HEWLETT
PACKARD

RTE-A

System Generation and Installation Manual

Software Services and Technology Division
11000 Wolfe Road
Cupertino, CA 95014-9804

Manual Part No. 92077-90034
E0495

Printed in U.S.A. April 1995
Eighth Edition

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.

RESTRICTED RIGHTS LEGEND

Use, duplication, or disclosure by the Government is subject to restrictions as set forth in subparagraph (c) (1) (ii) of the Rights in Technical Data and Computer Software clause at DFARs 252.227.7013.

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. (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	Dec 1983	
Reprint	Dec 1983	Update 2 incorporated
Update 3	Jun 1984	Add 15-Mb disk, MAC disk, 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)
Update 1	Jan 1989	Rev. 5010 (Software Update 5.1)
Fifth Edition	Jul 1990	Rev. 5020 (Software Update 5.2)
Sixth Edition	Dec 1992	Rev. 6000 (Software Update 6.0)
Seventh Edition	Nov 1993	Rev. 6100 (Software Update 6.1)
Eighth Edition	Apr 1995	Rev. 6200 (Software Update 6.2)

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 *RTE-A System Design Manual*, part number 92077-90013, 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

Chapter 2
Running the Generator

Chapter 3
Initialization Phase

Chapter 4
System Relocation Phase

Chapter 5
OS Module/Driver Partition Phase

Chapter 6
Table Generation Phase

Chapter 7
Memory Allocation Phase

Chapter 8
System Installation Concepts and Considerations

Chapter 9
Disk-Based System Installation Procedure

Chapter 10
Installing a Memory-Based System

Chapter 11
Multiuser and Spooling Setup

Appendix A
Error Messages

Appendix B
Worksheets

Appendix C
Configuring a Disk Subsystem

Appendix D
GEN Records

Appendix E
Device and Interface Driver Tables

Appendix F
Default DVT Entries By Driver Name

Appendix G
Example System Generation Answer File

Appendix H
VCP Boot Information

Appendix I
BUILD Program

Appendix J
Boot Extension BOOTEX

Appendix K
Installing Localizable Programs

List of Illustrations

Tables

1

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 disk-based RTE-A system.

Chapter 10 provides the installation procedure for a memory-based RTE-A system.

Chapter 11 explains how to download a memory-based system over a LAN and how to use the remote VCP.

Chapter 12 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 disk 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 disks.

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 DDS tape, CS/80 cartridge tape, or magnetic tape, you must restore the software onto the existing system. The media restoration procedures are given in the *RTE-A Primary System Software Installation Manual,* part number 92077-90038, and the utilities used for the system generation are described in either the *RTE-A User's Manual*, part number 92077-90002, or the *RTE-A Backup and Disk Formatting Utilities Reference Manual,* part number 92077-90004.

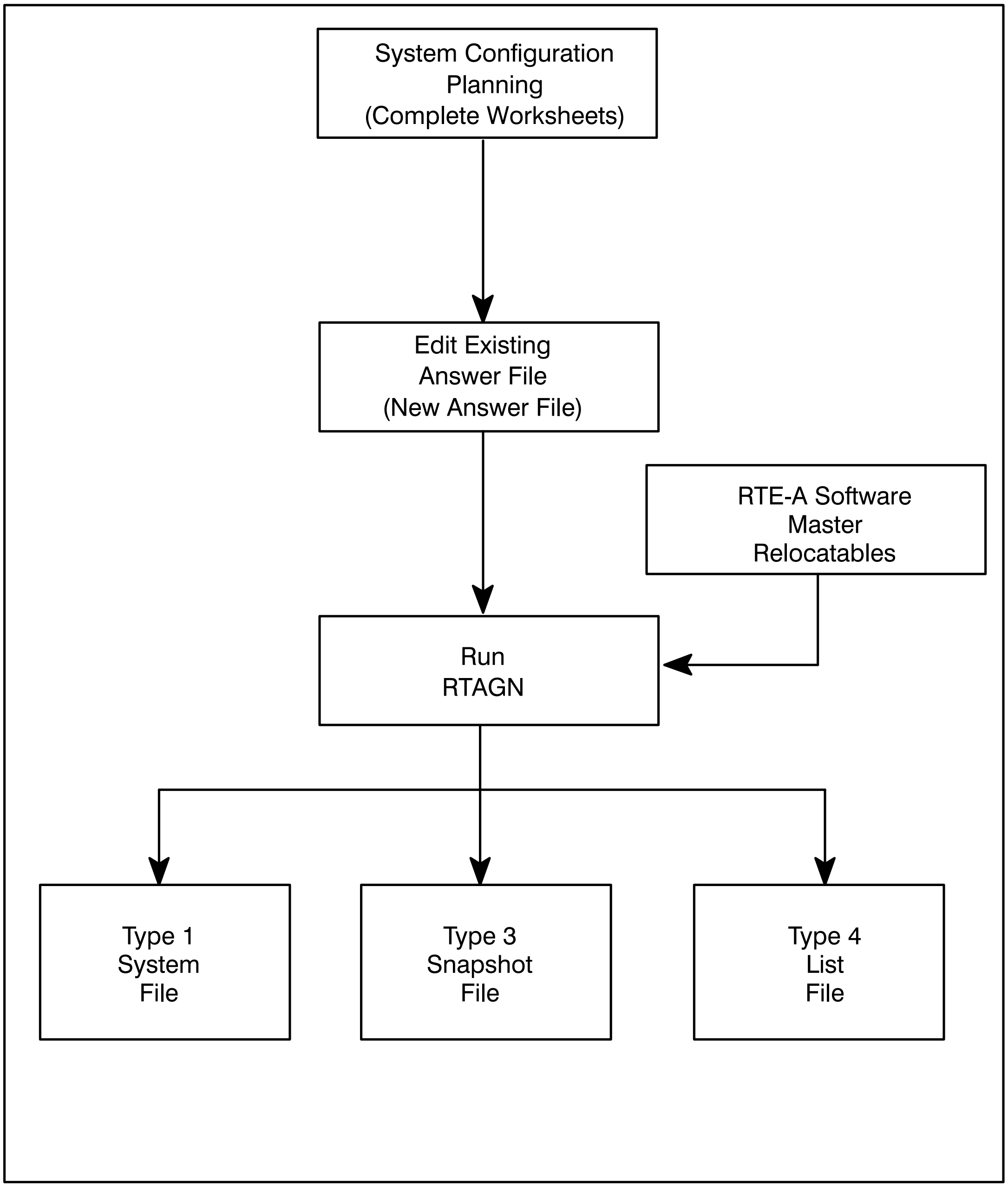


Figure 1-1. RTE-A System Generation Process

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 PRIMARY.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 because 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, because 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 disk-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 (described in Appendix I) 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 disk. A memory-based system that is downloaded over a LAN is described in Chapter 11. 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 the *DS/1000-IV Network Manager's Manual for RTE-A and RTE-6/VM,* part number 91750-90012.

A disk-based system refers to one that boots from disk 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.

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 disk, a cartridge tape drive (CTD), a magnetic tape, a DAT tape or a PROM module. You can also load a system from another computer over an HDLC link with DS or NS or in a local area network (LAN). 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 a manual boot can be a local terminal or a remote terminal. The remote terminal is connected to the local system through a direct DS link or a LAN. A direct DS 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. Booting over a LAN can be done from any computer system on the LAN; in other words, the system does *not* have to be a neighbor node.

Specific boot procedures are given in Chapters 9, 10, and 11. Appendix H contains detailed information on the boot parameters and several boot examples.

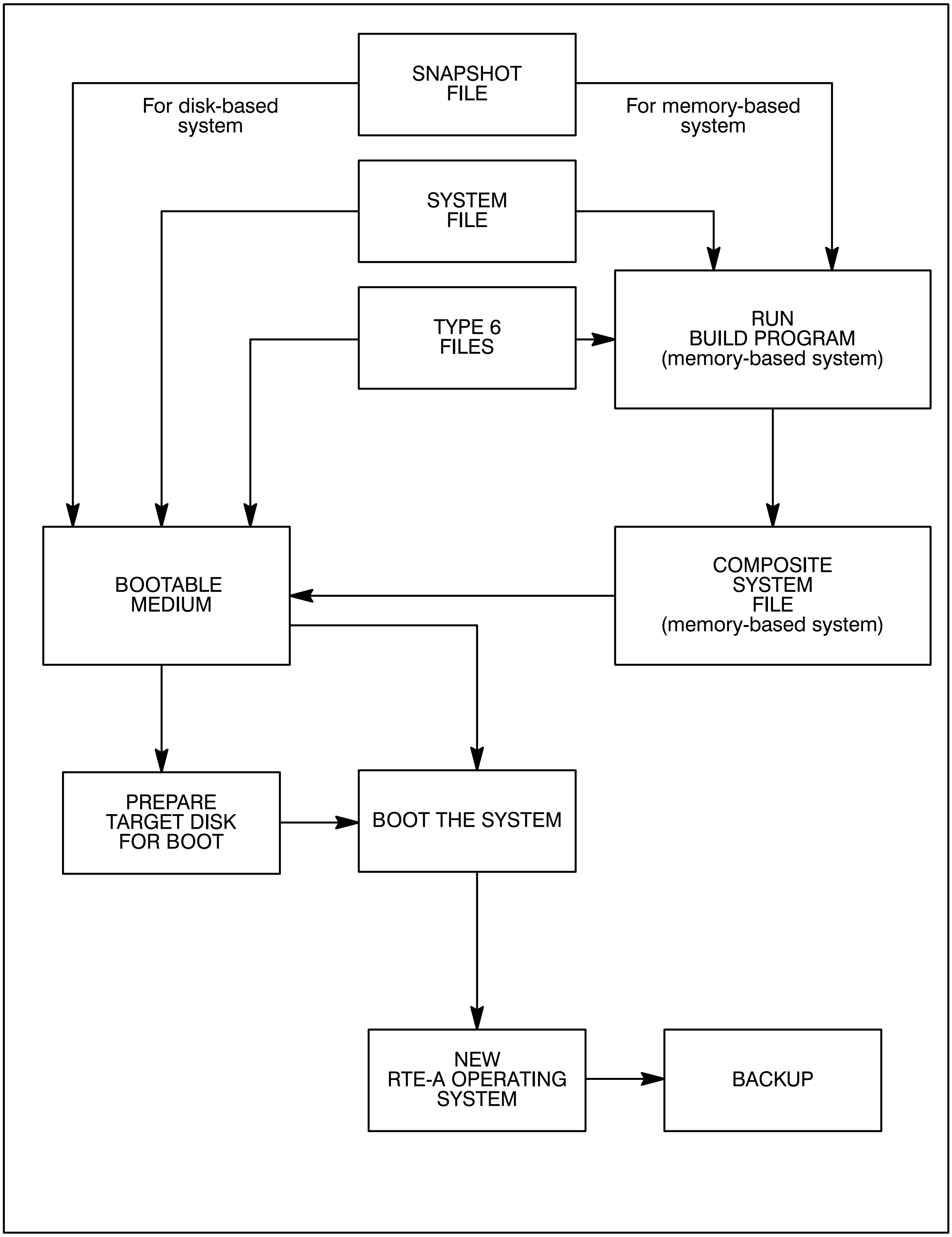

SNAPSHOT FILE
For disk-based system
For memory-based system
SYSTEM FILE
TYPE 6 FILES
RUN BUILD PROGRAM (memory-based system)
BOOTABLE MEDIUM
COMPOSITE SYSTEM FILE (memory-based system)
PREPARE TARGET DISK FOR BOOT
BOOT THE SYSTEM
NEW RTE-A OPERATING SYSTEM
BACKUP

Figure 1-2. RTE-A System Installation Process

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 lowercase italic 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 vertical bar (|) 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 subparameters. 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*, part number 92077-90002, 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

2

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. 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:

```
CI> 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:

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

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

Command File

A command file, also referred to as an answer 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 the list file descriptor, as opposed to the file name, starts with an apostrophe (’) or a caret (^) and if the specified file is in your current working directory or in a global directory, then the generator will overlay the file.

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 the system file descriptor, as opposed to the file name, starts with an apostrophe (') or a caret (^) and if the specified file is in your current working directory or in a global directory, then the generator will overlay the file. 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 generated 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, RTAGN ignores the additional characters when that entry point or module name is added to the snap file.

The snapshot file is required by LINK to load programs online. The snap file should be copied to /SYSTEM/SNAP.SNP because this is the default snap file name that LINK uses. Note that if an existing SNAP::0 file exists, it should be purged. Refer to the LINK SN command in the *RTE-A LINK User's Manual*, part number 92077-90035, for details on the default snap file name that LINK uses. If the default snapshot file name 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 descriptor, as opposed to the file name, starts with an apostrophe (') or a caret (^) and if the specified file is in your current working directory or in a global directory, then the generator will overlay the file. 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

3

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 *RTE-A System Design Manual*, part number 92077-90013. 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 the 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 *RTE-A System Design Manual* for the names of modules you can eliminate. The essential system modules are:

%ABORT
%EXEC
%IORQ
%MAPS
%PROGS
%SAM
%IOMOD
%UTIL
%RTIOA
%VCTR
%RPLxx
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)
```

where:

`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 (this 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.

4

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 *RTE-A System Design Manual* to determine which modules you can eliminate. The minimum system modules required are %ABORT, %EXEC, %IOMOD, %IORQ, %MAPS, %PROGS, %RTIOA, %UTIL, %SAM, %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; that is, 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.

SEARCH Command

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.

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[,`*address*`B]`

address`B` 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.

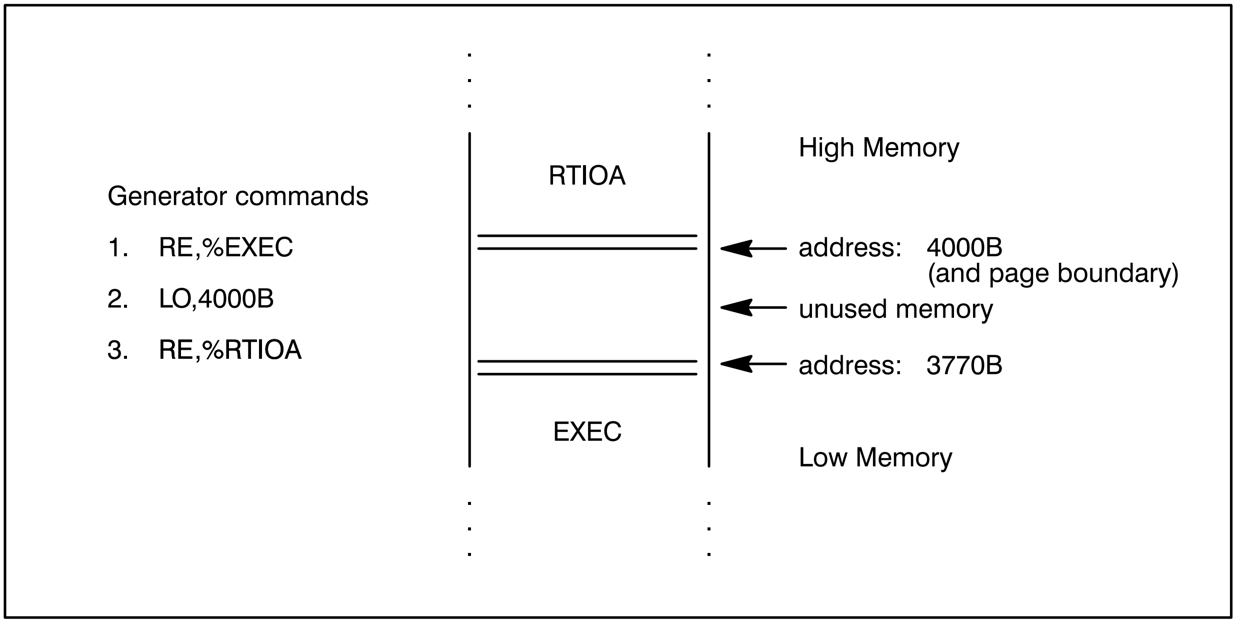


Figure 4-1. Results of Using the LOCC Command

BLOCC Command

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

Syntax: `BL[,`*address*`B]`

address`B` 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: 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	$IDRPL	RP'ing programs
22	0	ABORT	program abort processing
13	2	ALARM	timer signals
29	2	CDSFH	load code segment from disk (VC+)
34	2	CHECK	program security checks
91	2	CLASS	for class I/O
8	2	DSQ	required for DS/1000-IV, NS-ARPA/1000, or ARPA/1000
5	2	ENVRN	EXEC 39
30	0	IORQ	I/O requests
95	2	LOAD	EXEC 8, load and swap programs
80	2	LOCK	resource numbers, LU locks
85	2	MEMRY	EXEC 28, manage memory partitions
8	0	NSABP	network program abort processor for NS-ARPA/1000 and ARPA/1000
18	2	PERR	memory parity error recovery
18	2	SECOS	operating system security checks
53	2	SIGNL	signal processing
60	2	SPOOL	spool processing
354	2	SPSLG	performance monitoring
14	2	STAT	EXEC 26, PS and DS commands
85	2	TIME	time scheduling, time requests
42	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 to this value for each module in the above list (except $IDRPL, IORQ, NSABP, 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,1149
```

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 in 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 regenerate 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.

PARTITION (PA) 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. On the contrary, if you specify a module in the PA module list the dummy version will not be relocated out of $SYSA (even if you do not relocate the module in the OS partition phase.) This will result in undefined externals after the OS partition section. Thus, if you want the dummy version of a partitionable module be sure to remove the module from the PA module list and the OS partition relocation phase. (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.

5

OS Module/Driver Partition Phase

General Description

The RTE-A system is limited to 32 pages of logical memory. Because 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 non-partitionable 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 sizes of drivers used in the RTE-A system are given in Table 5-1. Table 5-2 gives the approximate sizes 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. Information on the tag area requirements for these modules can be found in the System Relocation Phase chapter.

Table 5-1. Partitionable Driver List

Driver	Description	Approximate Size (words)*
ADV00	Remote I/O Mapping for DS/1000 or NS-ARPA/1000	619
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)	1107
DD*30	Disk Device (ICD)	497
DD*33	Disk Device (CS/80)	1984
DD*60	X.25	818
DVN00	MRJE	1573
DDD63	RJE/MRJE	133
DDV63	RJE/MRJE	45
DDC00	Full Function Serial Driver	1094
DDC01	Full Function Serial Driver for Terminals with Attached Devices	1869
DDC12	2608S Line Printer Device (HP-IB)**	1565
DDM30	Disk Device (MAC)	1028
DDQ24	SCSI DAT Tape Driver	782
DDQ30	SCSI Disk Driver	638
DDX00	X.25	2694
DDX60	X.25	739

* Sizes are in decimal words and may vary slightly with each revision of the operating system.

** Also used for HP 256x with HP-IB.

NOTE: Some driver software is provided with the subsystem software/product. For example, DDX00 comes with X.25 and not with the RTE-A product.

Table 5-1. Partitionable Driver List (continued)

Driver	Description	Approximate Size (words)*
ID*00	Async Serial	473
ID*01	Asynchronous Serial with Modem	1003
ID*27	Integrated Disk Driver	891
ID*36	PROM Interface	136
ID*37	HP-IB Interface	1484
ID*41	Writable Control Store (WCS, A700)	190
ID*43	Powerfail Driver	294
ID*50	Parallel Interface (GPIO)	497
ID*52	Parallel Interface	471
ID*66	HDLC and bisync for NS-ARPA/1000 and DS/1000-IV	1458
ID*67	LAN	1605
ID100	ASIC Interface Driver for DDC00/DDC01	1035
ID101	ASIC I/F Driver for DDC00/DDC01 with Modems	1171
ID400	OBIO Interface Driver for DDC00/DDC01 (A400 only) with support for External Modems	1465
ID800	8-Channel MUX Interface Driver for DDC00/DDC01	1365
ID801	8-Channel MUX Interface Driver for DDC00/DDC01 with support for External Modems	1483
IDM00	Asynchronous MUX	1381
IDM37	HP-IB Driver for MAC Disks	1327
IDQ35	SCSI Disk Interface	1739
IDS00	Programmable Serial Interface (PSI)	1261
IDS64	DS/DL (Data Link) Slave	1368
DVS64	DS/DL (Data Link) Slave	1541
IDZ00	TELNET Pseudo Terminal Driver	931

* Sizes are in decimal words and may vary slightly with each revision of the operating system.

Table 5-2. Partitionable System Module Sizes

Module	Description	Approximate Size (words)*
$IDRPL	System ID duplicating and IDRPL	1131
Abort	Abort Processing	547
Alarm	Timer Signals	399
Cdsfh	(VC+ only) Load Code Segments From Disk	806
Check	(VC+ only) Security Checks for OS Calls	150
Class	Class I/O	1712
Dsq	Required for DS/1000	403
Envrn	(VC+ only) EXEC 39	711
IORQ	I/O request initiation	1173
Load	EXEC 8, Load and Swap Programs	1351
Lock	Resource numbers, LU locking	751
Memry	EXEC 28, Manage Memory Partitions	1990
NSABP	Network program abort processor for NS-ARPA/1000 and ARPA/1000	170
Perr	Memory Parity Error Recovery Routine	551
Secos	(VC+ only) Security Checks for USER calls	266
Signl	Signal Processing	1318
Spool	(VC+ only) Spool Processing	426
Spslg	Performance Monitor	2002
Stat	EXEC 26, PS and DS Commands	520
TIME	Time requests	1030
Vema	VMAIO Requests	569
Xcmnd	AS, BR, CD, DN, DT, GO, PR, SS, SZ, LU, VS, and WS Commands	988

* Size may vary slightly with each revision of the operating system.

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*, part number 92077-90019, 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.

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 (for example, RE, MS, and LOCC) are valid in the partition phase; however, system libraries (for example, $SYSLB and $SYSA) 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.

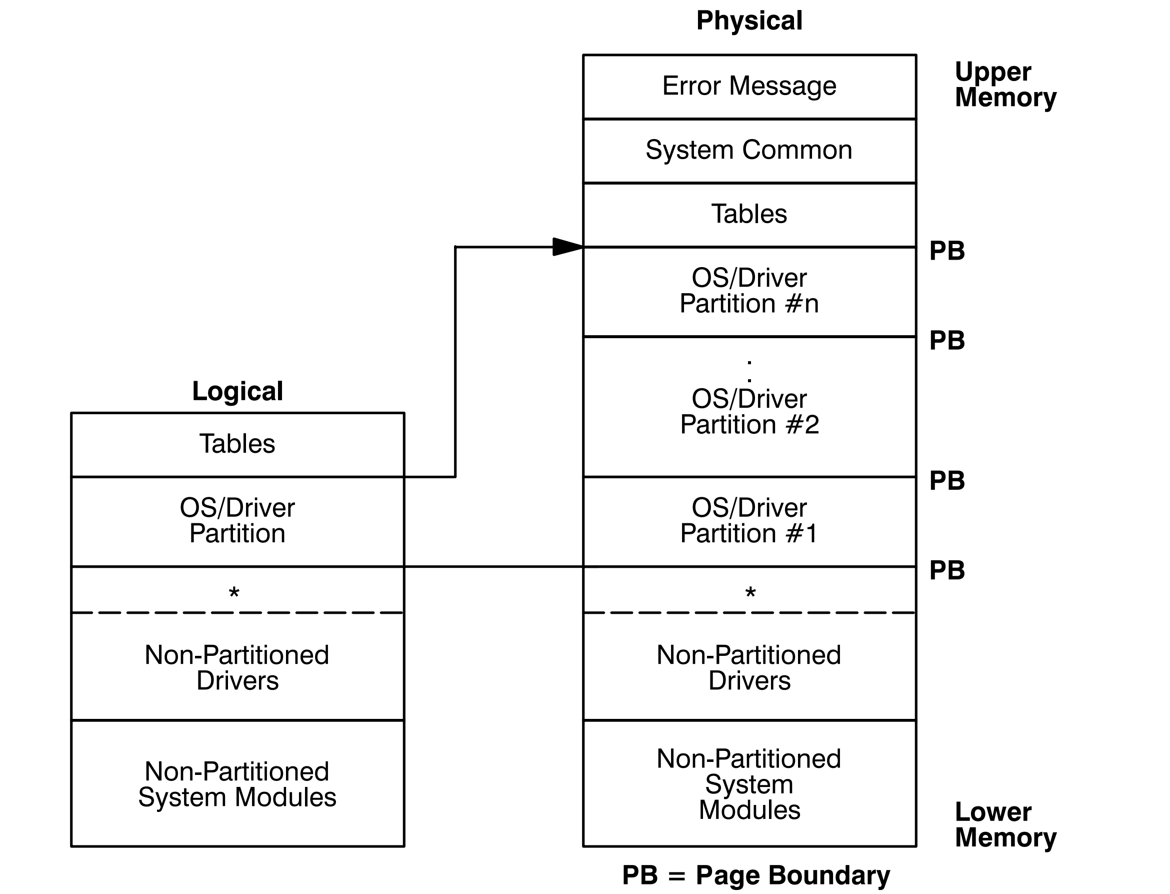
Syntax: `AL` or
`ALIGN`

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. There is only one base page for the whole system. Aligning modules/drivers usually reduces the number of base page links required to load the module. Because base page links can quickly become unavailable, it is advisable to align modules and drivers on page boundaries within a partition.



* The generator uses available space at the end of the non-partitioned relocated modules for table space that will fit in that area.

Figure 5-1. Sample Memory Maps

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.

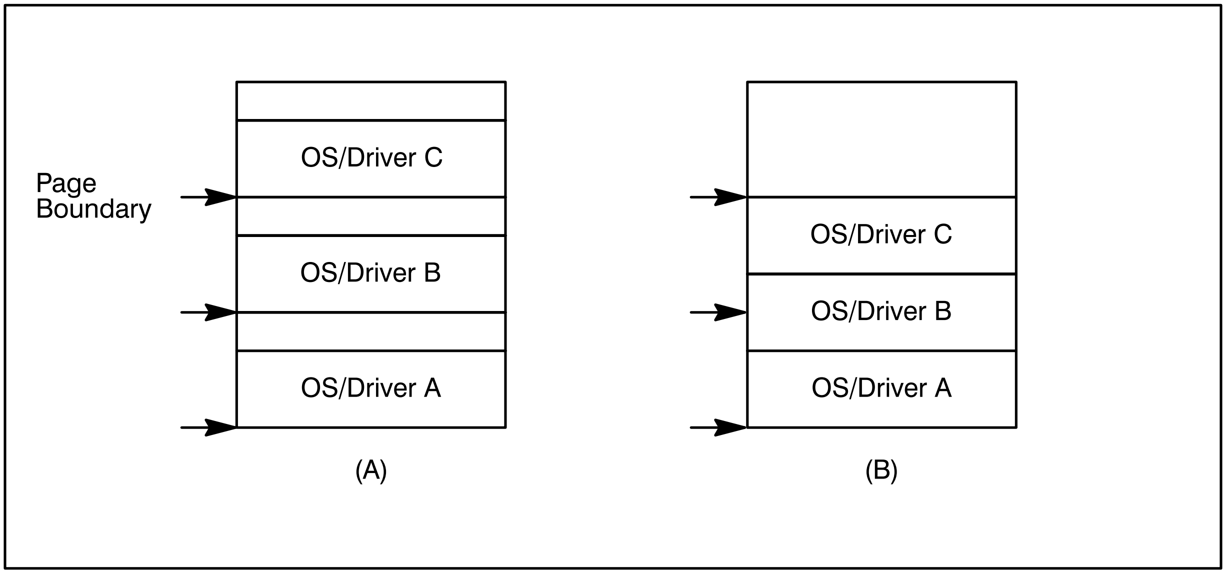


Figure 5-2. Sample Partitions

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
```

