
SPGET, 11-10
Specifying Background Priority and Timeslice
 Quantum, 7-5
Specifying Bad Pages, 9-3
Specifying Concurrent Users, 7-6
Specifying LOGOF buffer limits, 7-7
Specifying Memory Size, 9-3
Specifying Security Tables, 7-8
Specifying Shared Programs, 7-5
Specifying Spool Buffer Limits, 7-5
Specifying System Default Libraries, 7-9
Specifying System Memory Block, 7-6

- Specifying the Scratch LU, 9-5
- Specifying the System, 9-2
- Spool Buffer Limits, 7-5
- Spooling Initialization and Use, 11-7
- Spooling Limits Command SL, 7-13
- Spooling Setup, 11-1
- SS, J-7
- ST, 10-4
- Standard Generation File Entries, E-5
- Startup Program, 9-22
- Subsystem Installation, 9-24
- SW, J-6
- SWAP File, J-6
- Swap File, 9-4
- Swapping, J-7
- Switch Settings, Processor Card, 10-13
- Switch Settings for Autoboot, 9-24
- Switch Settings for Automatic Boot, 10-13
- Switch Settings for Processor Card, 9-24
- SY, 9-2
- SYSTEM, 9-16
- Sys Gen and Installation, Localizable Programs, L-3
- System Available Memory (SAM), 7-4
- System Available Memory Command SAM, 7-12
- System Common, 7-7
- System Default Libraries, 7-9
- System Design Considerations, Localizable Programs, L-3
- System File, 2-3
- System File Installation, 9-10
- System Generation Answer File, G-1
- System Generation, Localizable Programs, L-3
- System I/O Configuration, Worksheet, 6-4
- System Memory Block, 7-6
- System Memory Block Command MB, 7-14
- System Message Block, 7-8
- System Message Catalog Relocation, 7-16
- System Modules, 4-1
- System Parameters, 9-6
- System Partitioning Concerns, 4-7
- System Partitioning Records, D-4
- System Relocation Phase, 4-1
- System Relocation Phase Commands, 4-3
- SZ, 10-4

T

- Table Generation Phase, 6-1
- Commands, 6-10
- Tables
- Device, 6-1

- Interface, 6-1
- Tags, 4-7
- Target System
- Backing Up, 10-12
- Bootup, 10-9
- Testing, 10-12
- Target System Backup, 9-23
- Target System Booting, 9-19
- Terminal Language ID, L-7
- Testing the Target System, 10-12
- TG, 4-7
- Timeslice Quantum, 7-5
- Timeslicing, J-7
- Turning Session Accounting On/Off, 9-5

U

- Unlabeled (Blank) Common Command COM, 7-16
- US, 7-15

V

- VCP Boot Information, H-1
- VCP Default Parameters for Boot, H-3
- VCP Errors, H-1

W

- WELCOME File, 9-22
- Welcome File for Localizable Programs, L-8
- Worksheet
- IFT/DVT, 6-5
- System I/O Configuration, 6-4
- Worksheets
- Initialization and System Relocation, B-2
- Memory Allocation, B-8, B-9
- System I/O Configuration, B-5, B-6
- Table Generation, B-7

X - Y - Z

- XSAM, 7-4

Manual Part No. 92077-90034
Printed in U.S.A. August, 1987
E0887