6

Table Generation Phase

This chapter provides the information needed for planning and preparing the entries for 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 disk 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 timeout 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 a CS/80 disk 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 LUs 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. Disk 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 three most common interfaces like this are the SCSI, HP-IB, and 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
```

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
```

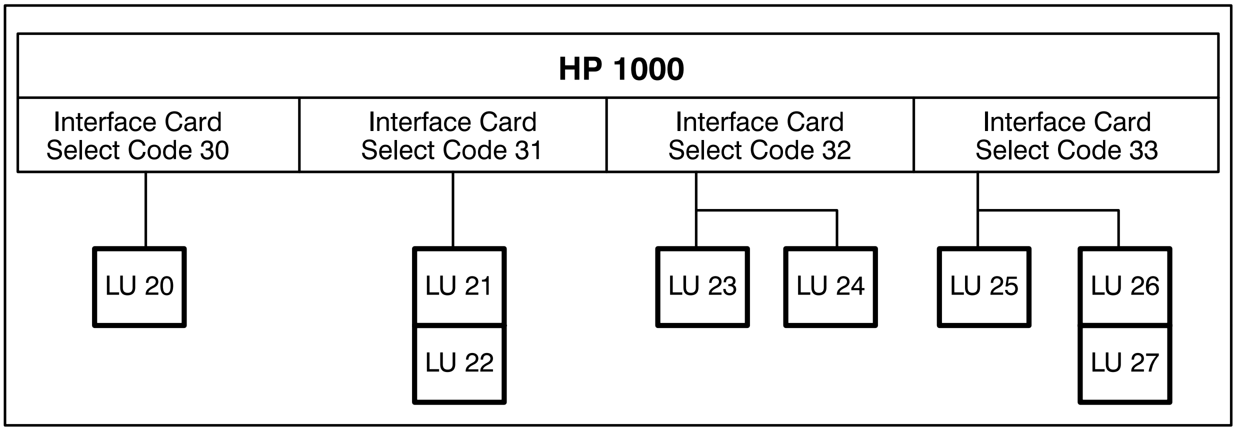

HP 1000
Interface Card Select Code 30
Interface Card Select Code 31
Interface Card Select Code 32
Interface Card Select Code 33
LU 20
LU 21
LU 22
LU 23
LU 24
LU 25
LU 26
LU 27

Figure 6-1. Example Node List Configuration

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 ensure that your LU numbers, select codes, Hewlett-Packard Interface Bus (HP-IB) addresses, and Small Computer System Interface (SCSI) addresses have been assigned correctly. An example 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 disk logical unit that you will include in your system, even though several disk LUs may be located on the same physical disk 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 Example 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/SCSI. Each device connected to an HP-IB/SCSI must have an address that is unique to that HP-IB/SCSI. Valid addresses are in the range of 0 to 7 and 36B, with 36B reserved for the HP-IB interface itself.

In the example worksheet shown on the next page, there is one HP-IB card with select code of 27B and one SCSI card with select code of 25B, respectively. Note that the HP-IB card has its own LU number. HP-IB address 36B sends control requests to the interface card. The HP-IB/SCSI addresses (0−7) serve to access individual devices connected to the bus.

Referring to the worksheet example, LUs 16, 17, and 24 all have the same HP-IB address (2) because they are all on the same device. Also, LUs 20 and 21 both have the same SCSI address.

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.

Example System I/O Configuration Worksheet

Device	**Interface**	**Logical Unit**	**Select Code**	**HP-IB/SCSI Address**
HP-IB	HP-IB	25	27B	36B
HP 7914	HP-IB	16	27B	2
HP 7914	HP-IB	17	27B	2
HP 7914 CTD	HP-IB	24	27B	2
SCSI disk	SCSI	20	25B	6
SCSI disk	SCSI	21	25B	6
Terminal	ASIC	1	20B	

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 LUs 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 example IFT/DVT worksheet shown at the end of this section has been completed for the HP-IB interface and the HP 7914 disk drives configured on the example I/O configuration worksheet. The origin of the required information for each entry is defined below.

IFT/DVT worksheet entry		Where to get information for that entry
INTERFACE NAME	–	I/O Configuration Worksheet.
I/O Slot #	–	Physical card cage slot number.
IFT	–	Table E-1, Device/Driver Correspondence: Interface Cards Category, HP 12009 HP-IB Interface entry.
SC	–	I/O Configuration Worksheet. The select code is set by switches on the interface card. If this is a pseudo driver, the select code is 0.
E	–	Table E-21, IFT Default File Values: HP 12009A Column.
QU	–	Table E-21.
TX	–	Table E-21: Table Extension entry blank, use the default value.
IT	–	Table E-21: DVT/IFT Device Type entry.
AL	–	LOcked. System disk 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 or SCSI Devices Category, HP 7914 entry.
Default File	–	This entry is generally the same as the device driver, or is a user-prepared file.
Model Number	–	Table E-19: Determined by disk subdivision. (Each model number must be entered in full.)
Logical Unit	–	I/O Configuration Worksheet.
Remaining DP Entries	–	Table E-19: Determined by disk subdivision.
Queuing:	–	FIFO (First In, First Out). This is the default.
NODE:	–	Disk LU numbers (Logical Unit entered above.)

Example IFT/DVT Worksheet for an HP-IB Card

Interface name: HP-IB
I/O slot number: 2

IFT, %ID*37, SC: 27B , E ID.37 , QU: PR , TX: ,
Interface Driver Name (Default file) Select Code Entry Point Queuing Table Extension

IT: 37B , AL: LO PS:
Interface Type Port Map Allocation Pseudo Driver

Device Name:	7914			
Device Driver:	DD*33			
Defaults File:	DD*33			
Model Number:	M7908_CF:0	M7908_CF:1	MTAPE	blank
Logical Unit:	LU: 16	LU: 17	LU: 24	LU: 25
Device Type:	DT: 33B	DT: 33B	DT: 26B	DT: 77B
Device Priority:	PR: 0	PR: 0	PR: 0	PR: 0
Timeout:	TO: 500	TO: 500	TO: 500	TO: 2000
Buffer Limits:	BL:	BL:	BL:	BL:
Table Extension:	TX: 72	TX: 72	TX: 72	TX: 0
Driver Extent:	DX: 8	DX: 8	DX: 8	DX: 1
Driver Parameters:				
start #	DP:1	DP:1	DP:1	DP:1
1	: 2	: 2	: 2	:36B
2	: 0	: 0	: 400B	: 0
3	: 0	: 0	:10000B	: 0
4	: 0	: 0	: 0	: 0
5	: 0	:19200	: 64464	: 0
start #	DP:6	DP:6	DP:6	DP:6
6	:400	:943	:	:
7	: 48	: 48	:	:
8	: 0	: 0	:	:
9	:	:	:	:
10	:	:	:	:
start #	DP:11	DP:11	DP:11	DP:11
11	:	:	:	:
12	:	:	:	:
13	:	:	:	:
14	:	:	:	:
15	:	:	:	:
Queuing:	QU: FI	QU: FI	QU: FI	QU: FI
Node 1: 16,17,24				
Node 2:				
Node 3:				
⋮				

Example IFT/DVT Worksheet for a SCSI Card

Interface name: SCSI
I/O slot number: 1

IFT, IDQ35 , SC: 25B , E IDQ35 , QU: PR , TX: 7
Interface Driver Name (Default file) — Select Code — Entry Point — Queuing — Table Extension

IT: 35B , AL: LO
Interface Type — Port Map Allocation

Device Name:	660S	
Device Driver:	DDQ30	
Defaults File:	DDQ30	
Model Number:	M64MB:0	M64MB:1
Logical Unit:	LU:20	LU:21
Device Type:	DT:30B	DT:30B
Device Priority:	PR: 0	PR:0
Timeout:	TO:2500	TO:2500
Buffer Limits:	BL:	BL:
Table Extension:	TX: 20	TX: 20
Driver Extent:	DX: 8	DX: 8
Driver Parameters:		
start #	DP:1	DP:1
1	: 6	: 6
2	: 0	: 0
3	: 0	: 0
4	: 0	: 0
5	: 6	: 4
start #	DP:6	DP:6
6	:4096	:4096
7	: 64	: 64
8	: 1	: 1
9	:	:
10	:	:
start #	DP:11	DP:11
11	:	:
12	:	:
13	:	:
14	:	:
15	:	:
Queuing:	QU:F1	QU:F1
Node 1:20,21		
⋮		

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 generation record file. The relocatable code for each driver or generation record file 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) is 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 are 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 are supplied from the default file. Transfer information in the format shown on the table generation worksheet. You need to enter as many IFT commands as you have IFT/DVT worksheets. For each IFT command you enter as many DVT commands as you have devices listed on the corresponding IFT/DVT worksheet. Refer to the example 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*[,E*entry 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.

E*entry point* specifies the entry point of the driver called to service requests 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      |   select code
          entry point
```

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
24B-27B
30B-33B
34B-37B
40B-43B
44B-47B
50B-53B
54B-57B
60B-63B
64B-67B
70B-73B
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 is 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 that 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 an example IFT entry and descriptions of the IFT parameters.

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

The above 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 does 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.

M*model* specifies the model number of the specific device described by this table, or specifies the size of the logical disk in megabytes and the position of a given logical disk. For example, the entry for an HP 7908 CS/80 disk can be specified as M7908_LF or M7908_CF. A model number specified with LF indicates long format and provides for longer disk 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 disk 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 disk, the differences are in format only) but the recommended format is CF.

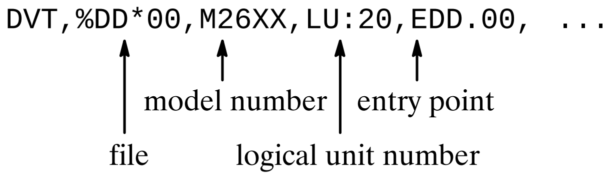
If a default file is specified, this parameter is usually required by the generator to select the parameter defaults for the specific device being configured. For example, in the case of the disk driver, DD*33, several different types of disks are supported. The correct track configuration for the given 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 disk 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 HP 7906 disk 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 disk 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.

E*entry 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,  ...
```

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 to which this device is connected.

TO:*to* specifies the decimal timeout value for the device. The value is a number that 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 timeout of 5.00 seconds.

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

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 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 are 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 (for example, with a disk) 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:1:A:B:C:D:E)
DP 2	=	B	
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:6:F:G:H:I:J)
DP 9	=	I	
DP 10	=	J	

If certain parameters must be replaced (that is, 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 *RTE-A Driver Reference Manual*, part number 92077-90011, for details. In the case of disk drives, these parameters are used to define the track map for each drive. Refer to Appendix C on disk configuration in this manual for more information on how to configure the disks 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 an HP 7906 disk drive, an HP 7908 disk drive with CTD, a line printer, and a digital voltmeter assigned the following LU numbers:

Device	LU Number
7906 disk LU	30
7906 disk LU	33
7906 disk LU	43
7906 disk LU	45
7908 CS/80 disk LU	10
7908 CS/80 disk LU	16
7908 CS/80 disk LU	17
7908 CS/80 disk 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.

7

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 disk 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 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. 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. If environment variables will be used in the system, allocate one resource number for each session user on the 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 online program loading.

A maximum of 254 ID segments, each requiring 45 words of memory, can be allocated. One ID segment is needed for each program that will be active concurrently with other programs. The same number of ID segment extensions will automatically be claimed from the beginning of XSAM.

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
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

[4*(number of ID segments)]	for systems with CDS programs
or	
[2*(number of ID segments)]	for systems without CDS programs

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 (for example, D.RTR). It is also useful to place frequently used NS-ARPA/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 * (# 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 *RTE-A 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 ID segment extensions, User-Defined Search Path Directories (UDSPs), prototype IDs, 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, in addition to the number of words used for ID segment extensions, is sufficient for most applications.

Specifying Class Buffer Limits

Buffer limits for class requests are specified with the CL command. (The CL command supercedes the SL command.) If the upper class limit does not leave at least 1000 words of SAM remaining, the generator issues a warning. The command

```
CL,1000,2000
```

can be used to set the lower and upper limits for class requests. With this command, the class system will suspend a program when more than 2000 words of SAM are being used by buffers allocated for that class number. This suspension allows the buffers to be used by other programs. Class requests will restart when less than 1000 words of SAM are being used by the allocated buffers.

The default lower limit is 1/16 of SAM and the upper limit is 1/8 of SAM. To specify the default values, either enter CL without parameters or enter "`CL,,,`".

If you do not have class I/O in your system, you may enter:

```
CL,0,0
```

If you do have class I/O, specifying the CL command with zeros does not give the default values.

Note that CI uses class I/O, so do not specify a limit of zero if your system has CI.

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 VC+, 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-ARPA/1000 or DS/1000-IV system for tables; refer to the NS-ARPA/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-ARPA/1000 or DS/1000-IV, 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 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 that 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 VC+ 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 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 $FDSLB 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-ARPA/1000, DS/1000-IV, IMAGE, and Graphics. 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 online 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	changes the default allocation of memory descriptors
SAM	allocates System Available Memory, Extended SAM
CL	specifies class 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 online
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 degraded 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. If environment variables will be used in the system, allocate one resource number for each session user on the system.

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-ARPA/1000 or DS/1000-IV 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 online with the CI OF, RP, and RU commands. However, additional ID segments cannot be created online.

An identical number of ID segment extensions will be claimed from the beginning of XSAM. Each extension is 5 (five) words in length.

Memory Descriptor Command RS

Purpose: Changes the default allocation of memory descriptors.

Syntax: `RS,`*n*

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

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

4*(number of ID segments) (CDS)
or
2*(number of ID segments) (non-CDS)

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

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.

Each ID segment defined for the system (via the ID command) requires five words of XSAM.

If the DS/1000-IV, NS-ARPA/1000, or ARPA/1000 networking product is in the system, more SAM may be required. Refer to the respective network generation manuals for more information on SAM and XSAM usage.

Class Limits Command CL

Purpose: Specifies class buffer limits.

Note: the CL and SL commands have the same functionality.

Syntax: `CL[,`*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 class buffers in the system. When the upper limit is reached on a class, further requests on that class causes the requester to be put in class-limit suspend state. When the program doing Class Gets on a class that is "in the limit" removes enough buffers to lower the number of words (charged to the class) below the lower limit, any programs in class-limit suspend on the class will be resumed. 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 block is used by the networking products, DS/1000-IV and NS-ARPA/1000. Refer to the DS/1000-IV or NS-ARPA/1000 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 that 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 then begins 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, 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 (VC+ only).

Syntax:

```
   re,%MSGTB                       message table
   end
[  re,%$M000  ]                    English language catalog
[  end        ]
[  re,other language catalog ]     if appropriate
[  end                       ]
[  re,security.rel ]               VC+ 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 are required for system and user application commands. Refer to the *RTE-A System Manager's Manual*, part number 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 online. 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 the system common relocation. Use an END statement to terminate the set. For VC+ systems, a second set of libraries may be specified following the non-CDS library specification. It too must be terminated with an END statement.

When linking programs, LINK searches these files in the order specified here to satisfy external references. For non-CDS programs, LINK searches only the first set of libaries; the second set is searched only for CDS programs. You may add any number of your own libraries to each list.

For many libraries, a non-CDS and a CDS version are supplied. Because non-CDS modules may not call CDS modules, it is important to order the list such that the CDS version is searched before the non-CDS version.

Note that only the names of these files are placed in the system snapshot file.

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,$FNDLB
LIB,$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,$FNDLB
LIB,$BIGLB
```

For VC+ systems, the suggested library sequence is

```
* System libraries

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

* System CDS Libraries

LIB,SEC1000CDS.LIB
LIB,$BGCDS
LIB,$FNDLB
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 *RTE-A LINK User's Manual*, part number 92077-90035, for more information.

8

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 disk-based systems.

A disk-based system as considered in this manual is a system that boots from disk 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 remotely over a LAN or over an HDLC link (with NS-ARPA/1000 or DS/1000-IV). 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 that 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 disk-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 disk-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.

Loading RTE-A Programs and Subsystems

The RTE_INSTALL.CMD file can be used to link RTE-A and a number of other RTE-A subsystems and products. When used with a CDS version of CI, RTE_INSTALL can be used to install the current revision of the following products and subsystems:

RTE-A
LANVCP
HP C/1000
IMAGE/1000-II
VC+
NS-ARPA/1000
FTN7X
FORMS/1000
MAIL/1000
ARPA/1000
DEBUG/1000
LP
PASCAL/1000
XDB

The LP and LANVCP subsystems are part of the VC+ product and XDB is included with the DEBUG/1000 product.

Note When using a non-CDS version of CI, RTE_INSTALL only installs RTE-A, FTN7X, and DEBUG/1000. (The non-CDS version of CI does not have enough free memory for all of the installation variables.)

Caution RTE_INSTALL will only work with the current revision of all of these products. This command file will not install previous revisions of any of these products correctly.

RTE_INSTALL calls the RTEA1.CMD, RTEA2.CMD, VC1.CMD, and VC2.CMD files. The RTEA1.CMD and RTEA2.CMD files create type 6 files for all programs provided with RTE-A, for example when you are updating your system software. These command files perform other tasks such as updating certain directories, creating libraries, and indexing libraries.

The installation command files use several CI variables to determine the particular tasks that will be performed. A description of each variable is as follows:

$RTE_RETURN This variable controls the LINK error handling in this command file. This may be set as follows:
A - Abort this transfer file if an error is encountered when linking a program.
S - Suspend the installation process and allow the user to correct the problem before proceeding.
C - Continue execution of this transfer file and only report link errors. This is the default.

$RTE_NFS If set to T, the version of $BIGLB required for the HP 1000 file server will be installed. (The HP 1000 file server ships with NS-ARPA/1000 and ARPA/1000.) This version of $BIGLB supports type 12 byte stream files. Setting this option causes $RTE_SLINK and $RTE_CDS to be set to T. The default is to install the version of $BIGLB that does not support type 12 files.

$RTE_CDS If set to T, RTEA1.CMD and RTEA2.CMD will not load the non-CDS versions of programs that are loaded as CDS by VC1.CMD and VC2.CMD, thereby saving time. The non-CDS programs that are not loaded are as follows:

CI CIX D.RTR DL LS GREP CMNDO

The default is to load the non-CDS versions of these utilities.

$RTE_SLINK If set to T, the target system will have the capability to create and access symbolic links in the file system. The versions of $BIGLB and $BGCDS that can access symbolic links will be installed and the LNS utility will be loaded. The default is to install $BIGLB and $BGCDS versions that cannot access symbolic links and to not load LNS.

$RTE_MKLINKS Set to T when symbolic links can be used when creating some of the target programs. The default is to make a copy of a file instead of making a symbolic link.

$RTE_HPMDM If set to T, the HPMDM modem controller utility is loaded by RTEA2.CMD. This utility requires file HPMDM_TABLE.REL to be relocated into the system common area of the target system. The default is to not load this utility.

$RTE_A990 If set to T, the following programs that are used only on A990 processors are loaded: CLOCK, SETVCPSTRING, DOWNLOAD, and A990FWID. This variable is used only by RTEA2.CMD. The default is to not load the A990 programs.

$RTE_LIBS May be set to the directory name where libraries are to be copied. The default is /LIBRARIES. Note that this does not cause LINK to search this directory when loading programs; it only allows an alternate directory structure to be used for the target system.

$RTE_CATS May be set to the directory name where NLS catalogs are to be copied. The default is /CATALOGS.

$RTE_HELP May be set to the directory name where help files are to be copied. The default is /HELP.

$RTE_SYS May be set to the directory name of the target /SYSTEM directory. The default is /SYSTEM.

$RTE_SCR May be set to a scratch directory to be used during the installation process. The default is /SCRATCH.

$RTE_MAPS May be set to a directory to be used to save the link map files. The default is to not save the map files.

Note The LINK NLS catalog file must be present in the /CATALOGS directory for LINK to run. If $RTE_CATS is set to another directory, you must copy file LINK.C000 from /RTE_A to /CATALOGS prior to linking up the programs.

Using RTE_INSTALL.CMD

The command string to run the RTE_INSTALL.CMD command file is as follows:

```
CI> RTE_INSTALL snap, [RTE-A s/w dir], [Q], [I]
```

where:

snap is the full path for the target system snap file. This cannot be a relative path.

RTE-A s/w dir is the directory containing the RTE-A relocatables. The default is /RTE_A.

`Q` indicates quiet mode. Do not ask for verification after displaying the list of products to be installed.

`I` inhibits product defaults. Do not install a product whose product directory variable is undefined. (By default RTE_INSTALL tries to install every product that is not defined.)

The return values for $RETURN1 are as follows:

-1 A fatal error was reported by RTE_INSTALL before the product installation phase.

0 All products were successfully installed.

1 No fatal errors were reported, but, some LINK errors were reported. (When this occurs RTE_INSTALL displays a list of the programs that could not be successfully linked.)

2 A fatal error was reported during the installation of a product and RTE_INSTALL terminated.

Product Installation Defaults

The product installation is controlled by CI variables. To change the defaults, you can set the CI variable to a different value prior to invoking RTE_INSTALL or you may edit this command file and change the default. (The I runstring parameter can be used to inhibit all product defaults.)

Product	**Variable**	**Default Setting** (directory)
RTE-A	$RTE_A	/RTE_A
VC+	$VCPLUS	/VCPLUS
NS-ARPA/1000	$NS1000	/NS1000
ARPA/1000	$ARPA	/ARPA1000
LP	$LP	/VCPLUS/LP
MAIL/1000	$MAIL	/RTE_A/MAIL
LANVCP	$LANVCP	/VCPLUS/LANVCP
FTN7X	$FTN7X	/FTN7X
PASCAL/1000	$PASCAL	/PASCAL
HP C/1000	$C1000	/C1000
DEBUG/1000	$DEBUG	/DEBUG
XDB	$XDB	/DEBUG
IMAGE/1000-II	$IMAGE2	/IMAGE2
FORMS/1000	$FORMS	/F1000

Setting a directory path for a product causes RTE_INSTALL to try to install the product from the given directory. If the directory cannot be found, an error message will be reported and the installation will continue, skipping the product.

By default, RTE_INSTALL tries to install every product that it finds. To disable the installation of a product, set the variable to a blank string before using this command file or use the I runstring parameter to inhibit the product defaults.

For example, to disable the installation of LANVCP:

```
CI> set lanvcp =
```

After RTE_INSTALL successfully installs a product, the associated product directory variable is set to a blank. This allows RTE_INSTALL to skip products that it has already installed if the command file is re-run from the same session.

Target Directory Defaults

The target directories are defined by CI variables. To override a default, you can set the CI variable to a different value prior to invoking RTE_INSTALL. The target directory defaults are generated by concatenating $RTE_DIR with the directory name.

Variable	**Default Setting** (directory)
$RTE_DIR	/TARGET/
$RTE_CATS	/TARGET/CATALOGS

$RTE_ETC	/TARGET/ETC
$RTE_HELP	/TARGET/HELP
$RTE_INC	/TARGET/INCLUDE
$RTE_LIBS	/LIBRARIES
$RTE_MAPS	<none> (link map files are not saved)
$RTE_PROGS	/TARGET/PROGRAMS
$RTE_SCR	/SCRATCH
$RTE_SYS	/TARGET/SYSTEM
$RTE_USR	/TARGET/USR
$RTE_USERS	/TARGET/USERS

Note that these defaults assume that the disk on which /TARGET resides has a large amount of free space. By altering the $RTE_DIR variable, you can change all of the defaults to be global directories instead of subdirectories.

For example, to update the current system's global directories, you can set $RTE_DIR to /. This will cause the target catalogs directory to be /CATALOGS. Note that $RTE_PROGS cannot be set to /PROGRAMS if you are updating RTE-A or VC+.

Note If you are updating your current system's global directories, you should shut down any subsystems that are currently running before installing the new version.

Note that $RTE_DIR has no effect on target directory variables that have already been set. You must either use a new copy of CI or unset all of the target directory variables.

RTE Variable Defaults

The RTE variable defaults are as follows:

Variable	Default setting	Action
$RTE_CDS	T	Target system will use CDS utilities. (T|F)
$RTE_NFS	F	Target system will not support the HP 1000 file server shipped with NS-ARPA/1000 and ARPA/1000. (T|F)
$RTE_SLINK	T	Target system supports symbolic links. (T|F)
$RTE_RETURN	C	Continue after link errors. (A|C|S) A - Abort the installation if an error is encountered when linking a program. S - Suspend the installation process and allow the user to correct the problem before proceeding. C - Continue the installation and only report link errors.
$RTE_A990	*	Link the A990 programs. (T|F)
$RTE_HPMDM	*	Link the HPMDM program. (T|F)
$RTE_MKLNKS	*	Create symbolic links on the target system. (T|F)

* RTE_INSTALL determines the defaults for these variables at runtime. $RTE_A990 is set to T if the target snap file contains the A990 RPLS. $RTE_HPMDM is set to T if the target system was

generated with the $HPMDM_TABLE common block. $RTE_MKLNKS is set to T if $RTE_SLINK is set to T and if the host system supports symbolic links.

Setting the variable $RTE_NFS to T causes versions of $BIGLB.LIB and $BGCDS.LIB to be created that support type 12 byte stream files. It also causes the FSRV monitor to be installed when NS-ARPA/1000 or ARPA/1000 is installed. Type 12 file support is required for any system that uses the HP 1000 file server shipped with NS-ARPA/1000 and ARPA/1000 or on any system that performs remote access to files created by the HP 1000 file server. Due to the FMP code growth associated with the type 12 file support, $RTE_CDS will also be set to T when $RTE_NFS is enabled. See the *File Server Reference Guide for NS-ARPA/1000 and ARPA/1000,* part number 91790-90054, and the *RTE-A Programmer's Reference Manual*, part number 92077-90007 for more information.

NS-ARPA/1000 Variable Defaults

The NS-ARPA/1000 variable defaults are as follows:

Variable	**Default Setting**	**Action**
$NS_DS3K	N	No DS/3000 access. (Y|N)
$NS_X25	N	No X.25 connection to DS/3000. (Y|N)

LP Spooler Variable Defaults

The LP Spooler variable defaults are as follows:

Variable	**Default Setting**	**Action**
$LP_UPDATE	T	updates the target spool/help directories. (T|F)
$LP_LOADPROGS	T	causes the LP spooler programs to be linked. (T|F)
$LP_REMOTE	T	installs incoming/outgoing remote printer support. (T|F)
$LP_LOCAL	F	does not shutdown/restart the local LP spooler. (T|F)

Mail/1000 Variable Defaults

The Mail/1000 variable defaults are as follows:

Variable	**Default Setting**	**Action**
$MAIL_START	N	does not start Mail. (Y|N)
$MAIL_SMTP	Y	installs the SMTP software. (Y|N|A(sk))
$MAIL_DNS	N	does not install the DNS software. (Y|N|A(sk))

Pascal/1000 Variable Defaults

The $RTE_CDS variable is used to determine if the CDS versions of the compiler should be used. By default, the compilers are linked such that the default code generation is set to CDSOF. The default working set for PASCOMP.RUN is set to 250 pages.

Image/1000-II Variable Defaults

Remote data base support is enabled if the target system contains the RD.TB common block.

Examples:

The following example builds a target system that supports the HP 1000 file server. The target system does not require the LANVCP or the MAIL/1000 subsystems.

```
***************************************************************
* Build versions $BIGLB.LIB and $BGCDS.LIB that support type 12
* byte-stream files.  This is required if the HP 1000 file server,
* FSRV, is going to be used.
*
set rte_nfs = T
*
* Abort the installation if any LINK errors are encountered.
*
set rte_return = A
*
* Create the /target directory on a disk that has enough free
* space for all of the products being installed.  (It requires
* approximately 100,000 free blocks to install all of the products
* and keep the link map files.)
*
crdir /target 50
*
* Save the link map files in /target/maps/
*
set rte_maps = /target/maps
*
* Skip the LANVCP and MAIL/1000 subsystems.
*
set lanvcp =
set mail =
/rte_a/rte_install /system/newsys.snp
*********************************************************
```

The following example updates PASCAL and DEBUG on a system that has already been updated to the current revision.

```
*****************************************************************
* Suspend the installation if any LINK errors are encountered.
*
set rte_return = S
*
* Set the target directories to the current system’s global
* directories.  (The type 6 files will be placed in /programs.)
```

```
*
set rte_dir = /
*
* Only install PASCAL and DEBUG.
*
set pascal = /pascal
set debug = /debug
/rte_a/rte_install  /system/snap.snp,/rte_a,Q,I
*****************************************************************
```

The following example installs RTE-A and VC+ and updates the current system’s global directories. The target programs directory is /NEWPROGS.

```
****************************************************************
* Set the target programs directory to /newprograms and set the
* rest of the target directories to the current system’s global
* directories.
*
set rte_progs = /newprogs
crdir /newprogs 17
set rte_dir = /
*
* Disable symbolic links on the target system.
*
set rte_slink = F
*
* Only install RTE-A and VCPLUS
*
set rte_a = /rev6200/rte_a
set vcplus = /rev6200/vcplus
/rte_a/rte_install  /system/newsys.snp,$rte_a,Q,I
****************************************************************
```

The Boot Process

A system can be booted from a disk, a Cartridge Tape Drive (CTD), a magnetic tape drive, a DAT tape drive, or a PROM module. You can also load a system from another computer over an HDLC link (with NS-ARPA/1000 or DS/1000-IV) or over a local area network (LAN). 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 disk) 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 Disk

When boot is from a disk, the VCP/loader ROM loads into memory and executes the “file” specified by a file 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 bootstring 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 a FMGR cartridge. If unable to find the file specified, BOOTEX switches to interactive mode and prompts for boot instructions.

Boot From Cartridge, Magnetic, or DAT Tape Drive

When boot is from a cartridge tape drive (CTD), or from a magnetic or DAT tape 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 a DS Link

When boot is over a DS link (HDLC link with DS/1000-IV or NS-ARPA/1000), 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 the boot file downloader 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 disk at the neighbor node. It can be at any node on the DS or NS-ARPA network, and the neighbor node can use the store-and-forward version of PROGL (loaded from %PROGZ) to boot the operating system into the destination computer.

Also, the destination computer must be connected by an HDLC link to the neighbor that is running PROGL. The VCP loader in the destination system can respond to the PROGL queries over the HDLC network link communications line.

When a file number and the destination computer select code of the HDLC 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. 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 disk boot (created by BOOTEX or BUILD) or an absolute binary file created from such a system file by binary to absolute system utility MI2AB.

Boot From a LAN Link

When boot is over a LAN link, the VCP loader ROM code at the destination system interacts with the download monitor program, VCPMT. VCPMT runs on the local or server node on a LAN. The user interface to VCPMT is RMVCP, remote VCP. These two programs are described in Chapter 11.

The operating system being booted is a memory-based system and is called a "download file." It is a file on the local or server node.

The destination computer or client node is any node on the LAN. The destination computer's LAN card must be set for remote VCP and, if desired, for Forced Cold Load operation. Upon boot-up, VCPMT sends the memory-based system download file to the remote node.

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 disk with CTD (HP 7912/14/42/46) you can install your system from the CTD tape to disk. This involves placing the tape cartridge in the CTD and performing the offline restore procedure. The *RTE-A Backup and Disk Formatting Utilities Reference Manual*, part number 92077-90249, 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 disk without cartridge tape (HP 7933/35/41/45), you can install your system from magnetic tape to disk. 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 for creating an offline (memory-based) version of ARSTR is described in the *RTE-A Backup and Disk Formatting Utilities Reference 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 disk configuration, you can keep them all on the same bootable disk 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 DAT tapes. You can then place a DAT tape on a DAT tape drive and boot each system.

Multiple memory-based systems become even easier to install with HDLC (with DS/1000-IV or NS-ARPA/1000) or over LAN. 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 connected on a LAN or 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.

Disk-based destination systems can also be booted over HDLC (with DS/1000-IV or NS-ARPA/1000) or over LAN. For example, if you have several disk-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.

9

Disk-Based System Installation Procedure

This chapter provides the installation procedure for a disk-based system. It is assumed that your system is to be booted from a disk drive, although it is possible to have a disk-based system that does not boot from a local disk (from a PROM module, HDLC link with DS/1000-IV or NS-ARPA/1000, or LAN).

For a disk-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 installed.

A summary of the steps in a disk-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 disk-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 disk LUs, defining partitions, setting up the swap area, and re-linking 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/1000-IV or NS-ARPA/1000 node, commands are not echoed.

Mirrored Disk 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 file descriptor 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 number 92050-90001.

Specifying the System

The following four commands specify the system file that is 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 a 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 creates or overwrites the SY file before booting. This prevents 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 disk 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 when 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 Disk Volumes

In addition to the system disk LU, BOOTEX can mount disk LUs you want to appear in the cartridge list of the target system when you boot it. Use the MC command to mount an LU:

`MC,`*lu* — mounts the disk LU specified by the *lu* parameter. This LU is defined in the target system. The *lu* parameter may be either positive or negative. Note that you are specifying an LU, not a FMGR CRN.

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

BOOTEX limits the number of physical disks that can be mounted. You may use the MC command to mount up to an additional three HP-IB disks, one SCSI disk, and one Micro/1000 integrated disk using the HP 12022A controller. Additional disks must be mounted in the welcome file.

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 disk blocks, in the filedescriptor portion of the command string:

`SW,`*filedescriptor*

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 purges the existing file, and then recreates this file by using the new file size specified in the filedescriptor string. If the specified file size is greater than the amount of space available on the disk, BOOTEX decreases the size and issues a message indicating the change.

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 * 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 issues a warning, but creates the swap file, if you specify a file size that is smaller than the default size. The minimum swap file size is 3 blocks. The maximum swap file size is 4096K words.

Swap file size should be chosen with care. The swap file should be large enough to handle all system needs; system functions are 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 disk space and does not improve 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:

(total required pages − available pages) * 8 = swap file size

Specifying the Scratch LU

You may specify that the VMA backing store scratch files are to be put on a FMGR cartridge with the SC command:

`SC[,`*lu*`]`

If an LU is specified, it must be a FMGR cartridge LU. That disk 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). You may specify the VMA backing store scratch files are to be put into the /SCRATCH directory with the following command:

```
SC,0
```

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 as follows:

```
AC[,ON|OFF|OA]
```

OA (only accounting) turns accounting on, however, when a user logs off, end-of-session information is not displayed at the user’s terminal.

If the AC command is specified without one of the above options, 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. Note that session accounting must be enabled for METER to function properly.

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 executes for the specified number of milliseconds and then suspends 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 *RTE-A 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 to 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 runs. 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 is also 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 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 specified programs are assigned the partition. Usually, there is only is one per AS command. The partition is big enough to hold the biggest program named. If progname is a CDS program, the command

AS,*progname*[:D]

assigns its data segment to a reserved partition. The :D entry is optional, because 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.

Note 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 are used.

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

If an RPd program uses shareable EMA and specifies that the shareable EMA area reside in a particular reserved partition, the size of the EMA area is 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 appears:

```
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.

Example 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, you should RP this

```
RP,CI,CM                program here.
RP,CI                   restore CI
ST,,1                   assign CI as startup program, using welcome file 1
EN                      end of RP phase
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 disk-based system, this medium is a disk volume.

Before performing any of the disk 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 disk surfaces. For CS/80 disks, there is only one such LU; HP 7906 removable platter disks have several. The correct LU is probably 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 disk volume. For FMGR cartridges, an empty BOOTEX file is created by the FMGR IN command, described in the *RTE-A User's Manual,* part number 92077-90002.

For other disk 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 disk volume (user input is underlined):

```
FMGR : CI
CI> in 13
Re-initialize valid directory [N]? y
Initializing disk
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 disk 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 physical backup of the CS/80 disk. This allows recovery in the event of even the most catastrophic errors. A logical or FST 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 a CS/80 disk, and LU 10 is the first LU on a SCSI disk. It touches cylinder 0, which means it is possible to boot from this disk LU. To prepare the disk for boot, it must be initialized so that the boot extension (BOOTEX) can be placed beginning at cylinder 0 of the disk. BOOTEX requires 768 blocks, so this area must be made available at the beginning of the disk LU. This is done with the CI IN command illustrated in the procedure given in the next section, “FPUT and BOOTEX for New Disk 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 disk LU; it also has an option to install a consoleless system. Refer to the *RTE-A User’s 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 disk volume does not work because 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 Disk Configurations

Often a new disk 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 disk LU that will boot the new target system is more involved than just installing the current system on another disk LU.

It is important that, before initializing any disks, the current (host) system should be adjusted so that its “system” disk is not the first disk LU, for example LU 16. This is because LU 16 needs 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 has 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 disk, then it already has been initialized with the boot extension in the 768-block area of LU 16.)

Assuming that the new disk 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 physical backup of a CS/80 disk and a TF or FST backup of the entire disk 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 disk 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 (user input is underlined):

```
CI> in 16 768
Re-initialize valid directory [N]? yes
Initializing Disk
CI>
```

Next, make the target snap, system, and boot extension files available on a disk 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 has no way of knowing about the new LU on the target system, because that LU is not a disk LU on the host system. In this case, LU 16 describes the same area on both systems, so there is no problem. If there is a discrepancy in the physical disk areas described by LU 16 on the two different systems, or if you will be booting your system file from an LU other than LU 16, run INSTL and create a BOOTEX file on directory NEWSYS (user input is underlined):

```
CI> wd /newsys
CI> instl
Enter snap file, system file, destination file, lu, source file,
[N] (no console option)
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[,offset0][,offset1]
CI> fput bootex::newsys 16 0
CI>
```

If LU 16 on both systems describes the same disk area and you will be booting your system file from LU 16, 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 (user input is underlined):

```
CI> wd /newsys
CI> instl
Enter snap file, system file, destination file, lu, source file,
[N] (no console option)
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 disk are contained in the following section. For more information about INSTL and FPUT, refer to the *RTE-A User's Manual*.

Preparation for Boot

Once a valid copy of the boot extension is installed, the system, snapshot, and boot command files must be placed on the LU that was specified when you ran INSTL. To accomplish this, the directories HELP, SYSTEM, PROGRAMS, and LIBRARIES must be established on the LU (LU 16 in our example) 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 disk LU, copy the system and snapshot files to the global directory SYSTEM, first setting the working directory to /SYSTEM. The system file should have a file type extension of SYS. If the S2 command is used in the boot file, you will need two copies of the system file. The snap file should be copied to /SYSTEM/SNAP.SNP because this is the default snap file name that LINK uses. Note that if an existing SNAP::0 file exists, it should be purged.

```
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 the global directory /SYSTEM.

Note The boot command file supplied with the Primary System Software can only be used to boot the primary. You must create a new boot command file or properly edit the primary boot command file. If you edit the primary boot command file, remove “rp,lucfg,prihp”. This will make CI the primary program and it will look for WELCOME1.CMD.

Also make any other changes appropriate for your specific configuration. If you leave LUCFG as the startup program in the boot command file, it will prohibit your system from booting. You will get *RTE-A Ready* and then the system will not respond to the terminal (see the section “Primary System Answer File” in the *RTE-A Primary System Software Installation Manual*, part number 92077-90038, for details on LUCFG).

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 12.

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 rename 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 WELCOMExx.CMD::SYSTEM, where xx is the second parameter of the ST command in the boot command file. In the example boot command file the command is ST,,1 thus CI transfers to file WELCOME1.CMD::SYSTEM. When this transfer file is completed, CI terminates and its ID segment is 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 schedules LOGON, which runs CI for qualified users. In a non-VC+ environment, an alternate name should be used for the startup version of CI. This version is then OFd 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 12. 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 and 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, and LINK, that require work areas on disk. 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 disk 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 (for example, 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 disk 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 because 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. 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 on 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 disk. 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 disk 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's Manual*, part number 92077-90002, 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's Manual* for other options. The following are the commands executed by auxiliary program CIX:

AT, CL, CO, CR, CRDIR, DC, IN, MO, PROT, PU, OWNER, TM, UNPU, RN

Format System

It is a good idea to generate a system with the VSCSI, FORMC, FORMT, or FORMF program in it, using the target system I/O configuration. VSCSI is used for formatting SCSI disks. FORMC is used for sparing bad tracks on the HP 7908, 7911, 7912, 7914, 7933, 7941/42, 7945/46 (CS/80) disks, while FORMT is used for formatting HP 7906, 7920, 7902, 9121 and 9895 mini-disks, and 7925 disks. All other non-CS/80 disks 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 disk structure) from a tape drive, and one to boot the new system from your hard disk.

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 tape. Boot the target system from the floppy disk/tape and verify operation.

If you are using a SCSI disk, use VSCSI to format your disk.

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

Boot the target system from floppy disk or tape. Mount, initialize, and run the INSTL program on the reformatted hard disk 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 disk.

Boot the target system from the hard disk and copy your files from the backup disks or cartridge tape.

Booting the Target System

Now that your files are copied onto the hard disk, your BOOTEX file is installed at a bootable location and 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

   `%B`*dd* [*ffbusc* [*file*]]

   The bootstring 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 the following string as a boot command (load a system and execute it).

   dd = device from which you are booting:

   `DC` = HP-IB disk drive or cartridge tape drive (CTD), or SCSI disk drive

   `DI` = 248x integrated disks

   `DS` = HDLC or LAN network link

   `MT` = magnetic tape or DAT tape

   `RM` = PROM module

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

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

   u = unit or head number:

Disk	Unit/Head #
7902	0 – 1
7906	0 – 3
9134A (4-vol)	0 – 3
7908/11/12/14/33/42/46 disk	0
7908/11/12/14/33/42/46 CTD	1
9145/9144A (standalone CTD)	0
other CS/80 disk	0
SCSI disk	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 disk.

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

Loading a system from disk 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/SCSI address, in the range of 0 to 7, of the disk drive that contains the system file.

If you are booting from an HP 7902 disk 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 disk set the unit/head parameter to 0.

If you are using an HP 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 either 2027<*defaultfile*> for booting from HP-IB disk or 6027<*defaultfile*> for booting from SCSI disk. This loads the default file from the disk LU that starts at head 0, cylinder 0, and sector 0 of the disk drive that has bus address 2 on the HP-IB connected to select code 27, or has bus address 6 on the SCSI connected to select code 27.

The boot extension (BOOTEX) should reside at the beginning of that disk. 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 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 disk 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 and 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 disk-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,progname,,,CM
```

CI must be restored to the progname 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 disk or magnetic tape. This will allow you to recover in the event of a disk hardware failure or an accidental purging of your system. The following methods can be used for system backup.

The FST or TF (or FC) utility will not create bootable system files; these should not be used for system backup. Refer to the *RTE-A System Manager's Manual*, part number 92077-90056, for information on a system backup strategy.

The ASAVE utility can be used online to save any disk LU containing the above system files onto DAT tape, magnetic tape, or CS/80 CTD. This LU can then be restored using an offline memory-based RTE-A system containing the ARSTR utility. Refer to the *RTE-A Backup and Disk Formatting Utilities Reference Manual*, part number 92077-90249, for more information on ASAVE and ARSTR.

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

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

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

Upgrading A990 Firmware

The A990 instruction set microcode is stored in EPROM but executes from Writeable Control Store (WCS). During the power-on self-test, the instruction set microcode is copied from the EPROM into the WCS. This allows you to modify the instruction set microcode while the computer is running. The benefit of this is that the instruction set microcode can be upgraded without physically removing the CPU board and EPROM. It is important to remember that this upgrade process is temporary because the contents of WCS are lost after a powerfail. Note also that the online upgrade process modifies the microcode in WCS to be functionally equivalent to the microcode in the new EPROM and not physically equivalent. Therefore, it is recommended that you perform a hardware upgrade of the actual EPROM to the most current revision, when possible.

The A990 EPROM contains an overall revision number that describes the revision level of the EPROM. When new versions of the EPROM are produced, the number is incremented. The EPROM revision number seen by the customer does not always increment by one, because not all revisions of EPROM are released for production.

HP provides a microcode upgrade file with RTE-A. This file contains the microcode necessary to upgrade the WCS from one revision to another. The name of the upgrade file is REV*x*TO*y*UPGRADE.MIC, where *x* and *y* are revision numbers. For example, the REV9TO12UPGRADE.MIC file is used to upgrade a Revision 9 EPROM to Revision 12. (The microcode upgrade file provided to upgrade from Revision 9 to Revision 10 is named REV10UPGRADE.MIC, which does not follow the above naming convention.)

The WCS is updated by running a program called DOWNLOAD. The DOWNLOAD program copies a microcode file to WCS, and also copies the contents of the file to a shareable EMA (SHEMA) partition in main memory. The microcode in main memory is used in case of a power failure. The DOWNLOAD program sets a flag in the operating system base page. The flag is two words, containing the physical page number of the SHEMA partition. When power returns after a power failure, VCP checks the flag, and if necessary, downloads the microcode from main memory into WCS, before passing control to RTE-A. This requires battery backup to sustain the information in main memory during the power failure.

The minimum revision of the A990 EPROM required to use the DOWNLOAD program is Revision 9. This is because the Revision 8 EPROM (the original release) does not contain a version of VCP that can download the microcode from main memory to WCS after a power failure. To determine the revision of your EPROM, link and run the A990FWID program as follows:

```
CI> wd /rte_a
CI> link a990fwid.lod /programs/a990fwid.run
CI> a990fwid
```

The output of the A990FWID program when run on a system with a Revision 9 EPROM is as follows:

```
Product Number    Supercode/Revision    Product Description
  (in octal)      octal      decimal
--------------    ------------------    ----------------------------
     000          000011           9     Entire EPROM
     001          000024          20     VCP
     002          000002           2     STST0 (Self-test module 0)
     005          000001           1     XILINX (memory/IO)
     003          000003           3     STSTC (Self-test module C)
     011          000002           2     STSTD (Self-test module D)
     012          000003           3     STSTE (Self-test module E)
     013          000004           4     STSTF (Self-test module F)
     014          000002           2     STSTG (Self-test module G)
     015          000005           5     STSTH (Self-test module H)
     210          000002           2     Baseset Microcode
```

The A990 EPROM contains several modules of code. Each piece is called a product, and has a product number. Each product has its own specific revision number. The overall revision number for the EPROM is the revision number for product 0 (zero). It can be seen in the above output that the revision of product 0 is decimal 9. If the overall revision number for the EPROM is less than 9, the DOWNLOAD program cannot be used, and you cannot perform an online upgrade of the WCS.

If the overall revision number for the EPROM is 9 or greater, then you may be able to update the WCS depending on whether the appropriate microcode upgrade file exists or not. If the REV*x*TO*y*UPGRADE.MIC file exists, and *x* and *y* are correct, then you can upgrade the WCS. For example, if you run A990FWID, and determine that the overall revision number of the EPROM is 9, and the file REV9TO12UPGRADE.MIC exists, then it is possible to upgrade the WCS to look like Revision 12.

To upgrade the WCS, link and run the DOWNLOAD program. This example uses the REV9TO12UPGRADE.MIC file. Substitute the file name that is appropriate for your system:

```
CI> wd /rte_a
CI> link download.lod /programs/download.run
CI> download rev9to12upgrade.mic
```

The output of the A990FWID program for a Revision 9 EPROM that has been upgraded to Revision 12 using the DOWNLOAD program is:

```
  Product Number    Supercode/Revision    Product Description
    (in octal)      octal      decimal
  --------------    ------------------    ----------------------------
       000          000014          12 *   Entire EPROM
       001          000024          20     VCP
       002          000002           2     STST0 (Self-test module 0)
       005          000001           1     XILINX (memory/IO)
       003          000003           3     STSTC (Self-test module C)
       011          000002           2     STSTD (Self-test module D)
       012          000003           3     STSTE (Self-test module E)
       013          000004           4     STSTF (Self-test module F)
       014          000002           2     STSTG (Self-test module G)
       015          000005           5     STSTH (Self-test module H)
       210          000002           2     Baseset Microcode

 * Control store has been upgraded to revision 12 (decimal) via software
   download.
```

Caution Because this download operation must be done whenever the machine loses memory, you should add the download command to your system welcome file. This will upgrade the control store contents every time the system is booted. Add the following line to the beginning of the welcome file:

```
download /rte_a/revxtoyupgrade.mic
```

Substitute the appropriate *x* and *y* values for your system.

The microcode upgrade file can be copied to a different directory. If this is done, make sure that you change the above line in the welcome file to reflect the current location of the microcode file.

Troubleshooting the Download Process

The DOWNLOAD process entails some risk, because the internal programming of the CPU is being modified. It is possible for the DOWNLOAD program to hang the computer if bad microcode is downloaded. If this occurs, the system can only be recovered by restoring the original instruction set microcode to the WCS. This means that the microcode must be loaded from the EPROM. If battery backup is not installed, either press the reset switch on the CPU, or cycle power. If battery backup is installed, turn off the battery backup switch, cycle power, and turn on the battery backup switch. This is because if battery backup/auto-restart is enabled, then VCP would keep trying to reload the bad microcode every time power was cycled. Main memory must be cleared to prevent this.

It is also dangerous to download a microcode upgrade file onto a system in which a different microcode upgrade file has already been downloaded. The DOWNLOAD program prevents this from happening by changing the name of the SHEMA partition containing the microcode to

"A990 Firmware". This prevents subsequent execution of the DOWNLOAD program because it cannot rename the SHEMA partition. This also prevents the customer from accidentally removing the SHEMA partition.

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 through 4 on the processor card. The various settings are listed in Table 9-1.

The switch settings given in Table 9-1 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 through 4.

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.

Installing Optional Subsystems

For installation of subsystems such as IMAGE or the NS-ARPA/1000 and DS/1000-IV networks included in your system, refer to the appropriate subsystem manuals. If your system has the optional NS or DS product, 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.

Table 9-1. Processor Card Switch Settings

Computer Action	Switches*							
	S1	S2	S3	S4	S5	S6	S7	S8
Loop on self-test Test 2 regardless of error.	D	D	D	D	z	y	D	M
Loop on self-test Test 2 and stop on error.	D	D	U	D	z	y	D	M
Run Virtual Control Panel (VCP) routine on completion of self-test.	D	U	U	D	z	y	D	M
For the following switch settings, the computer action indicated will occur if memory is lost, otherwise, the program is restarted (JMP 4B). If the autorestart feature is disabled (switch S8 down), the program cannot restart and VCP will execute. When a loader finishes an autoboot, it starts execution of the loaded program at location 02.								
Run VCP.	U	D	D	D	z	y	D	M
Speed sense and run VCP. See Notes 2 and 3.	U	U	D	U	z	y	D	M
Execute program from PROM card. In order to autoboot from PROM, the card must have select code 22. This is equivalent to the loader command %BRM. **	U	D	U	D	z	y	D	M
Load and execute the program via the HDLC or LAN card. To autoboot via the HDLC or LAN card, the card must have select code 24. This is equivalent to loader command %BDS. **	U	U	D	D	z	y	D	M
Load and execute program from the first file of the disk (via HP-IB or SCSI). To autoboot via HP-IB or SCSI, the interface card must have select code 27; and, for HP-IB, the disk drive must have HP-IB address 2 or, for SCSI, address 6. This is equivalent to the loader command %BDC. **	U	U	U	D	z	y	D	M
Execute bootstring from the time-of-day clock RAM (non-volatile RAM) on the A990 CPU card.	D	D	D	U	z	y	D	M

* D = Down (On); U = Up (Off)
** Refer to your HP 1000 computer reference manual for details.

y = D, normal mode, system console uses ENQ-ACK handshaking.
y = U, system console does not use ENQ-ACK handshaking.
z = D, normal mode, break enabled.
z = U, break disabled (but not halts).
M = D, disable autorestart (battery backup not installed).
M = U, enable autorestart (battery backup installed).

Notes:
1. Do not use any switch combination that is not specified above.
2. Use this switch configuration for normal computer operation.
3. Speed sense allows VCP to execute at any baud rate supported by the 8-channel MUX.

Installing DS Transparency Software

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

Two monitors, DSRTR and TRFAS, are used by the 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 DS transparency software for remote file access.

Install the DS transparency software as follows:

1. DSRTR and TRFAS are loaded by both DS/1000-IV and NS-ARPA/1000 installation command files. The NS-ARPA/1000 installation can be invoked by the RTE_INSTALL script. Otherwise, load DSRTR and TRFAS for your system, using the supplied load files and relocatables. Your system must have the NS-ARPA/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/NSINIT. This can be done by specifying the DS 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, or initialize NS by running NSINIT. 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.

5. Make sure the appropriate DS/NS physical link LUs are in the appropriate user and group LU access tables. Users trying to access nodes whose physical link LUs are not in their session LU access table will get a DS/NS error containing SC03 (security violation detected).

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

```
CI> li welcome1.cmd::system>27
```

where the local system is node 27. This should list the file, although the following message may 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 bootup, 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 (that is, user.group /password). To eliminate the default logon name, the *logon-name* parameter must be left off (that is, RU,DSRTR,ID). The account corresponding to the logon name specified need not exist when DSRTR is run (“DSRTR: warning – new default logon is not currently valid” will be issued). But if it does not exist when a request arrives, the request will fail (“no such account” error). Thus, it should be created before any requests arrive.

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 KILLSES), a new default logon account is defined, 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.

10

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 and, if the +D option is used, files to put on the RAM disk.
2. Merge the system file and the program files with the BUILD program.
3. Install the merged system file onto 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 disk cartridge) they are on. If you are initializing a RAM disk, you also need the files that are to be put on the RAM disk.

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 *RTE-A LINK User's Manual*, part number 92077-90035, 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 RTE-A are transportable, and can be run on other systems having the same RTE-A revision. (Programs loaded on one revision of RTE-A cannot always be transported to systems that run a different version of RTE-A.) If a program is not transportable, the BUILD program prevents 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*, part number 92077-90013.

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 that 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, all of your type 6 files, and your RAM disk 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 disk 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.

Note Refer to Appendix I for additional information on the BUILD program.

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[,+D][,command[,list[,output[,snap[,system[,op]]]]]]
```

where:

`+D` specify +D if you wish to include an initialized RAM disk.

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

list list file. Default is your terminal.

output output file. The merged system is output to this file. If this parameter is omitted, BUILD prompts for the name. The default file size is 2048 blocks. If your system size is greater than 256 pages (a file size of 2048 blocks), specify the file size in the output file namr. For example, a 1024-page system would be specified as:

output`::::8192`

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.

The 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 are prompted for them. If the files do not exist, an error message is displayed, and then you are 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 automatically creates reserved partitions of the proper size for each program that you merge with the system. If you answer NO, you 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 disk-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 (VC+ 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 can 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. At this point, if you have not included the +D option, BUILD finishes the build. If you have included the +D option, BUILD prompts you for the information it needs to initialize zero or more RAM disks at this point.

When you specified +D, you commanded BUILD to build a memory-based system with disk capabilities. This causes BUILD to do the following things differently:

1. A memory descriptor is built to describe memory left at the end of the build.
2. Swap file information is cleared.
3. BUILD does not require the output file to be large enough to hold the largest system that could be built with the specified memory.
4. BUILD optionally initializes one or more RAM disks.

These actions collectively set up the structures for disk- and memory-handling code in the operating system. To make use of these capabilities you must generate the system as if it were a disk-based system, that is, the disk- and memory-handling routines must be generated in.

If the +D option was specified, BUILD now prompts as follows:

```
   Initialize RAM disk LU [LU or /E]?
```

Enter either “LU[,MC]” or “/E”.

A /E at this point ends the BUILD in a normal manner. The LU you enter must have been generated into the system as a RAM disk. If the LU is followed by MC, BUILD mounts the LU. This means that the disk is mounted when the system is first started.

Next, BUILD prompts for files to put on the RAM disk:

```
   RAM disk file [file to move or /E]?
```

Enter either a file name or “/E”.

If you enter a file name, that file is moved to the output file and is available on the system by the given name. Entering /E terminates the file-moving process for the current global directory on the current RAM disk. If no files are moved, the /E causes BUILD to finish the current RAM disk LU by building the bit map, root directory, and volume header information. BUILD then returns to the Initialize RAM disk LU question above.

If one or more files precede the /E, BUILD prompts:

```
   RAM disk global directory name?
```

Enter a directory name.

BUILD enters this name in the root directory of the RAM disk and then returns to the RAM disk file question above. This allows you to build several global directories on the same RAM disk.

BUILD builds the RAM disk with as little wasted memory as possible. This means that, if you want a scratch area, you should generate in a RAM disk that you initialize once the system is up. If several RAM disks are initialized and one or more of them is to be dismounted and deallocated, it should be the last one as this releases memory adjacent to the rest of free memory.

BUILD Completion Codes

On termination BUILD passes a completion code to the calling program in the five return parameters recoverable with a RMPAR call. In CI, these are the set variables $RETURN1 through $RETURN5.

The values returned are:

P1	**P2**	**P3**	**P4**	**P5**	
AB	OR	TE	D		If the BUILD failed.
EN	DE	D			If the BUILD was successful.

Example Command File

The example 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 tries 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 example 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,FI,LE,,,     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.

Installing Merged System File on a Bootable Medium

The merged system file must be installed onto the proper bootable medium. The installation procedures for different media are described in the following subsections.

PROM

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.

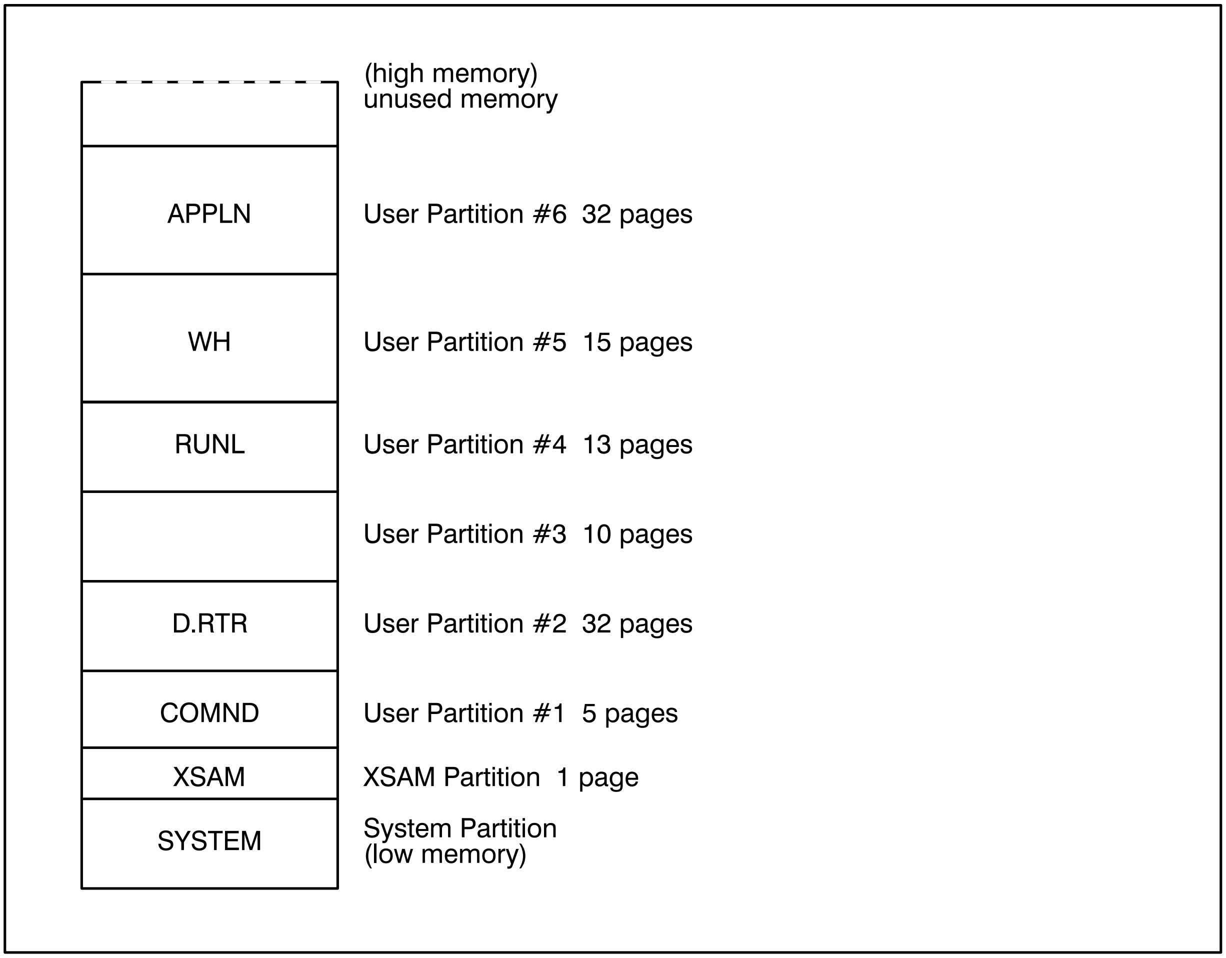

(high memory)
unused memory
APPLN
User Partition #6 32 pages
WH
User Partition #5 15 pages
RUNL
User Partition #4 13 pages
User Partition #3 10 pages
D.RTR
User Partition #2 32 pages
COMND
User Partition #1 5 pages
XSAM
XSAM Partition 1 page
SYSTEM
System Partition
(low memory)

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

CTD Tape

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 User's Manual*) to install the system file on the tape.

Magnetic or DAT Tape

To be bootable from magnetic or DAT tape, the system file must be stored onto the tape as binary code. Use the CI operator command CO to copy the system file onto tape. Refer to the *RTE-A User's Manual* for details.

Disk Volume

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

```
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 disk, it is called bootable file 0. If it begins 256 blocks (128 words per block) further in on the disk it is called bootable file 1, starting at 512 blocks is bootable file 2, and so on.

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

```
RU,FPUT,filename,lu,[offset0],[offset1]
```

where:

filename is the name of the merged system file which needs to be installed.

lu is the LU of the destination disk.

offset0 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.

offset1 is the BCM block number. This is for installing the diagnostic file.

Refer to the *RTE-A User's Manual* for further discussion of FPUT.

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

FMGR Cartridge

To be directly bootable from a FMGR cartridge, the system must be on an LU that starts at physical cylinder 0 and sector 0 of the disk drive. If the system begins on cylinder 0, sector 0 of the disk, it is called bootable file 0. If it begins 256 blocks (128 words per block) further in on the disk, it is called bootable file 1; starting at 512 blocks is bootable file 2, and so on. 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 that contain all of the systems that must be directly bootable from that cartridge.

From Disk, 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 *RTE-A User's Manual*, to install BOOTEX. (No installation setup is required to boot over a DS or NS link.)

Using BOOTEX, you can boot your memory-based system from any file on the same disk LU as BOOTEX. It is not necessary to place the file at a special location. BOOTEX uses 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

After your merged system file is installed in the proper format at a bootable location, you can boot it up. (Refer to Appendix H, "VCP Boot Information.") 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 prompts 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

   `%B`*dd*`[`*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 = disk drive or cartridge tape drive (CTD) via HP-IB; SCSI disk drive

DI = 248x integrated disk controller

DS = disk drive over DS/1000-IV or HP 12076 LAN Card

MT = magnetic tape drive via HP-IB; DAT tape drive

ff file number: not required for %BRM.

For %BDS (boot over HDLC or LAN), the *ff* parameter is an octal number from 00000 to 77777. It is converted to ASCII to form the P*fffff* file name. Refer to "DS Link" and "LAN Link" later in this chapter.

For %BDC (boot from disk), *ff* 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 disk, this is the HP-IB or SCSI address (0 to 7) of the disk unit that contains the system file. For other devices, including DS and LAN, the VCP ignores the *b* parameter, but requires a 0 as a place holder if the *ff* parameter is not zero. On DS (HDLC) and LAN, the destination VCP needs no information about the location of the system file on the source system.

u unit or head number:

Disk	**Unit/Head #**
7902	0 − 1
7906	0 − 3
7908/11/12/14/33/42/46 (Disk)	0
7908/11/12/13/33/42/46 (CTD)	1
9145/9144 Standalone CTD	0
other CS/80 disk	0
SCSI disk	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.CMD if BOOTEX is installed on a file system volume, SYSTEM if BOOTEX is installed on a FMGR cartridge. Omit this parameter when you load via DS (HDLC), LAN, CTD, CTU or PROM.

SS load and halt. If you append SS to the system file name, the system file is loaded and the computer stops with a VCP prompt (VCP>). A %R or %E in response to the VCP prompt starts execution.

Full details on the boot procedure are contained in Appendix H.

Special Considerations

Note that there are minor differences in the boot procedure for the different devices. These are discussed in the following paragraphs.

- ***PROM module.*** The VCP/loader ROM loads 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 uses a default of 22 (load from the PROM module having select code 22).

- ***Disk.*** Loading a memory-based system from disk 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 or SCSI address, in the range of 0 to 7, of the disk drive that contains the system file.

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

  If you are using an HP 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 disk, set the unit parameter to 0 for disk.

  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 HP-IB) or 6027 (SCSI) for booting from disk. This loads bootable file 0 from the disk 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 HP 9144/9145 CTD, the unit number is 0.

- ***Magnetic Tape.*** When you load your system from magnetic tape, the loader ROM rewinds the tape and then positions 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 (HDLC) link, the program PROGL, running on the neighbor computer system, translates the file number (*ff*) into a file descriptor. The default descriptor is P*nnnnn*: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 P*nnnnn*: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 bootstring.
  DEF <filename>  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 recognizes 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.

- ***LAN Link.*** When you load your system over a LAN link, the program VCPMT running on the download server translates the file number *ff* into a file descriptor. If *ff* is non-zero, the file number *ff* is translated into the file name Pnnnnn, where nnnnn is *ff*. VCPMT on the server then searches for Pnnnnn on the server; first in directory /FILES802, then in all FMGR cartridges. If *ff* is 0, VCPMT performs a translation of the file number 0 in order to determine which system file is to be downloaded. This translation process is described in the “Selecting the System File to Download” section in Chapter 11.

  The select code (sc) in the bootstring must be the select code of the LAN card. The default is 24.

  Bootstring parameter *b* for LAN is zero; therefore, if you can default the file number to zero (source file P00000) and the select code to 24, the boot command becomes simply:

  ```
  %BDS
  ```

  After you have entered your bootstring, 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 FST utility to copy your files to a CTD, mag tape, or removable disk, or you can use the CO command to make the copy on a removable disk. You can also use the ASAVE utility to back up one or more disk LUs to magnetic tape or CTD. If you have a CS/80 disk with CTD, you can use the offline backup procedure to back up the whole disk. Refer to the *RTE-A Backup and Disk Formatting Utilities Reference Manual*, part number 92077-90249, 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 FST 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 FST.

Setting Switches for Automatic Boot

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 through 4 on the processor card to the proper settings. The switch settings are shown in Table 10-1.

The switch settings given in Table 10-1 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 continues where it left off when power failed. Otherwise, the computer boots or goes into VCP mode, according to the settings of switches 1 through 4.

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.

Table 10-1. Processor Card Switch Settings

Computer Action	Switches*							
	S1	S2	S3	S4	S5	S6	S7	S8
Loop on self-test Test 2 regardless of error.	D	D	D	D	z	y	D	M
Loop on self-test Test 2 and stop on error.	D	D	U	D	z	y	D	M
Run Virtual Control Panel (VCP) routine on completion of self-test.	D	U	U	D	z	y	D	M
For the following switch settings, the computer action indicated will occur if memory is lost, otherwise, the program is restarted (JMP 4B). If the autorestart feature is disabled (switch S8 down), the program cannot restart and VCP will execute. When a loader finishes an autoboot, it starts execution of the loaded program at location 02.								
Run VCP.	U	D	D	D	z	y	D	M
Speed sense and run VCP. See Notes 2 and 3.	U	U	D	U	z	y	D	M
Execute program from PROM card. In order to autoboot from PROM, the card must have select code 22. This is equivalent to the loader command %BRM. **	U	D	U	D	z	y	D	M
Load and execute the program via the HDLC or LAN card. To autoboot via the HDLC or LAN card, the card must have select code 24. This is equivalent to loader command %BDS. **	U	U	D	D	z	y	D	M
Load and execute program from the first file of the disk (via HP-IB or SCSI). To autoboot via HP-IB or SCSI, the interface card must have select code 27; and, for HP-IB, the disk drive must have HP-IB address 2 or, for SCSI, address 6. This is equivalent to the loader command %BDC. **	U	U	U	D	z	y	D	M
Execute bootstring from the time-of-day clock RAM (non-volatile RAM) on the A990 CPU card.	D	D	D	U	z	y	D	M

* D = Down (On); U = Up (Off)

** Refer to your HP 1000 computer reference manual for details.

y = D, normal mode, system console uses ENQ-ACK handshaking.
y = U, system console does not use ENQ-ACK handshaking.
z = D, normal mode, break enabled.
z = U, break disabled (but not halts).
M = D, disable autorestart (battery backup not installed).
M = U, enable autorestart (battery backup installed).

Notes:
1. Do not use any switch combination that is not specified above.
2. Use this switch configuration for normal computer operation.
3. Speed sense allows VCP to execute at any baud rate supported by the 8-channel MUX.

11

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 disk 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 disk 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>

Enter the size of the Environment Variable Block,  in pages [2]: 4

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 modified.” is an LU access table that is used to control access to disk LUs. Details of the LU access table are in the *RTE-A System Manager’s Manual*, part number 92077-90056.

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*, part number 92077-90002.

6. Environment variables allow programs within a session to share variables (VC+ Only), and are stored in a memory space called the Environment Variable Block (EVB). Details of the EVB are in the *RTE-A User’s Manual*.

7. Because 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”.

8. Because 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.

9. 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 disk 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, MASTERGROUP, 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 is placed on the first available LU. The second command creates the subdirectory NOTES in the global directory /SYSTEM. Because 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 is 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*.

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 number 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*. 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*.

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. There 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 automatically RPs LOGON and CI (calling it CM) if they are not already RP'd. PROMT searches 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 it rereads the file. Note that OF,PROMT,ID makes PROMT unavailable, preventing multiusers from logging on and likely requiring an 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, Because 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 displays 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, because both PROMT and CM are 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 is ignored. Refer to Chapter 7 for instructions on allocating SAM.

It is recommended that PROMT be run from the WELCOME.CMD file on bootup to allow it to set up LOGON and CM (see above). The following command is sufficient:

```
   PROMT, –1
```

RESTR Program

RESTR allows recovery of a multiuser system that is out of SAM. PROMT detects this problem and takes appropriate steps to allow you to fix the problem. Depending on the source of the SAM shortage, SAM may recover without your interaction. For example, programs using SAM may release some of it. Once the SAM shortage is fixed (by offing programs, etc.) enter:

```
   GO,RESTR
```

and RESTR will restore the system to a "normal" multiuser environment.

Caution If you run out of SAM and recover with the procedure described above, your system integrity cannot be guaranteed. At this stage you MUST shutdown all applications and reboot. Failure to do so can cause unpredictable results.

The SAM, WH, and RESTR utilities should be RP'ed in the welcome file at boot time to allow recovery of a multiuser system that has run out of SAM. If RESTR is not RP'd, PROMPT attempts to RP it when PROMT detects that the system is out of SAM; however, the RP may fail due to the SAM problem. If RESTR is not RP'd and the RP RESTR fails, PROMT takes no further action and you will not be able to get the CM>, System>, or RTE: prompts. See the *RTE-A System Manager's Manual* and the *RTE-A Virtual Code+ Manual,* part no. 92078-90001, for further details.

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, and so on) 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.

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 27 pages of data per copy, plus one code partition of about 51 pages that is shared by all the copies. Additional memory is used by Environment Variable Blocks (refer to the *RTE-A System Manager's Manual* for more information).

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 CI.LOD and CIX.LOD.

In order to use CI capabilities such as aliases, functions, and file name completion, you must use the shareable version of CI. 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 gives you the CDS versions of the FMP routines. If you have programs containing any FMP calls in non-CDS code, LINK aborts 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.

A

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.

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 re-code 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 nonpartitioned 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 REelocation, SEarch, or MSearch command. Ensure that the module exists and that the form of the offending command properly references it.

Class limit specification error

Check to ensure that the class limits are numeric.

(CLASS not included in this system, syntax check only)

The system module CLASS has not been included in this generation, so the class 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 class limit should be 1,000 words less than SAM
The class 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, and so on.).

WARNING: System common module not type 6

The generator expected the relocation of a type 6 module. This is a warning only, because other module types may use system common. This warning refers to the module type field in the relocated NAM record file, not the type of the file that contains the record.

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 (for example, 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 disk 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 (VC+ 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 (for example, 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 disk 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 disk 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. (VC+ 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

CS/80 Disk Loader Errors

LOADER ERROR 411	TO reading disk type or error in initialization process: check HP-IB address, card, cable
LOADER ERROR 412	TO reading status: is device a disk?
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 disk and the disk transfer has overwritten memory.
LOADER ERROR 460	Disk not identifiable: disk ID in B-Register

SCSI Disk Loader Errors

LOADER ERROR 411	Check condition after inquiry command. TO error reading the disk, or an error in the initialization process.
LOADER ERROR 415	Check condition after read/write, parity error/timeout during DMA transfer. TO error during a seek.

Magnetic Tape Loader Errors

LOADER ERROR 510	Timeout during initialization/read ID.
LOADER ERROR 511	Timeout when issuing end/select unit.
LOADER ERROR 512	Magnetic tape off line.
LOADER ERROR 513	No write ring.
LOADER ERROR 514	Timeout during End command.
LOADER ERROR 515	Timeout waiting for rewind completion.
LOADER ERROR 517	Timeout waiting for DMA transfer.
LOADER ERROR 520	Parity error during DMA transfer.
LOADER ERROR 521	Timeout doing a PHI flush.
LOADER ERROR 522	Timeout waiting for DSJ.
LOADER ERROR 523	Bad DSJ response.
LOADER ERROR 525	Timeout waiting for Mag Tape Not Busy.
LOADER ERROR 530	Timeout after issuing a command.
LOADER ERROR 531	Parallel Poll timeout 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.

SCSI DAT Tape Loader Errors

LOADER ERROR 510	Check condition after inquiry command.
LOADER ERROR 513	Check condition after read/write, parity error/timeout during DMA transfer.
LOADER ERROR 560	Check condition after rewind command.

HP 12022A Disk Interface Loader Errors

LOADER ERROR 610	Timeout after SDH (Sector Drive Head) for read/write.
LOADER ERROR 611	Timeout after cylinder high.
LOADER ERROR 612	Timeout after cylinder low.
LOADER ERROR 613	Timeout after sector.
LOADER ERROR 614	Timeout after sector count.
LOADER ERROR 615	Timeout after read/write command.
LOADER ERROR 616	Timeout after DMA read/write transfer.
LOADER ERROR 617	Parity error during transfer.
LOADER ERROR 620	Fixed disk not ready.
LOADER ERROR 630	Timeout after request status register.
LOADER ERROR 631	Timeout after read status register.
LOADER ERROR 632	Timeout after waiting for not busy.
LOADER ERROR 633	Timeout after request error register.
LOADER ERROR 634	Timeout after read error register.
LOADER ERROR 635	Status Error: A-Register = status register; * B-Register = error register. *
LOADER ERROR 650	Timeout after SDH register for restore.
LOADER ERROR 651	Timeout after restore.
LOADER ERROR 660	Disk not defined.

Other Loader Errors

LOADER ERROR 1024 or
LOADER ERROR 1025 Possible meanings:

1. Booting from CS/80 disk 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 disk was not successful. BOOTEX may be corrupt.

4. Faulty tape control board in the CS/80 runstring.

5. Incorrect VCP file number in the runstring.

* Refer to the CS/80 Disk Driver DD*33 section of the *RTE-A Driver Reference Manual*, part number 92077-90011, for descriptions of the Status Register and the Error Register.

B

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 ,Use current page linking
*
*
*
*              SYSTEM RELOCATION PHASE (Use one of the following RPLs)
*
*
*     ###################################################################
*     #                                                                 #
*     #   If you use an RPL file for a hardware configuration with      #
*     #   fewer features than yours, your system will not be performing #
*     #   as well as it could.  You may not use an RPL file for a       #
*     #   hardware configuration with more features than yours.         #
*     #                                                                 #
*     #   Choose the correct RPL file for your system from the          #
*     #   choices below and remove the ”*” in front of it.             #
*     #                                                                 #
*     ###################################################################
*
*                                                      DOUBLE PRECISION
*    RPL FILE             PROCESSOR TYPE     CDS        FLOATING POINT
*   ----------           ----------------   -----     ------------------
*re /rte_a/%rpl40,,            A400          NO              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
*re /rte_a/rpl_a990.rel,,      A990          NO              YES
*re /vcplus/rpl_a990_cds.rel,, A990          YES             YES
*

*     ###########################################################
*     #                                                         #
*     #  For A900 with either NS/1000 or %envrn, choose either  #
*     #  rpl_a900_rev4.rel (rev.4 firmware or later) or         #
*     #  xmb.rel (pre rev.4 firmware)                           #
*     #                                                         #
*     ###########################################################
```

```
*
*re /rte_a/rpl_a900_rev4.rel
*re /rte_a/xmb.rel,mb02         For NS/1000 with A900 pre rev.4 firmware only
*re /rte_a/xmb.rel,mb12         For either NS/1000 or %envrn with
*                                  A900 pre rev.4 firmware only
*re /rte_a/xmb.rel,mb01         For %envrn with A900 pre rev.4 firmware only
*re /rte_a/xmb.rel,mb10         For %envrn with A900 pre rev.4 firmware only
*re /rte_a/xmb.rel,mb21         For %envrn with A900 pre rev.4 firmware only
*
*
*
*     #################################################################
*     #                                                               #
*     #   If you are using an A600 (not an A600+) WITHOUT CDS, then   #
*     #   the following paragraph applies.  If you are using any      #
*     #   other type of A–Series processor, (A400 with or without     #
*     #   CDS, A600+, A700 or A900) then the following paragraph      #
*     #   does NOT apply.                                             #
*     #                                                               #
*     #   The two routines .DMP and .DDI must 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 (using %RPL60).  If you are using any other RPL,       #
*     #   these modules will cause duplicate entry point errors       #
*     #   if included in the generation.                             #
*     #                                                               #
*     #################################################################
*
*!RPL60_!re /rte_a/$math,.dmp   Required for A600 (not A600+)   (RPL60)
*!RPL60_!re /rte_a/$math,.ddi   Required for A600 (not A600+)   (RPL60)
*

* SYSTEM MODULES
*
* =======================
* System Relocation Phase
* =======================
le,off,                   Do not list the module entry points
er,,                      Echo errors to the terminal
re %vctr:92570,,          Entry points
tg 700                    Number of tags required
re %mapos:92077,,         Partitioned OS tag routines
re %rpl60:92077,,         No CDS no double precision floating point
re %exec:92077,,          EXEC request processing
re %rtioa:92077,,         Real–Time I/O control
re %maps:92570,,          Dynamic mapping system routines
re %progs:92570,,         Program state processing
re %util:92570,,          System variables and utilities
re %sam:92077,,           System available memory
re %sched:92570,,         Programmatic program scheduling
re %strng:92077,,         Runstring passing
re %erlog:92570,,         Error Logging
re %opmsg:92077,,         Operating system messages
re %sycom:92570,,         Operator commands
re %iomod:92570,,         I/O module
re %id*43:92077,,         Power fail driver
re %signl:92570,,          Signals module
re %$idrpl:92570,,         System ID dup IDRPL
re $math:24998,.dmp,       Required for A600 without CDS (RPL60)
re $math:24998,.ddi,       Required for A600 without CDS (RPL60)
*

*
* NON-PARTITIONED DRIVERS
*
REL,
REL,
REL,
REL,
REL,
REL,
REL,
REL,
*
* SEARCH LIBRARIES
*
se /libraries/$SYSLB.LIB,,           Search the System Library
se /libraries/$MATH.LIB,,            Search the Math Library
pa PERR,XCMND,STAT,DSQ,VEMA
pa CDSFH
ms /libraries/$SYSA.LIB,,            Search the System Dummy Library
*
end,                                 Terminate System Relocation
*
*
* OS PARTITIONS
*
*
re /rte_a/%XCMND,,                   Operator Command Extension
al
re /rte_a/%STAT,,                    Status Commands
re /rte_a/%DSQ,,
end
*
*
* re  /vcplus/%CDSFH,,               CDS Systems Only
al
re /rte_a/%PERR,,                    Parity Error Handling
re /rte_a/%VEMA,,                    Virtual Memory Module
end
*
*
* Driver partitions
*
*
re  /rte_a/%dd*33,,                  Device driver for CS/80 disks
end
*
re /rte_a/id.100.rel,,               Interface driver for 12005 ASIC
end
*
re /rte_a/%id*37,,                   Interface driver for CS/80 disk
re /rte_a/%id*52,,                   Parallel Interface intercomputer
```

```
                                    communications driver
end
*
re /rte_a/id800.rel,,               Interface driver for 12040D 8 channel MUX
re /rte_a/%dd*23,,                  Device driver for magnetic tape drive
end
*
re /ns1000/dsrel/%adv00,,           Device driver for NS/1000
re /rte_a/ddco1.rel,,               Terminal device driver
end
*
re /rte_a/%id*50,,                  Interface driver for Parallel Card
re /rte_a/%dd*30,,                  Device driver for disks (ICD)
end
*
re /rte_a/%dd*12,,                  Device driver for HP-IB line printer
re /rte_a/%dd*20,,                  CTU device driver
end
*
re /rte_a/%ddc12,,                  Device driver for HP-IB line printer
end
*
re /rte_a/%id*27,,                  Interface driver for 248x integrated disk
re /rte_a/%dd*24,,                  Device driver for 7974A/7978A streaming
*                                   tape
end
*
*        end driver partition
*
```

System I/O Configuration Worksheet

Device	Interface	Logical Unit	Select Code	HP-IB/SCSI Address

IFT, ,SC,E ,QU: TX: ,IT:

IFT/DVT Worksheet

Interface name: ________________________________
I/O slot number: ________

IFT, ______________ , SC:____ , E_________ , QU:__________ , TX:_____________,
Interface Driver Name (Default file) Select Code Entry Point Queuing Table Extension

IT:_________ , AL:___________
Interface Type Port Map 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:
Timeout:	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 Parameters:					
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{,Eentry point}{,QU:qu}{,TX:tx}{IT:it}{,AL,al}{,PS}
*
*        DVT Format:
*
*            DVT,  {namr},{Mmodel number},–
*                   LU: logical unit,–
*                  {Eentry 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 and XSAM allocation
*
SL,                                 ,Spool limits
*
BG,                                 ,Background swap priority allocation
*
QU                                  ,Time slice value
*
SP,                                 ,Shared programs
*
MB,                                 ,Memory block for NS (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,/rte_a/%MSGTB,,                  Message table
END
*
RE,/rte_a/%$M000,,                  Message module
END
*
RE,/vcplus/security.rel,,           VC+ only
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
```

C

Configuring a Disk Subsystem

HP supplies a system configuration that may be used with the default values found in the relocatable code or gen file for the disk 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, /rte_a/%DD*33, M7908_CF:0, LU:16, ...
DVT, /rte_a/%DD*33, M7908_CF:1, LU:17, ...

DVT, /rte_a/%DDQ30_GEN.REL, M64MB:0, LU:10, ...
DVT, /rte_a/%DDQ30_GEN.REL, M64MB:1, LU:11, ...
```

M7908_CF:x or M64MB:x is the model number and default value designator. To see what these default values are for the various disks, refer to the appropriate table contained in Appendix E of this manual.

If you want to configure the disk 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,/rte_a/%DD*33,M7908_CF:0,LU:16,DP:1:a:b:c:d:e,DP:6:f:g:h
DVT,/rte_a/%DD*33,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.

Disk Subsystem Categories

There are four categories of disk subsystems: the CS/80 (command-set 80 and subset 80) disks with and without Cartridge Tape Drive (CTD), Multiple Access Controller (MAC) disks, the compact disk subsystems, and the SCSI disks. The model numbers associated with each type of drive are:

CS/80:	7908P/R	–	16Mbyte	(fixed disk)
	7941A	–	24Mbyte	(fixed disk without CTD)
	7942A	–	24Mbyte	(fixed disk with CTD)
	7911P/R	–	28Mbyte	(fixed disk)
	7907A	–	20/20Mbyte	(one fixed, one removable disk)
	7945A	–	55Mbyte	(fixed disk without CTD)
	7946A	–	55Mbyte	(fixed disk with CTD)
	7912P/R	–	65Mbyte	(fixed disk)
	7957A/B	–	81Mbyte	(fixed disk)
	7958A	–	130Mbyte	(fixed disk)
	7914P/R	–	132Mbyte	(fixed disk with/without CTD)
	7914CT	–	132Mbyte	(fixed disk with CTD)
	7914ST	–	132Mbyte	(fixed disk with 7974A tape unit)
	7958B	–	152Mbyte	(fixed disk)
	7962B	–	152Mbyte	(expandable fixed disk)
	9262B	–	152Mbyte	(removable disk)
	7959B	–	304Mbyte	(fixed disk)
	7963B	–	304Mbyte	(expandable fixed disk)
	9263B	–	304Mbyte	(removable disk)
	7936A	–	307Mbyte	(fixed disk)
	C2200	–	335Mbyte	(fixed disk)
	7933A	–	404Mbyte	(fixed disk)
	7935A	–	404Mbyte	(removable disk)
	7937A	–	571Mbyte	(fixed disk)
	C2202	–	670Mbyte	(fixed disk with cache)
	C2203	–	670Mbyte	(fixed disk)
MAC:	7906M/S	–	20Mbyte	(1 fixed, 1 removable disk)
	7920M/S	–	50Mbyte	(removable disk)
	7925M/S	–	120Mbyte	(removable disk)
Compact:	243x/8x*	–	10Mbyte	(integral fixed)
		–	14.5Mbyte	(integral fixed)
		–	20Mbyte	(integral fixed)
	12122A	–	20Mbyte	(fixed)
		–	270Kbyte	(floppy)
		–	630Kbyte	(floppy)
	9121S	–	270Kbyte	(floppy)
	9121D	–	2x270Kbyte	(floppy)
	9122C	–	2x2Mbyte	(dual-sided floppy)
	9122D	–	2x630Kbyte	(dual-sided floppy)
	9133XV	–	15Mbyte	(fixed)
		–	270Kbyte	(floppy)
	9133D	–	15Mbyte	(fixed)
		–	630Kbyte	(floppy)

* Micro/1000 integrated disk using HP 12022A controller.

Compact:	9133H	–	20Mbyte	(fixed)
(cont.)		–	630Kbyte	(floppy)
	9133L	–	40Mbyte	(fixed)
		–	630Kbyte	(floppy)
	9895A	–	2.3Mbyte	(dual-sided floppy)
SCSI:	C2212A	–	330Mbyte	(fixed disk)
	C2460F/R	–	422Mbyte	(fixed disk)
	C1701A/C/M	–	650Mbyte	(magneto-optical disk)
	C1711C/M	–	650Mbyte	(magneto-optical disk)
	C2213A	–	660Mbyte	(fixed disk)
	C2214A/B	–	670Mbyte	(fixed disk)
	C2216T	–	670Mbyte	(fixed disk)
	C2461F/R	–	670Mbyte	(fixed disk)
	C2481A	–	670Mbyte	(fixed disk)
	C3020T/R	–	1GB/2GB	(fixed disk/DDS (DAT) tape drive)
	C2217T	–	1.3Gbyte	(fixed disk)
	C2462F/R	–	1.3Gbyte	(fixed disk)
	C2482A	–	1.3Gbyte	(fixed disk)
	C1716T	–	1.3Gbyte	(magneto-optical disk)
	C2550A	–	1.3Gbyte	(magneto-optical disk)
	C3021T/R	–	2GB/2GB	(fixed disk/DDS (DAT) tape drive)
	C3023T/R	–	2Gbyte	(fixed disk)
	C3024T/R	–	2GB/2GB	(fixed disk/fixed disk)
	C3040T/R	–	2Gbyte	(fixed disk)
	C3041T/R	–	2Gbyte	(fixed disk)

Disk Configuration Steps

Disk configuration involves several steps:

1. Divide the disk into groups of contiguous tracks.

2. Allocate spare tracks for each of these groups (9895, 9121, and MAC disks 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 disk subsystem.

File System Considerations

For a disk 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, disk space is logically divided by LUs, with one directory per LU. Dividing a disk 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 disk LUs, with the remaining disk 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.

The hierarchical FMP file system has a limitation of 128 blocks per track and the FMGR file system has a limitation of 64 blocks per track. The maximum number of tracks per LU is dependent on the disk driver being used. With SCSI disk drives, using driver DDQ30, the maximum number of tracks per LU is 65534. With all other types of disk drives, the maximum number of tracks per LU is 32767.

Note Due to the sector skipping algorithm of FMGR, the number of blocks per track on a FMGR disk LU must not be a multiple of 7. The recommended track sizes are either 48 or 64 blocks per track.

Configuring SCSI Disks

The default generation records available for configuring a SCSI disk are given in source file DDQ30_GEN.MAC. You can list this file to see the gen records available. The gen records are organized into groups according to the size of the disk LUs and are named as follows:

`M`*xxx*`MB:`*y*	for 16, 32, 128, 256, and 512-Mbyte disk LUs
`M`*xxx*`GB:`*y*	for 1-Gbyte disk LUs

where:

xxx is the size of the logical disk unit and
y is the number of the gen record (starting from 0) that corresponds to the position of that logical disk unit.

Because most disks will not be exactly filled by the modular sizes, some “end pieces” are named by the model number:

`M`*xxx*`MB_`*b*`:`*z*

where:

xxx is the size in Mbytes,
b is the blocking factor minus 1, and
z is an alphabetic chosen to be unique among the GEN records of the same size.

For the 650-Mbyte HP C1701A Magneto-Optical Disk Drive, model number M650A_1 or M650A_3 is used to define the entire surface as a single logical LU.

Refer to file DDQ30_GEN.MAC for more detailed information regarding the SCSI gen records.

SCSI Disk Driver Parameters

Configuring a disk into the desired logical units involves setting driver parameters in the DVT for each LU. These parameters will contain all necessary information. The driver parameters (DP:) for the DVT entries are:

DP1 SCSI Address
DP2 Disk Drive Unit Number (bits 15 through 8)
DP3 Pass Through Fence
DP4 High Order Word of Starting Block Number
DP5 Low Order Word of Starting Block Number
DP6 Number of Tracks for the LU
DP7 Number of Blocks per Track for the LU
DP8 Spin-Up/Down (bit 15); Eject (bit 14); Blocking Factor minus 1 (bits 1 and 0).

SCSI Address (DP1)

The SCSI address is a switch-selectable number that identifies a disk (or other device) connected to the SCSI card. The SCSI address must be a unique number for each device.

Note that bit 15 of DP1 should be set for devices that do not retry the RESELECT command if it times out. Setting this bit can impact performance, but may be required for such devices as the C1701C Magneto-Optical Disk drive when connected to the same bus as a DAT drive.

Disk Drive Unit Number (DP2)

The SCSI disk drive unit number is always set to 0.

Pass-Through Fence (DP3)

The pass-through fence is the number of bytes above which *pass-through mode* is enabled. Pass-through mode allows data to be transferred directly between the HP 1000 and the SCSI device. Memory on the SCSI interface card is not used for data transfers in pass-through mode. The default value is 6144 bytes. You can alter the pass-through fence by modifying DP3 at generation time. The value must be a multiple of 128, that is, bits 0 through 6 of DP3 are ignored.

Starting Block Number (DP4 and DP5)

Driver parameters 4 and 5 form a double-integer starting block address.

Number of Tracks (DP6)

The total number of tracks in the disk LU.

Number of Blocks per Track (DP7)

The number of blocks per track on that LU. The default value is 64.

Driver Parameter 8 (DP8)

Driver parameter 8 defines the blocking factor for all SCSI hard disks, and the spin-up/down and eject on dismount feature for the magneto-optical disk.

Bits 1 and 0 define the blocking factor minus one. The default blocking factor for all SCSI hard disks is 2 (DP8 equals 1), which is 512 bytes per SCSI block. The value to use is a function of the specific disk media being generated into the system. The blocking factor values are:

blocking factor of 1	= 256 bytes/sector	(DP8 = 0)
blocking factor of 2	= 512 bytes/sector	(DP8 = 1)
blocking factor of 3	= 768 bytes/sector	(DP8 = 2)
blocking factor of 4	= 1024 bytes/sector	(DP8 = 3)

Bit 15, if set, indicates that the SCSI driver manages spin-up and spin-down.

Bit 14, if set, indicates that the SCSI driver ejects on last dismount.

The SCSI disk device driver has the capability to spin-up/down or eject the media from a SCSI magneto-optical (MO) device when the disk is no longer in use. Sending a 40b control request to a SCSI MO disk LU indicates that the LU is being used. When a 41b request is issued and no other LU on the node list for the MO disk is being used, the media will spin-down or eject. (A detailed explanation of this feature can be found in the file DDQ30_GEN.MAC.)

D.RTR can be configured to issue the CN 40b and 41b requests to the SCSI disk LU whenever a file system mount or dismount command is processed. By default, this feature is disabled. Refer to the instructions in the files /VCPLUS/DDRTR_CDS.LOD and /VCPLUS/DDMAX.MAC to enable this feature. (Note that this also requires that all of the MO disk LUs on the same node list be generated with the same value for driver parameter 8, setting either bit 15 or bit 14, but not both.)

Configuring Non-CS/80 HP-IB Disks

This section covers the physical structure of a disk drive, the logical structure of a disk, and the driver parameters associated with a disk subsystem.

Physical Disk Structure

The magnetic storage medium in a disk drive is the disk platter. The disk 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 disk heads.

Disk heads are electro-mechanical devices that move over a surface, reading data from the disk or writing data to the disk as the disk rotates. There is one head per disk surface. If the disk has more than one head, all of the heads are connected to the same servo-motor, thus all heads move together.

Logical Disk 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 disk driver allows information to be written across track boundaries. This means that a system disk may operate in one of two modes: surface mode or cylinder mode. When you configure a disk, the LUs on that disk must be all in surface mode or all in cylinder mode. Each disk type has a standard mode that is determined by access considerations. Access modes are further discussed in the sections on the individual disk types.

Surface Mode

In surface mode, each disk LU is made up of tracks that are all on one disk 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 disk.

Cylinder Mode

In cylinder mode, tracks are arranged in groups of cylinders. A disk cylinder includes all tracks with a given track number on all surfaces of the disk. 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 disk. 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 disks.

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.

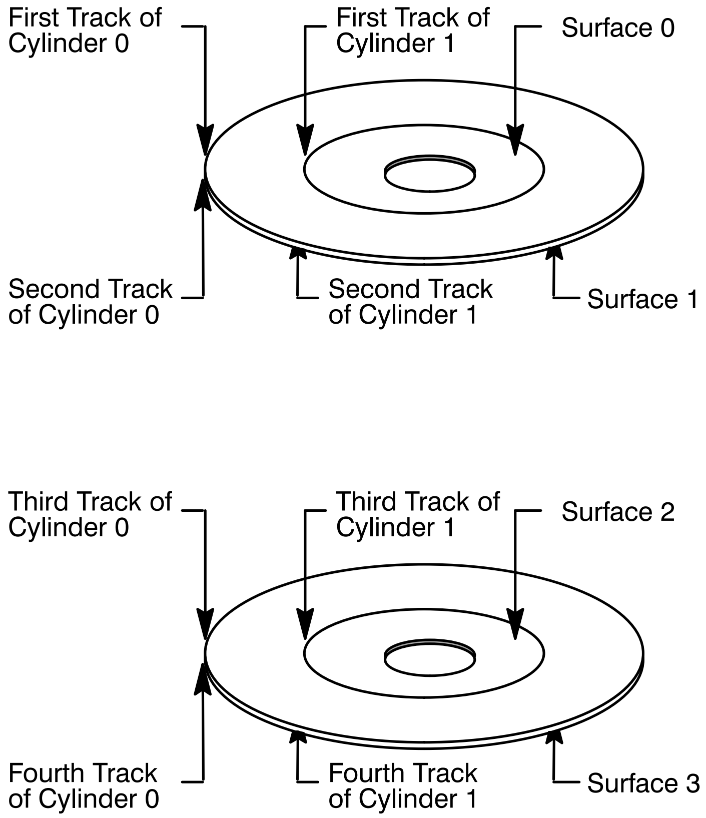

First Track of Cylinder 0
First Track of Cylinder 1
Surface 0
Second Track of Cylinder 0
Second Track of Cylinder 1
Surface 1
Third Track of Cylinder 0
Third Track of Cylinder 1
Surface 2
Fourth Track of Cylinder 0
Fourth Track of Cylinder 1
Surface 3

Figure C-1. Arrangement of Disk Cylinders

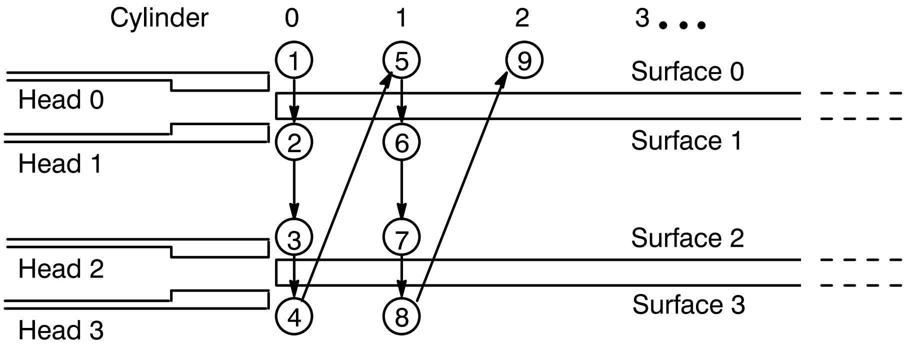


Figure C-2. Track Access in Cylinder Mode

Non-CS/80 HP-IB Disk Driver DD*30 Parameters

Configuring a disk 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 HP-IB disk DVT entries are the following:

Driver Parameter 1: HP-IB Address
2: Disk 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 Disk
8: Number of Surfaces for LU

To help in configuring the disk, 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 disk (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.

Disk Drive Unit Number (DP2)

This number is important only for the 9895, 7906M, and 7925M disk drives and the 248x integrated hard disk. For any other type of disk drive, enter 0. The disk drive unit is a switch-selectable number that identifies the drive to the disk controller. Directions for setting this number are contained in the reference manual for your disk.

Starting Head for LU (DP3)

Disk head numbers start at 0 (the top surface). This parameter defines the head (or surface) on which a disk LU begins.

Starting Cylinder for LU (DP4)

Cylinder numbers start at 0 (the outermost cylinder). This parameter defines the cylinder on which a disk LU begins.

Number of Spares for LU (DP5)

Each disk 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 disks. For all other disks, set the number of spare tracks to zero.

Number of Tracks for LU (DP6)

This parameter is the total number of tracks in the disk 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:

[(no. of heads)*(no. of cylinders)] − no. of spare tracks = no. of tracks for LU

Number of Blocks per Track for Disk (DP7)

This number indicates the amount of storage available on each track of the disk. This number is different for each disk type and is given in the sections on the individual disk 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 disk drive (cylinder mode). This number must be the same for all disk LUs referring to the same disk controller.

The number of surfaces is usually either one (for surface mode) or all (for cylinder mode). The 7906M disk 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 disk only in surface mode.

Configuring a RAM Disk

The RAM disk driver IDR37.REL should be relocated along with the other drivers. The IFT/DVT entries are as follows:

```
   ift,idr37.rel
   dvt,,, dx:8, dt:33b, LU:xx
```

where: *xx* = LU number for the RAM disk.

As many dest entries as desired may be controlled by the single IFT entry.

After booting the system, the RAM disk must be assigned memory to manage. This is done with a control request 25b as follows:

```
   CN xx 25B, zz [yy]
```

where: *xx* is the LU number for the RAM disk.

zz is the number of pages of RAM to use.

yy is the number of pages per track (1 through 8); the default is 4.

D.RTR uses the track size to set the default directory size (1 track). This is the amount of space it will use for the root directory and the default for the global subdirectories. It is recommended that you use a small track size to conserve memory.

Configuring CS/80 Disks

Cylinder mode is the only mode of the CS/80 disks. CS/80 disks 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 disks. 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 disk subsystem, do not allow the areas specified to overlap each other and do not exceed the highest block address of the disk.

CS/80 Driver Parameters

Configuring a disk 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:

Disk Driver Parameters

DP1	HP-IB Address
DP2	Disk 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 Disk 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);Disk Unit Number(7 bits);Volume Number(8 bits)
DP4	\ Starting Block Address of Disk 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 disk 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 disk, modify the driver parameter section of the IFT/DVT worksheet to include the above parameters explicitly:

Disk Driver Parameters		LU Number				
DP1	HP-IB Addr.	DP: 1	DP: 1	DP: 1	DP: 1	DP: 1
DP2	Unit & Vol #	:	:	:	:	:
DP3	Start	:	:	:	:	:
DP4	Block	:	:	:	:	:
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	Disk Cache	:	:	:	:	:
DP5	Block	:	:	:	:	:
DP6	Disk Cache	DP: 6	DP: 6	DP: 6	DP: 6	DP: 6
DP7	Block	:	:	:	:	:
DP8	Reserved	:	:	:	:	:
	:					

HP-IB Address (DP1)

The HP-IB address is a switch-selectable number that identifies a disk (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.

Disk Drive Unit Number (DP2)

The disk drive unit is a number that identifies the drive to the disk controller. Directions for setting this number may be found in the reference manual for your disk.

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 disk unit number is contained in the upper 7 bits while the disk 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 Disk (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:

Most Significant Block = Previous Starting Block +
(Number of Tracks * Number of Blocks/Track)

For example, a sample generation includes the following:

```
DVT,/rte_a/%DD*33,LU:16,M7912_CF:0,DP:1:0:0,DP:3:0:0:0:400,DP:7:48:0
DVT,/rte_a/%DD*33,LU:17,M7912_CF:1,DP:1:0:0,DP:3:0:0:19200:943,DP:7:48:0
   :
DVT,/rte_a/%DD*33,LU:18,M7912_CF:2,DP:1:0:0,DP:3:0:0:64750:320,DP:7:48:0
DVT,/rte_a/%DD*33,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 disk 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 disk 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 Disk (DP7)

The Number of Blocks per Track for Disk indicates the amount of storage available on each track of the disk. 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 disk. For example, the 7933 disk has a capacity of 1,579,872 blocks. Dividing this by the default 48 blocks/track yields 32,914 tracks. Note that the hierarchical FMP file system has a limitation of 128 blocks per track; you cannot mount a disk LU with a track size greater than 128 blocks per track. The FMGR file system has a limitation of 64 blocks per track. For optimal performance, the recommended value is 64 blocks per track.

Reserved Parameter (DP8)

The Reserved parameter must always be 0.

Starting Block Address of Disk Cache

All accesses to and from the Cartridge Tape Drive are made via a disk cache. The size of the disk cache must be in 256 physical blocks; therefore, if the layout of the disk is defined differently from the primary, be sure there is room for the 256 physical blocks for the disk cache.

Caution If you are changing the track map layout of a CS/80 disk 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 Disk (Unit 0)

LU	18
DP1	0
DP2	0
DP3	0
DP4	0
DP5	64750
DP6	320
DP7	48
DP8	0

Sample 7907 Worksheet, Removable Disk (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 Disk	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 disk cache is 286 blocks, derived from the difference between 64750 (the maximum number of blocks on the 7908 disk) 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 disk.

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 disk.

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 disk.

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 disk.

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 disk.

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 disk.

Sample 9122C/D Worksheet

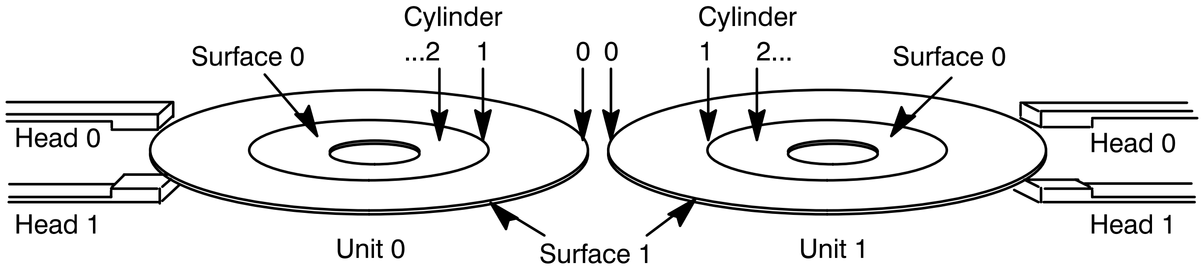
LU	0	1
DP1	0	0
DP2	0	400B
DP3	0	0
DP4	0	0
DP5	0	0
DP6	154	154
DP7	16	16
DP8	0	0

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 Disk Drive

The organization of the 7902 dual flexible disk 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.



A cylinder is one track from surface 0 and one track from surface 1.

Figure C-3. 7902 Dual Flexible Disk Drive

The 7902 Disk 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 disk drives. Typically, one logical unit is assigned to each unit of a flexible disk. In this case you can use the defaults provided by the disk 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.

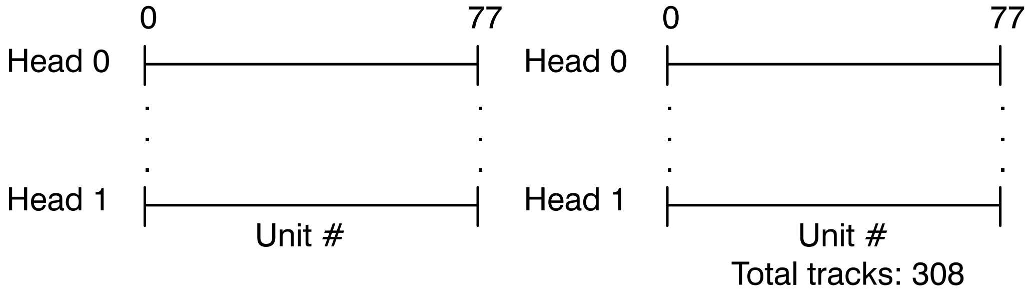
To check to see that you have not overlapped any LUs or wasted any disk space, add up all of the tracks and spares of all the LUs on the disk drive. This number should equal the Total Tracks number shown on the 7902 Worksheet.

You should make one LU out of each of the flexible disk drives of the 7902. Each flexible disk 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 disk is fairly common, so by using this convention you can easily transport your flexible disks 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 Disk Worksheet

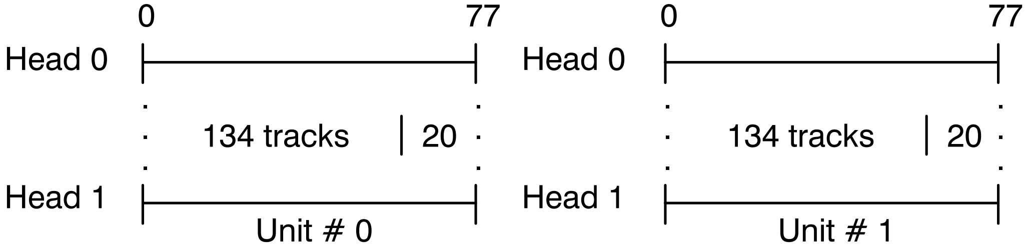
Cylinders:



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 Disk 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 Disk Drives

On certain HP 1000 computers, flexible mini disks are available. These drives support dual-sided flexible mini disks. Usually two independent disk 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.

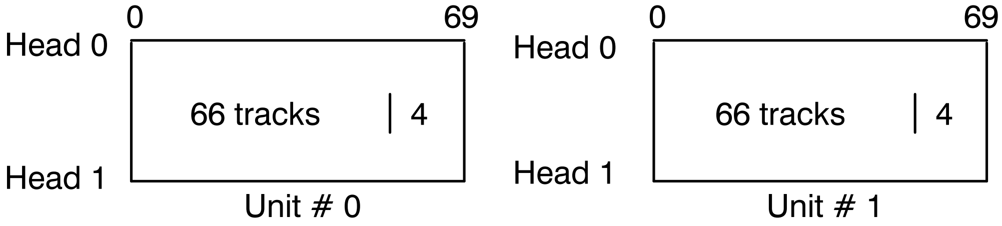
The mini-disk controller for the Model 6 is located in the backplane, and a set of dip switches on this board select the HP-IB address of the two disk units. Mini disks may be daisy-chained from the same HP-IB board that controls the 7902, 7906, 7908, 7911 and 7912 disk drives. The unit numbers of the two mini-disk 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 disk 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 disk.

It is wasteful and cumbersome to try to define more than one logical unit per mini disk. This example shows two disk units, each defined as one LU:

Sample Model 6 Integrated Disk Worksheet

Cylinders:

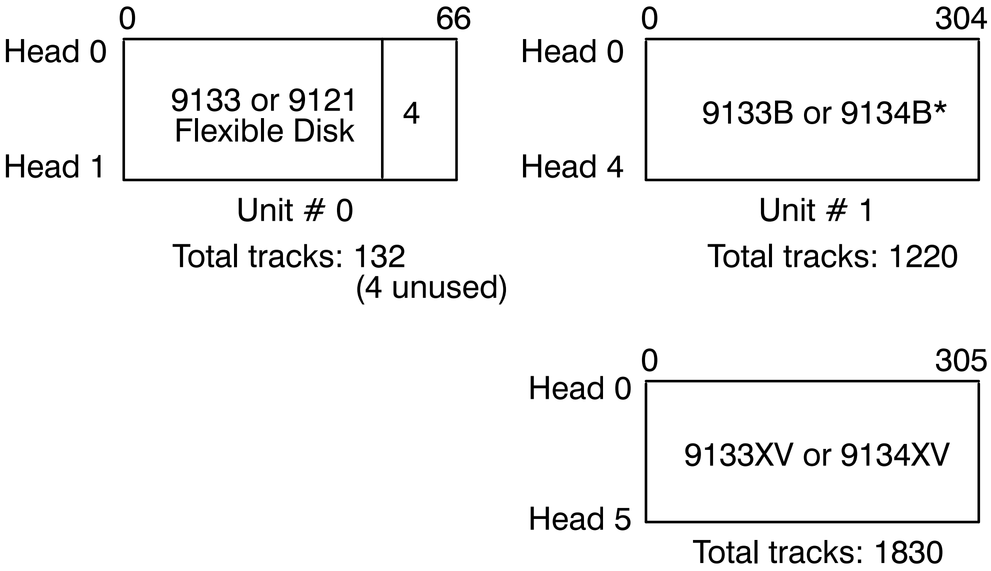


Each unit – 66 tracks used, 4 tracks unused: Total tracks: 140

*	Disk 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 Disk Worksheet

Cylinders:



* The only difference between the 9133 and 9134 is the microfloppy 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-Volume			
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

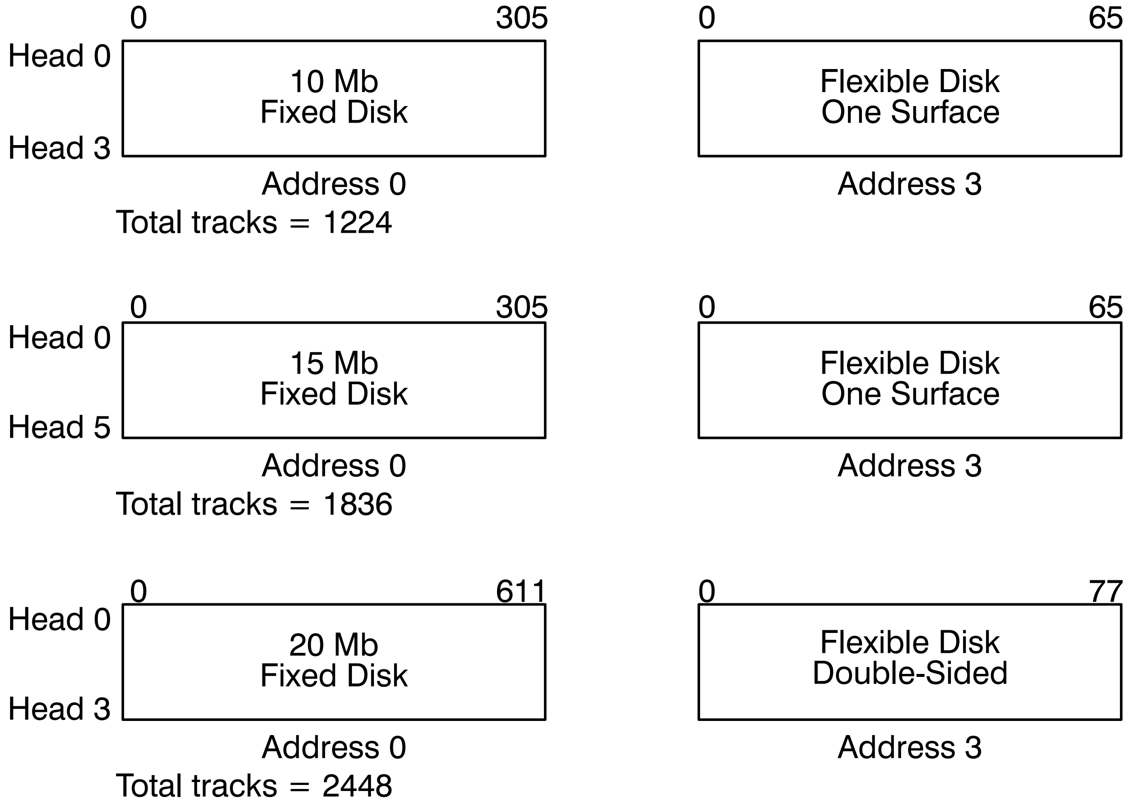
		9133B/4B Single Volume			9133XV/34XV Single Volume			
DP1	HPIB Address	7	7	7	7	7	7	7
DP2	Unit Number	0	0	0	0	0	0	0
DP3	Start Head	0	0	0	0	0	0	0
DP4	Start Cyl	0	118	152	0	104	159	236
DP5	Spares	0	0	0	0	0	0	0
DP6	Tracks	472	136	612	624	330	462	414
DP7	Blks/Track	31	31	31	31	31	31	31
DP8	Surfaces	4	4	4	6	2	6	6

Microsystems With Integrated Peripherals (243x/8x)

The integrated peripherals options on microsystems provide two disk drives, one fixed and one removable.

Configuration:

Cylinders:



		mf*	10Mb Hard Disk			15Mb Hard Disk				20Mb Hard Disk		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

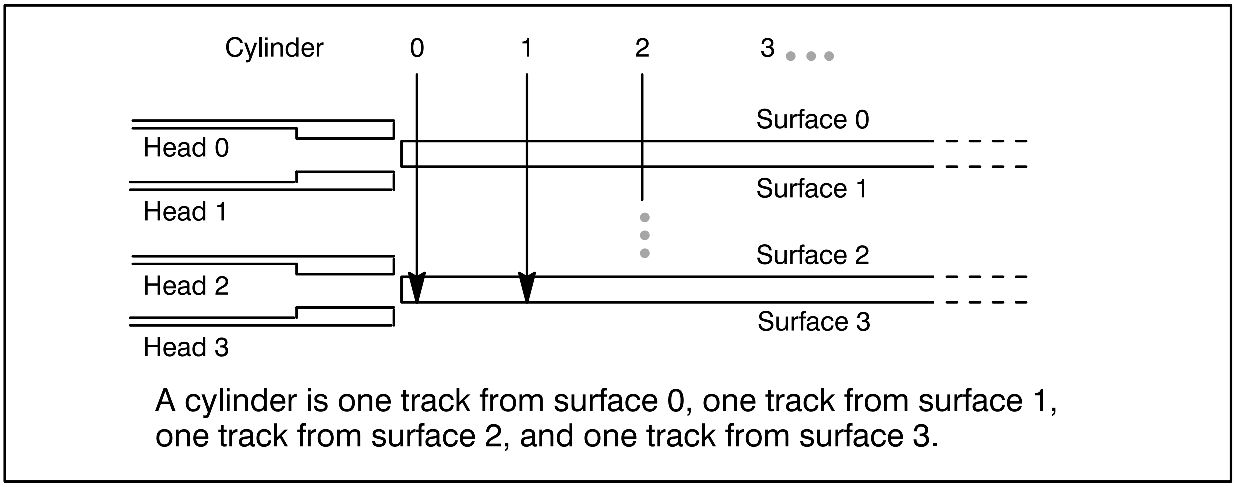
You can select between different ports and drives on the Disk Interface card (12022A). First, the drive Address (DP1) allows you to select between microfloppies, or one of two hard disks. Each microfloppy has its own port, both of which are Address 3. The Unit number (DP2) selects microfloppy unit 0 or unit 1. Note that only one microfloppy, unit 0, is currently provided for the microsystem.

Hard disks have no Unit specifications, and thus this parameter is set to zero. Hard disks are selected by Address 0 or 1. The proper address parameter is factory set and should never be changed.

Special Considerations for the 7906M Disk Drive

The 7906M disk 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 disk 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.



A cylinder is one track from surface 0, one track from surface 1, one track from surface 2, and one track from surface 3.

Figure C-4. 7906M Disk Drive

The 7906M disk 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 disk drive. Due to the physical structure of the disk 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 disk in shortened cylinder mode, use device type 34B (DT:34B in the DVT statement), and specify driver DDM30 (%DDM30).

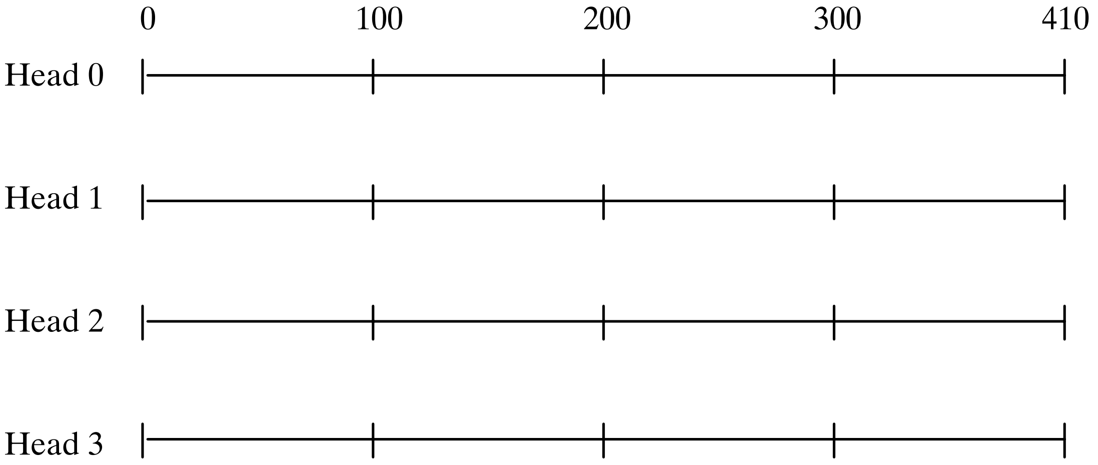
While up to 31 LUs may be defined for one 7906M disk drive, a good plan is to define no more than six. The total number of tracks on a 7906M is 1644. Of these, you should use approximately 1600 to make up your LUs and distribute the remaining tracks over the LUs as spares. The FORMT disk utility program lets you use spare tracks to replace bad tracks. FORMT is described in the RTE-A Utilities Reference manual.

Using the Worksheet

The 7906M Disk 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 Disk Configuration Worksheet Cylinders:

Cylinders:

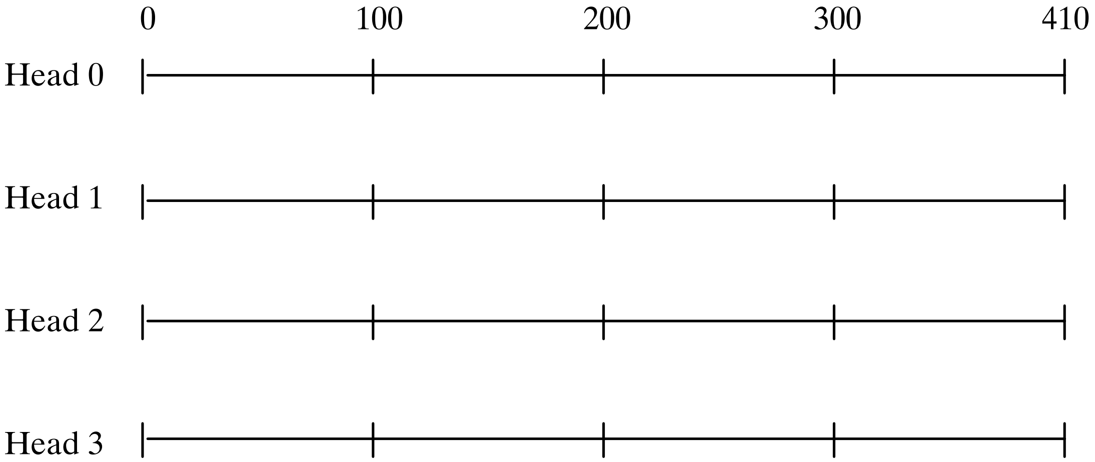


Total Tracks: 1644

	Disk 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									

7906 Disk Configuration Worksheet

Cylinders:



Total Tracks: 1644

	Disk 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) disk 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 disk 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 disk generation setting up three disks 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 disk, 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 disk as long as the disk is assigned as unit 0 and is configured in surface mode. Cylinder mode is the standard configuration for a 7920 or 7925, and the only mode supported for boot of these disks. Note, however, that it is perfectly valid to generate a 7925 in surface mode as an auxiliary disk.

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 DISKS       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 Disk Generator Answer File Statements

D

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 or gen 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 disk 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.

Example 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/1000 Reference Manual*, part number 92059-90001.)

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 (for example, $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 (for example, $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.

E

Device and Interface Driver Tables

This appendix contains information that describes supported disk drives, and tables that show the standard entries for the answer file (Table E-1), and the default device file values (Table E-2 through Table E-26). Note that the loaded driver size may vary, depending on your system requirements. For current driver sizes, refer to the online communicator in directory /HP1000_INFO.

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 name
DD.*nn*	Driver name in NAM statement
DD.*nn*	Driver entry point

CS/80 Configuration Scheme

The standard configuration for CS/80 disks is based on a building block scheme in which all CS/80 disks are subdivided into logical units of the same size. The smallest capacity disk, 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 disk (the 20-Mbyte Model 7907) uses all of the default values for LUs 16, 17, and 24, plus the values for its additional disk logical units. Each larger-capacity disk then uses all of the default values for the next smaller disk (including LU 24 if the CTD is a part of the disk), plus the file values for each additional disk logical unit. Appendix K contains the procedure for converting to this standard HP CS/80 disk configuration.

Table E-1. Standard Generation File Entries

Device		IFT Entry	DVT Entry	Node List
TERMINALS using Revision D Compatible Serial Interface [1]				
2382				
2392/93A				
2397A	(D-MUX)	IFT,ID800/01.REL,SC:*sc*	DVT,DDC00/01.REL,MHP_Term:*n*,LU:*lu* (*n* = port number, 0 to 7) (use DDC01.REL if	
262xA/P	(ASIC)	IFT,ID100/01.REL,SC:*sc*	DVT,DDC00/01.REL,MHP_Term:A,LU:*lu* slaved devices are	
2645	(A400 OBIO)	IFT,ID400.REL [2]	DVT,DDC00/01.REL,MHP_Term:*n*,LU:*lu* (*n* = port number, 0 to 3) connected to terminal)	
700/41				
700/9x				
2635	(D-MUX)	IFT,ID800/01.REL,SC:*sc*	DVT,DDC00.REL,MHP_2635:*n*,LU:*lu* (*n* = port number, 0 to 7)	
2635	(ASIC)	IFT,ID100/01.REL,SC:*sc*	DVT,DDC00.REL,MHP_2635:A,LU:*lu*	
2635	(A400 OBIO)	IFT,ID400.REL [2]	DVT,DDC00.REL,MHP_2635:*n*,LU:*lu* (*n* = port number, 0 to 3)	
non-HP	(D-MUX)	IFT,ID800/01.REL,SC:*sc*	DVT,DDC00.REL,MTerm:*n*,LU:*lu* (*n* = port number, 0 to 7)	
non-HP	(ASIC)	IFT,ID100/01.REL,SC:*sc*	DVT,DDC00.REL,MTerm:A,LU:*lu*	
non-HP	(A400 OBIO)	IFT,ID400.REL [2]	DVT,DDC00.REL,MTerm:*n*,LU:*lu* (*n* = port number, 0 to 3)	
TERMINALS using Revision C Compatible Serial Interface [1]				
2382	(C-MUX)	IFT,%IDM00,SC:*sc*	DVT,%DD*00,M26*xx*,LU:*lu*,DP:1:20004B,TX:57 [3]	
2382	(ASIC)	IFT,%ID*00/01,SC:*sc*	DVT,%DD*00,M26*xx*,LU:*lu* [3]	
2392/93A	(C-MUX)	IFT,%IDM00,SC:*sc*,TX:20	DVT,%DD*00,M26xx,LU:*lu*,DP:1:20004B,TX:57 [3]	
2392/93A	(ASIC)	IFT,%ID*00/01,SC:*sc*	DVT,%DD*00,M26xx,LU:*lu* [3]	
2397A	(C-MUX)	IFT,%IDM00,SC:*sc*,TX:20	DVT,%DD*00,M26xx,LU:*lu*,DP:1:20004B,TX:57 [3]	
2397A	(ASIC)	IFT,%ID*00/01,SC:*sc*	DVT,%DD*00,M26xx,LU:*lu* [3]	
262xA/P	(C-MUX)	IFT,%IDM00,SC:*sc*	DVT,%DD*00,M26xx,LU:*lu*,DP:1:20004B,TX:57 [3]	
262xA/P	(ASIC)	IFT,%ID*00/01,SC:*sc*	DVT,%DD*00,M26xx,LU:*lu* [3]	
2635	(C-MUX)	IFT,%IDM00,SC:*sc*	DVT,%DD*00,M2635:0/1(:1 = printer),LU:*lu*,DP:1:20004B,TX:57 [3]	*
2635	(ASIC)	IFT,%ID*00/01,SC:*sc*	DVT,%DD*00,M2635:0/1(:1 = printer),LU:*lu* [3]	*
2645	(C-MUX)	IFT,%IDM00,SC:*sc*	DVT,%DD*00,M2645,LU:*lu*,DP:1:20004B,TX:57 [3]	
2645	(ASIC)	IFT,%ID*00/01,SC:*sc*	DVT,%DD*00,M2645,LU:*lu* [3]	
700/41 [3]	(C-MUX)	IFT,%IDM00,SC:*sc*,TX:20	DVT,%DD*00,M26xx,LU:*lu*,DP:1:20004B,TX:57 [3]	
700/41 [3]	(ASIC)	IFT,%ID*00/01,SC:*sc*	DVT,%DD*00,M26xx,LU:*lu* [3]	
700/9x	(C-MUX)	IFT,%IDM00,SC:*sc*,TX:20	DVT,%DD*00,M26xx,LU:*lu*,DP:1:20004B,TX:57 [3]	
700/9x	(ASIC)	IFT,%ID*00/01,SC:*sc*	DVT,%DD*00,M26xx,LU:*lu* [3]	
TELNET Pseudo-Terminal LUs				
Terminals using ENQ/ACK		IFT,IDZ00.REL,SC:*sc*	DVT,DDC00.REL,MHP_Telnet,LU:*lu*	
Terminals using Xon/Xoff		IFT,IDZ00.REL,SC:*sc*	DVT,DDC00,REL,MTelnet,LU:*lu*	
CTUs using Revision D Compatible Serial Interface [1]				
264x CTU (MUX)		IFT,ID800/01.REL,SC:*sc*	DVT,DDC01.REL,MHP_Ctu:L/R,LU:*lu* (:L = left CTU, :R = right CTU)	*
264x CTU (ASIC)		IFT,ID100/01.REL,SC:*sc*	DVT,DDC01.REL,MHP_Ctu:L/R,LU:*lu* (:L = left CTU, :R = right CTU)	*
264x CTU (A400 OBIO)		IFT,ID400.REL [2]	DVT,DDC01.REL,MHP_Ctu:L/R,LU:*lu* (:L = left CTU, :R = right CTU)	*
CTUs using Revision C Compatible Serial Interface [1]				
264x CTU (MUX)		IFT,%IDM00,SC:*sc*	DVT,%DD*20,M264x:1/2(:1 = left CTU, :2 = right CTU),LU:*lu*	*
264x CTU (ASIC)		IFT,%ID*00/01,SC:*sc*	DVT,%DD*20,M264x:1/2(:1 = left CTU, :2 = right CTU),LU:*lu*	*

* Node list may be required for this device. Refer to the “Node Lists” section in this manual.

[1] Use the “01” version of the interface driver (for example, ID801.REL or %ID*01) for Modem support.

[2] The A400 OBIO select code is always 77B and should not be supplied in the IFT specification line; it is defaulted.

[3] The primary program defaults to FMGR and the secondary to COMND. To change these to CI and CM, append the following to the DVT entry:
,DP:5:CI:20040B:0,DP:9:CM:20040B:20040B:CM

Table E-1. Standard Generation File Entries (continued)

Device		IFT Entry	DVT Entry	Node List
PRINTERS using Revision D Compatible Serial Interface [1]				
2225D				
2227A				
2228A				
2235B/D				
2276A	(D-MUX)	IFT,ID800/01.REL,SC:*sc*	DVT,DDC00,MHP_Printer,LU:*lu* or DVT,DDC01,MHP_Slaved_Serial/HPIB,LU:*lu*	
256x	(ASIC)	IFT,ID100/01.REL,SC:*sc*	DVT,DDC00,MHP_Printer,LU:*lu* or DVT,DDC01,MHP_Slaved_Serial/HPIB,LU:*lu*	
2631	(OBIO)	IFT,ID400.REL [2]	DVT,DDC00,MHP_Printer,LU:*lu* or DVT,DDC01,MHP_Slaved_Serial/HPIB,LU:*lu*	
2686/7A				
2684D				
293x				
34400A				
PRINTERS using Revision C Compatible Serial Interface [1]				
2225D	(C-MUX)	IFT,%IDM00,SC:*sc*,TX:20	DVT,%DD*00,M2225,LU:*lu*,DP:1:2B,TX:57 [4]	
2225D	(ASIC)	IFT,%ID*00/01,SC:*sc*,TX:20	DVT,%DD*00,M2631,LU:*lu*	
2227A	(C-MUX)	IFT,%IDM00,SC:*sc*,TX:20	DVT,%DD*00,M2631,LU:*lu*,DP:1:2B,TX:57 [4]	
2227A	(ASIC)	IFT,%ID*00/01,SC:*sc*,TX:20	DVT,%DD*00,M2631,LU:*lu*	
2228A	(C-MUX)	IFT,%IDM00,SC:*sc*,TX:20	DVT,%DD*00,M2631,LU:*lu*,DP:1:2B,TX:57 [4]	
2228A	(ASIC)	IFT,%ID*00/01,SC:*sc*,TX:20	DVT,%DD*00,M2631,LU:*lu*	
2235D	(C-MUX)	IFT,%IDM00,SC:*sc*,TX:20	DVT,%DD*00,M2631,LU:*lu*,DP:1:2B,TX:57 [4]	
2235D	(ASIC)	IFT,%ID*00/01,SC:*sc*,TX:20	DVT,%DD*00,M2631,LU:*lu*	
2276A	(C-MUX)	IFT,%IDM00,SC:*sc*	DVT,%DD*00,M2631,LU:*lu*,DP:1:2B,TX:57 [4]	
2276A	(ASIC)	IFT,%ID*00/01,SC:*sc*	DVT,%DD*00,M2631,LU:*lu*	
256x	(C-MUX)	IFT,%IDM00,SC:*sc*	DVT,%DD*00,M256x,LU:*lu*,DP:1:2B,TX:57	
256x	(ASIC)	IFT,%ID*00/01,SC:*sc*	DVT,%DD*00,M256x,LU:*lu*	
2631	(C-MUX)	IFT,%IDM00,SC:*sc*	DVT,%DD*00,M2631,LU:*lu*,DP:1:2B,TX:57	
2631	(ASIC)	IFT,%ID*00/01,SC:*sc*	DVT,%DD*00,M2631,LU:*lu*	
2686/7A	(C-MUX)	IFT,%IDM00,SC:*sc*	DVT,%DD*00,M2631,LU:*lu*,DP:1:2B,TX:57 [4]	
2686/7A	(ASIC)	IFT,%ID*00/01,SC:*sc*	DVT,%DD*00,M2631,LU:*lu*	
2684D	(C-MUX)	IFT,%IDM00,SC:*sc*	DVT,%DD*00,M2631,LU:*lu*,DP:1:2B,TX:57 [4]	
2684D	(ASIC)	IFT,%ID*00/01,SC:*sc*	DVT,%DD*00,M2631,LU:*lu*	
293x	(C-MUX)	FT,%IDM00,SC:*sc*	DVT,%DD*00,M2631,LU:*lu*,DP:1:2B,TX:57 [4]	
293x	(ASIC)	IFT,%ID*00/01,SC:*sc*	DVT,%DD*00,M2631,LU:*lu*	
34400A	(C-MUX)	FT,%IDM00,SC:*sc*	DVT,%DD*00,M2631,LU:*lu*,DP:1:2B,TX:57 [4]	
34400A	(ASIC)	IFT,%ID*00/01,SC:*sc*	DVT,%DD*00,M2631,LU:*lu*	
PRINTERS using HP-IB Interface				
2235B/D	(HP-IB)	IFT,%ID*37,SC:*sc*	DVT,%DD*12,M2932A,LU:*lu*,DT:12B,DP:1:*HP-IBaddr*	
256x	(HP-IB)	IFT,%ID*37,SC:*sc*	DVT,%DDC12,,LU:*lu*,DP:1:*HP-IBaddr*	
2631	(HP-IB)	IFT,%ID*37,SC:*sc*	DVT,%DD*12,M2631,LU:*lu*,DP:1:*HP-IBaddr*	
2608S	(HP-IB)	IFT,%ID*37,SC:*sc*	DVT,%DDC12,M2608S,LU:*lu*,DP:1:*HP-IBaddr*	
293x	(HP-IB)	IFT,%ID*37,SC:*sc*	DVT,%DD*12,M2932,LU:*lu*,DP:1:*HP-IBaddr*	

[1] Use the “01” version of the driver (for example, ID801.REL or %ID*01) for Modem support.

[2] The A400 OBIO select code is always 77B and should not be supplied in the IFT specification line; it is defaulted.

[4] Setting DP1 to 2 disables ENQ/ACK and enables form feed.

Table E-1. Standard Generation File Entries (continued)

Device		IFT Entry	DVT Entry	Node List
PLOTTERS using Revision D Compatible Serial Interface [1]				
7440 7510A 7550 7570A 7585 7586 7595/6	(D-MUX)	IFT,ID800/01.REL,SC:*sc*	DVT,DDC00.REL,MPlotter:*n*,LU:*lu* (*n* = port number, 0 to 7)	
	(OBIO)	IFT,ID400.REL,SC:*sc* [2]	DVT,DDC00.REL,MPlotter:*n*,LU:*lu* (*n* = port number, 0 to 4)	
PLOTTERS using Revision C Compatible Serial Interface [1]				
7440	(C-MUX)	IFT,%IDM00,SC:*sc*,TX:20	DVT,%DD*00,M744x,LU:*lu*,DP:1:20004B,TX:57,TO:0	
7510A	(C-MUX)	IFT,%IDM00,SC:*sc*,TX:20	DVT,%DD*00,M755x,LU:*lu*,DP:1:20004B,TX:57,TO:0,	
7550	(C-MUX)	IFT,%IDM00,SC:*sc*,TX:20	DVT,%DD*00,M755x,LU:*lu*,DP:1:200004B,TX:57,TO:0	
7570A	(C-MUX)	IFT,%IDM00,SC:*sc*,TX:20	DVT,%DD*00,M755x,LU:*lu*,DP:1:20004B,TX:57,TO:0	
7585	(C-MUX)	IFT,%IDM00,SC:*sc*,TX:20	DVT,%DD*00,M758x,LU:*lu*,DP:1:20004B,TX:57,TO:0	
7586	(C-MUX)	IFT,%IDM00,SC:*sc*,TX:20	DVT,%DD*00,M758x,LU:*lu*,DP:1:20004B,TX:57,TO:0	
7595/6	(C-MUX)	IFT,%IDM00,SC:*sc*,TX:20	DVT,%DD*00,M755x,LU:*lu*,DP:1:20004B,TX:57,TO:0	
PLOTTERS using HP-IB Interface				
7440	(HP-IB)	IFT,%ID*37,SC:*sc*	DVT,,,LU:*lu*,TO:0,DT:77B,DX:1,DP:1:*HP-IBaddr*	
7510A	(HP-IB)	IFT,%ID*37,SC:*sc*	DVT,,,LU:*lu*,TO:0,DT:77B,DX:1,DP:1:*HP-IBaddr*	
7550	(HP-IB)	IFT,%ID*37,SC:*sc*	DVT,,,LU:*lu*,TO:0,DT:77B,DX:1,DP:1:*HP-IBaddr*	
7570A	(HP-IB)	IFT,%ID*37,SC:*sc*	DVT,,,LU:*lu*,TO:0,DT:77B,DX:1,DP:1:*HP-IBaddr*	
7585	(HP-IB)	IFT,%ID*37,SC:*sc*	DVT,,,LU:*lu*,TO:0,DT:77B,DX:1,DP:1:*HP-IBaddr*	
7586	(HP-IB)	IFT,%ID*37,SC:*sc*	DVT,,,LU:*lu*,TO:0,DT:77B,DX:1,DP:1:*HP-IBaddr*	
7595/6	(HP-IB)	IFT,%ID*37,SC:*sc*	DVT,,,LU:*lu*,TO:0,DT:77B,DX:1,DP:1*HP-IBaddr*	

[1] Use the “01” version of the interface driver (for example, ID801.REL or %IDM01) for Modem support.

[2] The A400 OBIO select code is always 77B and should not be supplied in the IFT specification line; it is defaulted.

Table E-1. Standard Generation File Entries (continued)

Device	IFT Entry	DVT Entry	Node List
DISK DRIVES using HP-IB Interface			
C2200 335MB Disk	IFT,%ID*37,SC:*sc*	DVT,%DD*33,Mc2200c:0/1/2/3/4/5/6,LU:*lu*,DP:1:*HP-IBaddr*	*
C2202 670MB Disk w/ cache	″	DVT,%DD*33,Mc2200c:0/1/2/3/4/5/6/7/8/9,LU:*lu*,DP:1:*HP-IBaddr*	*
C2203 670MB Disk	″	DVT,%DD*33,Mc2200c:0/1/2/3/4/5/6/7/8/9,LU:*lu*,DP:1:*HP-IBaddr*	*
12122A 20MB Fixed	IFT,%ID*37,SC:*sc*	DVT,%DD*33,M9133_CF:0/1/2,LU:*lu*,DP:1:*HP-IBaddr*	*
12122A Floppy	″	DVT,%DD*33,M_FLOPPY_CF:1,LU:*lu*,DP:1:*HP-IBaddr*	*
248x 10Mb Fixed	IFT,%ID*27,SC:*sc*	DVT,%GEN27,M2480:0/1/2,LU:*lu*	*
248x 15Mb Fixed	″	DVT,%GEN27,M2480:4/5/6/7,LU:*lu*	*
248x 20Mb Fixed	″	DVT,%GEN27,M2480:11/12,LU:*lu*	*
248x Floppy (SS)	″	DVT,%GEN27,M2480:3,LU:*lu*	*
248x Floppy (DS)	″	DVT,%GEN27,M2480:14,LU:*lu*	*
7902	IFT,%ID*37,SC:*sc*	DVT,%DD*30,M7902:0/1,LU:*lu* (:0 = left disk, :1 = right disk)	
7906H	IFT,%ID*37,SC:*sc*	DVT,%DD*30,M7906:0/1/2/3,LU:*lu*,DP:1:*HP-IBaddr*	*
7920H	IFT,%ID*37,SC:*sc*	DVT,%DD*30,M7920:0/1/.../8/9,LU:*lu*,DP:1:*HP-IBaddr*	*
7925H	IFT,%ID*37,SC:*sc*	DVT,%DD*30,M7925:0/1/.../7/8,LU:*lu*,DP:1:*HP-IBaddr*	*
7906M/S 20Mb (1 fix/1 rem)	IFT,%ID*37,SC:*sc*	DVT,%DDM30,M7906:0/1/2/3,LU:*lu*,DP:1:*HP-IBaddr*	*
7920M/S 50Mb (removable)	IFT,%ID*37,SC:*sc*	DVT,%DDM30,M7920:0/1/.../8/9,LU:*lu*,DP:1:*HP-IBaddr*	*
7925M/S 120Mb (removable)	IFT,%ID*37,SC:*sc*	DVT,%DDM30,M7925:0/1/.../7/8,LU:*lu*,DP:1:*HP-IBaddr*	*
7907A 41MB Fixed Disk	IFT,%ID*37,SC:*sc*	DVT,%DD*33,M7907FXD_CF:0/1/2,LU:*lu*,DP:1:*HP-IBaddr*	*
7907A 41MB Rem. Disk	″	DVT,%DD*33,M7907REM_CF:0/1/2,LU:*lu*,DP:1:*HP-IBaddr*	*
7908A 17MB Disk w/ CTD	″	DVT,%DD*33,M7908_CF:0/1,LU:*lu*,DP:1:*HP-IBaddr*	*
7911 28MB Disk w/ CTD	″	DVT,%DD*33,M7911_CF:0/1/2/3/4,LU:*lu*,DP:1:*HP-IBaddr*	*
7912 66MB Disk w/ CTD	″	DVT,%DD*33,M7912_CF:0/1/2/3/4/5/6,LU:*lu*,DP:1:*HP-IBaddr*	*
7914 132MB Disk w/ CTD	″	DVT,%DD*33,M7914_CF:0/1/2/3/4/5/6/7,LU:*lu*,DP:1:*HP-IBaddr*	*
7933A 404MB Fixed Disk	″	DVT,%DD*33,M7933_CF:0/1/2/3/4/5/6/7/8/9,LU:*lu*,DP:1:*HP-IBaddr*	*
7935A 404MB Rem. Disk	″	DVT,%DD*33,M7933_CF:0/1/2/3/4/5/6/7/8/9,LU:*lu*,DP:1:*HP-IBaddr*	*
7936A 307MB Disk	″	DVT,%DD*33,M7936_CF:0/1/2/3/4/5/6/7,LU:*lu*,DP:1:*HP-IBaddr*	*
7937A 571MB Disk	″	DVT,%DD*33,M7937_CF:0/1/2/3/4/5/6/7/8/9,LU:*lu*,DP:1:*HP-IBaddr*	*
7941A 24MB Disk	″	DVT,%DD*33,M7941_CF:0/1/2/3,LU:*lu*,DP:1:*HP-IBaddr*	*
7942A 24MB Disk w/ CTD	″	DVT,%DD*33,M7941_CF:0/1/2/3,LU:*lu*,DP:1:*HP-IBaddr*	*
7945A 55MB Disk	″	DVT,%DD*33,M7945_CF:0/1/2/3/4/5,LU:*lu*,DP:1:*HP-IBaddr*	*
7946A 55MB Disk w/ CTD	″	DVT,%DD*33,M7945_CF:0/1/2/3/4/5,LU:*lu*,DP:1:*HP-IBaddr*	*
7957A/B 81MB Disk	″	DVT,%DD*33,M7957_CF:0/1/2/3/4/5/6/7,LU:*lu*,DP:1:*HP-IBaddr*	*
7958A 130MB Disk	″	DVT,%DD*33,M7958_CF:0/1/2/3/4/5,LU:*lu*,DP:1:*HP-IBaddr*	*
7958B 152MB Disk	″	DVT,%DD*33,M7958B_CF:0/1/2/3/4/5/6,LU:*lu*,DP:1:*HP-IBaddr*	*
7959B 304MB Disk	″	DVT,%DD*33,M7959_CF:0/1/2/3/4/5/6/7,LU:*lu*,DP:1:*HP-IBaddr*	*
7962B 152MB Disk	″	DVT,%DD*33,M7958B_CF:0/1/2/3/4/5/6,LU:*lu*,DP:1:*HP-IBaddr*	*
7963B 304MB Disk	″	DVT,%DD*33,M7959_CF:0/1/2/3/4/5/6/7,LU:*lu*,DP:1:*HP-IBaddr*	*
9122D Dual 3-1/2" floppy	IFT,%ID*37,SC:*sc*	DVT,%DD*33,M_FLOPPY_CF:0/1,LU:*lu*,DP:1:*HP-IBaddr*	*
9133D/9134D	″	DVT,%DD*33,M9133_CF:0/1,LU:*lu*,DP:1:*HP-IBaddr*	*
9133H/9134H	″	DVT,%DD*33,M9133_CF:0/1/2,LU:*lu*,DP:1:*HP-IBaddr*	*
9133L/9134L/9153C w/ 40MB	″	DVT,%DD*33,M9133_CF:0/1/2/3,LU:*lu*,DP:1:*HP-IBaddr*	*
9133D/H/LIntegrated floppy	″	DVT,%DD*33,M_FLOPPY_CF:1,LU:*lu*,DP:1:*HP-IBaddr*	*
9133XV 15MB w/ floppy	″	DVT,%DD*30,M9134x:0/1/2/3,LU:*lu*,DP:1:*HP-IBaddr*	*
9133XV Integrated floppy	″	DVT,%DD*30,M9121:1,LU:*lu*,DP:1:*HP-IBaddr*	*
9134A 5-1/4" Disk	″	DVT,%DD*30,M9134:0/1/2/3,LU:*lu*,DP:1:*HP-IBaddr*	*
9153B/C 20MB w/ floppy	″	DVT,%DD*33,M9133_CF:0/1/2,LU:*lu*,DP:1:*HP-IBaddr*	*
9895A Flex. Mini Disks	″	DVT,%DD*30,M7902:0/1,LU:*lu* (Refer to the "Special Considerations for the 7902 Disk Drive" section in this manual.)	
DISK DRIVES using SCSI Interface			
See file DDQ30_GEN.MAC for generation information for SCSI disk drives.			

* Node list may be required for this device. Refer to the "Node Lists" section in this manual.

Table E-1. Standard Generation File Entries (continued)

Device	IFT Entry	DVT Entry	Node List
TAPE DRIVES using HP-IB Interface			
35401 CTD (autochanger)	IFT,ID*37,SC:*sc*	DVT,%DD*33,M35401,LU:*lu*,DP:1:*HP-IBaddr*	
9144/45 CTD (standalone)	″	DVT,%DD*33,M9144,LU:*lu*,DP:1:*HP-IBaddr*	
79xx CTD (integrated)	″	DVT,%DD*33,MTAPE,LU:*lu*,DP:1:*HP-IBaddr*	
7970E Mag Tape Drive	IFT,ID*37,SC:*sc*	DVT,%DD*23,M7970E:0/1/2/3,LU:*lu*,DP:1:*HP-IBaddr*	
7974A Mag Tape Drive	″	DVT,%DD*24,M7974,LU:*lu*,DP:1:*HP-IBaddr*:*density*	
7978A/B Mag Tape Drive	″	DVT,%DD*24,M7978,LU:*lu*,DP:1:*HP-IBaddr*:*density*	
7979/80 Mag Tape Drive	″	DVT,%DD*24,M7974/7978,LU:*lu*,DP:1:*HP-IBaddr*	
TAPE DRIVES using SCSI Interface			
See file DDQ24_GEN.MAC for generation information for SCSI tape drives.			
INTERFACES			
12005A/B ASIC (Rev C compat.)	IFT,%ID*00,SC:*sc*	(use DVT entry for associated device(s))	
12005A/B ASIC (Rev D compat.)	IFT,ID100/01.REL,SC:*sc*	(use DVT entry for associated device(s))	
12006A Parallel Interface	IFT,%ID*50,SC:*sc*	(use DVT entry for associated device(s)) ,DX:3	
12006A Parallel Interface	IFT,%ID*52,SC:*sc*	(use DVT entry for associated device(s))	
12009A HP-IB Interface [5]	IFT,%ID*37,SC:*sc*,– TX:130+4**n* [6]	(use DVT entry for associated device(s))	
12040A/B/C MUX Interface	IFT,%IDM00,SC:*sc*	(use DVT entry for associated device(s)) ,TX:57	
12040D MUX Interface	IFT,ID800/01.REL,SC:*sc*	(use DVT entry for associated device(s))	
12065A Video Interface	IFT,%ID*50,SC:*sc*	(use DVT entry for associated device(s)) ,DX:3	
12100A A400 OBIO	IFT,ID400.REL	(use DVT entry for associated device(s))	
12153 WCS	IFT,%ID*41,SC:*sc*	(use DVT entry for associated device(s))	
12205 Control Store	IFT,%ID*42,SC:*sc*	(use DVT entry for associated device(s))	
37222 MUX Interface	IFT,%IDM00,SC:*sc*	(use DVT entry for associated device(s)) ,TX:57	

[5] For high-speed devices, special cable length restrictions apply. Refer to the HP 12009A HP-IB Interface Reference Manual for a description of the high-speed data cable restrictions.

HP-IB address: The HP-IB driver, ID*37, requires that the HP-IB address be entered into driver parameter 1 (DP1) at generation time. The HP-IB address must be between 0 and 7 for disks, magnetic tape drives, and printers that use device drivers DD*12, DDC12, DD*33, DD*30, DDM30, DD*23, or DD*24.

[6] Where *n* is the number of HP-IB 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 134 words.

Table E-2. Terminal and Printer Default File Values

	CRT Terminal	Printing Terminal	Terminal CTU	ASIC Printer	HP-IB Printers	HP-IB Line Printers
Model Number	2621 2645 26XX	2635	2645:1 2645:2 2645X:1 2645X:2	2631	2631 2673A 2932/3/4A	2563 2564 2566 2608S
Entry Point	DD.00	DD.00	DD.20	DD.00	DD.12	DDC12
Device Type	5	0	20B	12B	12B	12B
Timeout	0	0	0	0	1000	500
Buffer Limits	100,400	100,400	NONE	100,400	NONE	NONE
Table Extension	32*	32*	45*	32*	36	98
Priority	0	0	0	0	0	NONE
Queuing	FIFO	FIFO	FIFO	FIFO		NONE
Number of Driver Parameters	12	12	1	12	2	8
Driver Parameter Area Defaults	DP1: 1 DP 2: 0 DP 3: 10400B DP 4: 0 DP 5: FM DP 6: GR DP 7: 20040B DP 8: 0 DP 9: CO DP 10: MN DP 11: D DP 12: 0	DP1: 3 DP 2: 0 DP 3: 10400B DP 4: 0 DP 5: FM DP 6: GR DP 7: 20040B DP 8: 0 DP 9: CO DP 10: MN DP 11: D DP 12: 0	DP 1:1 for models 2645:1 264X:1 DP 1:2 for models 2645:2 264X:2	DP1: 3 DP 2: 0 DP 3: 10400B DP 4: 0 DP 5: 0 DP 6: 0 DP 7: 0 DP 8: 0 DP 9: 0 DP 10: 0 DP 11: 0 DP 12: 0	DP 2: 1 (Should be set to 0 for the 2631 and 2673 printers only)	DP 2: 0 DP 8: 500

For HP-IB devices DP 1 = HP-IB addr (0−7) assigned at generation time.

* When generating device drivers DD*00 or DD*20 using MUX interface driver IDM00, you must override default DVT extensions and specify 57 as the number of table extensions (e.g., TX :57).

Table E-3. Flexible Disk Default File Values

	9895/7902 Disk	9895/7902 Disk
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-4. 9133A/9134A Disk (Four-Volume Format) Default File Values

	9133/9134 Disk			
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-5. 9133B/9134B Disk (Single-Volume Format) Default File Values

	9133/9134 Disk		
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 2: 0 DP 3: 0 DP 4: 0 DP 5: 0 DP 6: 472 DP 7: 31 DP 8: 4	DP 1: 0 (HP-IB Addr.) DP 2: 0 DP 3: 0 DP 4: 118 DP 5: 0 DP 6: 136 DP 7: 31 DP 8: 4	DP 1: 0 (HP-IB Addr.) DP 2: 0 DP 3: 0 DP 4: 152 DP 5: 0 DP 6: 612 DP 7: 31 DP 8: 4

Table E-6. 9133D/H/L 9134D/H/L Default File Values

	9133/9134 Disk			
Model Number	M9133_CF:0	M9133_CF:1	M9133_CF:2	M9133_CF:3
Entry Point	DD.33	DD.33	DD.33	DD.33
Device Type	33B	33B	33B	33B
Timeout	500	500	500	500
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: 604 DP 7: 48 DP 8: 0	DP 1: 0 DP 2: 0 DP 3: 0 DP 4: 0 DP 5: 28992 DP 6: 604 DP 7: 48 DP 8: 0	DP 1: 0 DP 2: 0 DP 3: 0 DP 4: 0 DP 5: 57984 DP 6: 413 DP 7: 48 DP 8: 0	DP 1: 0 DP 2: 0 DP 3: 0 DP 4: 1 DP 5: 12272 DP 6: 1622 DP 7: 48 DP 8: 0

9133D/9134D

9133H/9134H or 9153/9154

9133L/9134L

Table E-7. 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-8. 9122C/D Dual-Sided Microfloppy Default File Values

	9122C/D	
	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	700	700
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

* DP 6: 66 = Single-sided format
DP 6: 154 = Double-sided format

** DP 7: 16 = Double density
DP 7: 32 = High density

Table E-9. 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

* 9121 uses two surfaces in the generation. It is expected to have two surfaces genned in so that the driver does not have to be rewritten in order to be used with microfloppies. The even tracks are considered one surface and the odd tracks another.

Table E-10. Magnetic and CS/80 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 with a date code of 2401 or lower, use device type 23B instead of 24B.

Table E-11. 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-12. 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-13. 7920 Default File Values

	Cartridge 0	Cartridge 1	Cartridge 2	Cartridge 3	Cartridge 4
Model Number	M7920:0	M7920:1	M7920:2	M7920:3	M7920:4
Entry Point	DD.30	DD.30	DD.30	DD.30	DD.30
Device Type	32B	32B	32B	32B	32B
Timeout	3000	3000	3000	3000	3000
Buffer Limits	NONE	NONE	NONE	NONE	NONE
Table Extension	25	25	25	25	25
Number of Driver Parameters	8	8	8	8	8
Driver Parameter Area Defaults	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 3: 0	DP 3: 0	DP 3: 0	DP 3: 0	DP 3: 0
	DP 4: 0	DP 4: 83	DP 4: 166	DP 4: 249	DP 4: 332
	DP 5: 9	DP 5: 9	DP 5: 9	DP 5: 9	DP 5: 9
	DP 6: 406	DP 6: 406	DP 6: 406	DP 6: 406	DP 6: 406
	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

Cartridge 5	Cartridge 6	Cartridge 7	Cartridge 8	Cartridge 9
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 3: 0	DP 3: 0	DP 3: 0	DP 3: 0	DP 3: 0
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: 10
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 8: 5	DP 8: 5	DP 8: 5	DP 8: 5	DP 8: 5

Table E-14. 7920M Default File Values

	Cartridge 0	Cartridge 1	Cartridge 2	Cartridge 3	Cartridge 4
Model Number	M7920:0	M7920:1	M7920:2	M7920:3	M7920:4
Entry Point	DDM30	DDM30	DDM30	DDM30	DDM30
Device Type	32B	32B	32B	32B	32B
Timeout	3000	3000	3000	3000	3000
Buffer Limits	NONE	NONE	NONE	NONE	NONE
Table Extension	51	51	51	51	51
Number of Driver Parameters	8	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	DP 1: NONE DP 2: 0 DP 3: 0 DP 4: 332 DP 5: 9 DP 6: 406 DP 7: 48 DP 8: 5

Cartridge 5	Cartridge 6	Cartridge 7	Cartridge 8	Cartridge 9
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

Table E-15. 7925 Default File Values

	Cartridge 0	Cartridge 1	Cartridge 2	Cartridge 3	Cartridge 4
Model Number	M7925:0	M7925:1	M7925:2	M7925:3	M7925:4
Entry Point	DD.30	DD.30	DD.30	DD.30	DD.30
Device Type	32B	32B	32B	32B	32B
Timeout	3000	3000	3000	3000	3000
Buffer Limits	NONE	NONE	NONE	NONE	NONE
Table Extension	25	25	25	25	25
Number of Driver Parameters	8	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: 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	DP 1: NONE DP 2: 0 DP 3: 0 DP 4: 186 DP 5: 9 DP 6: 549 DP 7: 64 DP 8: 9	DP 1: NONE DP 2: 0 DP 3: 0 DP 4: 248 DP 5: 11 DP 6: 1024 DP 7: 64 DP 8: 9

Cartridge 5	Cartridge 6	Cartridge 7	Cartridge 8
DP 1: NONE DP 2: 0 DP 3: 0 DP 4: 363 DP 5: 11 DP 6: 1024 DP 7: 64 DP 8: 9	DP 1: NONE DP 2: 0 DP 3: 0 DP 4: 478 DP 5: 11 DP 6: 1024 DP 7: 64 DP 8: 9	DP 1: NONE DP 2: 0 DP 3: 0 DP 4: 593 DP 5: 11 DP 6: 1024 DP 7: 64 DP 8: 9	DP 1: NONE DP 2: 0 DP 3: 0 DP 4: 708 DP 5: 11 DP 6: 1024 DP 7: 64 DP 8: 9

Table E-16. 7925M Default File Values

	Cartridge 0	Cartridge 1	Cartridge 2	Cartridge 3	Cartridge 4
Model Number	M7925:0	M7925:1	M7925:2	M7925:3	M7925:4
Entry Point	DDM30	DDM30	DDM30	DDM30	DDM30
Device Type	32B	32B	32B	32B	32B
Timeout	3000	3000	3000	3000	3000
Buffer Limits	NONE	NONE	NONE	NONE	NONE
Table Extension	51	51	51	51	51
Number of Driver Parameters	8	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: 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	DP 1: NONE DP 2: 0 DP 3: 0 DP 4: 186 DP 5: 9 DP 6: 549 DP 7: 64 DP 8: 9	DP 1: NONE DP 2: 0 DP 3: 0 DP 4: 248 DP 5: 11 DP 6: 1024 DP 7: 64 DP 8: 9

Cartridge 5	Cartridge 6	Cartridge 7	Cartridge 8
DP 1: NONE DP 2: 0 DP 3: 0 DP 4: 363 DP 5: 11 DP 6: 1024 DP 7: 64 DP 8: 9	DP 1: NONE DP 2: 0 DP 3: 0 DP 4: 478 DP 5: 11 DP 6: 1024 DP 7: 64 DP 8: 9	DP 1: NONE DP 2: 0 DP 3: 0 DP 4: 593 DP 5: 11 DP 6: 1024 DP 7: 64 DP 8: 9	DP 1: NONE DP 2: 0 DP 3: 0 DP 4: 708 DP 5: 11 DP 6: 1024 DP 7: 64 DP 8: 9

Table E-17. 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 disks.

Table E-18. 7907A Removable Disk Default File Values

	Unit 1 Removable Disk		
	0	1	2
Model Number	M7907REM_CF:0	M7907REM_CF:1	M7907REM_CF:2
Entry Point	DD.33	DD.33	DD.33
Device Type	33B	33B	33B
Timeout	500	500	500
Buffer Limits	NONE	NONE	NONE
Table Extension	72	72	72
Priority	0	0	0
Queuing	FIFO	FIFO	FIFO
Number of Driver Parameters	8	8	8
Driver Parameter Area Defaults	DP 1: 0 DP 2: 400B DP 3: 0 DP 4: 0 DP 5: 0 DP 6: 400 DP 7: 48 DP 8: 0	DP 1: 0 DP 2: 400B DP 3: 0 DP 4: 0 DP 5: 19200 DP 6: 943 DP 7: 48 DP 8: 0	DP 1: 0 DP 2: 400B DP 3: 0 DP 4: 0 DP 5: 64750 DP 6: 320 DP 7: 48 DP 8: 0

Table E-19. 7907/08/11/12/14/33/41/42/45/46 Default File Values

Entry Point	DD.33	Priority	0
Device Type	33B	Queuing	FIFO
Buffer Limits	NONE	Number of Driver Parameters	8
Table Extension	72		

	Cartridge 0	Cartridge 1	Cartridge 2	Cartridge 3	Cartridge 4
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	M7911_CF:4 M7945_CF:4 M7912_CF:4 M7914_CF:4 M7933_CF:4
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: 14592 DP 6: 266 DP 7: 48 DP 8: 0	DP 1: 0 DP 2: 0 DP 3: 0 DP 4: 1 DP 5: 27392 DP 6: 352 DP 7: 48 DP 8: 0

7908
7907 (Fixed)
7941/7942
7911
7945/7946
7912
7914
7933

	Cartridge 5	Cartridge 6	Cartridge 7	Cartridge 8	Cartridge 9
	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
	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

7945/7946
7912
7914
7933

Table E-20. 248x Default File Values

Entry Point	none	Table Extension	0
Device Type	30B	Priority	0
Timeout	5000	Queuing	none
Buffer Limits	none	Number of Driver Parameters	9

	10-Mbyte Disk		
Model Number	2480:0	2480:1	2480:2
Driver Parameter Area Defaults	DP 1: 0	DP 1: 0	DP 1: 0
	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: 616
	DP 7: 31	DP 7: 31	DP 7: 31
	DP 8: 4	DP 8: 4	DP 8: 4
	DP 9: 0	DP 9: 0	DP 9: 0

	15-Mbyte Disk				
Model Number	2480:4	2480:5	2480:6	2480:7	2480:13
Driver Parameter Area Defaults	DP 1: 0	DP 1: 0	DP 1: 0	DP 1: 0	DP 1: 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 4: 0	DP 4: 104	DP 4: 159	DP 4: 236	DP 4: 104
	DP 5: 0	DP 5: 0	DP 5: 0	DP 5: 0	DP 5: 0
	DP 6: 624	DP 6: 330	DP 6: 462	DP 6: 420	DP 6: 1212
	DP 7: 31	DP 7: 31	DP 7: 31	DP 7: 31	DP 7: 31
	DP 8: 6	DP 8: 6	DP 8: 6	DP 8: 6	DP 8: 6
	DP 9: 0	DP 9: 0	DP 9: 0	DP 9: 0	DP 9: 0

	20-Mbyte Disk		Single-Sided Microfloppy	Double-Sided Microfloppy
Model Number	2480:11	2480:12	2480:3	2480:14
Driver Parameter Area Defaults	DP 1: 0	DP 1: 0	DP 1: 3	DP 1: 3
	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: 489	DP 4: 0	DP 4: 0
	DP 5: 0	DP 5: 0	DP 5: 0	DP 5: 0
	DP 6: 1956	DP 6: 476	DP 6: 66	DP 6: 154
	DP 7: 31	DP 7: 31	DP 7: 16	DP 7: 16
	DP 8: 4	DP 8: 4	DP 8: 1	DP 8: 2
	DP 9: 0	DP 9: 0	DP 9: 0	DP 9: 0

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-21. 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	130 (n * 4)	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.

Table E-22. 7936 Default File Values

	Cartridge 0	Cartridge 1	Cartridge 2	Cartridge 3
Model Number	M7936_CF:0	M7936_CF:1	M7936_CF:2	M7936_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
Number of Driver Area Defaults	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: 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

Cartridge 4	Cartridge 5	Cartridge 6	Cartridge 7
M7936_CF:4	M7936_CF:5	M7936_CF:6	M7936_CF:7
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

Table E-23. 7937 Default File Values

	Cartridge 0	Cartridge 1	Cartridge 2	Cartridge 3	Cartridge 4	Cartridge 5	Cartridge 6	Cartridge 7	Cartridge 8	Cartridge 9
Model Number	M7937_CF:0	M7937_CF:1	M7937_CF:2	M7937_CF:3	M7937_CF:4	M7937_CF:5	M7937_CF:6	M7937_CF:7	M7937_CF:8	M7937_CF:9
Entry Point	DD.33	DD.33	DD.33	DD.33	DD.33					
Device Type	33B	33B	33B	33B	33B					
Buffer Limits	NONE	NONE	NONE	NONE	NONE					
Table Extension	72	72	72	72	72					
Number of Driver Area Defaults	8	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: 4160 DP 7: 64 DP 8: 0	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

Table E-24. 7957 Default File Values

	Cartridge 0	Cartridge 1	Cartridge 2	Cartridge 3	Cartridge 4	Cartridge 5	Cartridge 6	Cartridge 7
Model Number	M7957_CF:0	M7957_CF:1	M7957_CF:2	M7957_CF:3	M7957_CF:4	M7957_CF:5	M7957_CF:6	M7957_CF:7
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				
Number of Driver Area Defaults	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: 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

Table E-25. 7958A/B Default File Values

	Cartridge 0	Cartridge 1	Cartridge 2	Cartridge 3
Model Number	M7958_CF:0 M7958B_CF:0	M7958_CF:1 M7958B_CF:1	M7958_CF:2 M7958B_CF:2	M7958_CF:3 M7958B_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
Number of Driver Area Defaults	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: 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

7958A
7958B

Cartridge 4	Cartridge 5	Cartridge 6
M7958_CF:4 M7958B_CF:4	M7958_CF:5 M7958B_CF:5	M7958B_CF:6
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: 4 DP 5: 48366 DP 6: 3125 DP 7: 64 DP 8: 0	DP 1: 0 DP 2: 0 DP 3: 0 DP 4: 7 DP 5: 51758 DP 6: 1307 DP 7: 64 DP 8: 0

7958A

7958B

Table E-26. 7959/7963 Default File Values

	Cartridge 0	Cartridge 1	Cartridge 2	Cartridge 3
Model Number	M7959:0	M7959:1	M7959:2	M7959:3
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
Number of Driver Area Defaults	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: 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

Cartridge 4	Cartridge 5	Cartridge 6	Cartridge 7
M7959:4	M7959:5	M7959:6	M7959:7
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: 4757 DP 7: 64 DP 8: 0

F

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.

DD*00

Entry Point	TX	TO	QU	BL
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									

DD*12

Entry Point	TX	TO
DD.12	36	1000

Model	DT	DX	DP
2631	12B	2	2:0
2673A			2:0
2697B			2:1
2932A			2:1

DD*20

Entry Point	TX	TO	QU	DT	DX
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

DD*23

Entry Point	TX	TO	DT	DX
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

DD*24

Entry Point	TX	DX	QU	BL	PR	TO	DT
DD.24	43	5	FI	UN,0,0	0	500	24B

Model	DP	
7974	1:4	1600
7978	1:4	1600

DD*30

Entry Point	TX	DX	QU	BL	PR
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

DD*33

Entry Point	TX	DX	QU	BL	PR	TO
DD.33	72	8	FI	UN,0,0	0	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

DD*33 (continued)

Entry Point	TX	DX	QU	BL	PR	TO
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

DD*33 (continued)

Entry Point	TX	DX	QU	BL	PR	TO
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	214	48	0
7933:2	33B	2:0	0	0	29472	300	48	0
7933:3	33B	2:0	0	0	43872	214	48	0
7933:4	33B	2:0	0	0	54144	215	48	0
7933:5	33B	2:0	0	0	64750	400	48	0
7933:6	33B	2:0	0	1	18414	539	48	0
7933:7	33B	2:0	0	1	44288	424	48	0
7933:8	33B	2:0	0	1	64640	340	48	0
7933:9	33B	2:0	0	2	15424	764	48	0
7933:10	33B	2:0	0	2	52096	764	48	0
7933:11	33B	2:0	0	3	23232	758	48	0
7933:12	33B	2:0	0	3	59616	902	48	0
7933:13	33B	2:0	0	4	37376	902	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

DD*33 (continued)

Entry Point	TX	DX	QU	BL	PR	TO
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		0	23	1312	1477	48	0

DD*33 (continued)

Entry Point	TX	DX	QU	BL	PR	TO
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

DD*33 (continued)

Entry Point	TX	DX	QU	BL	PR	TO
DD.33	72	8	FI	UN,0,0	0	500

Model	DT	DP							
7958B_CF:0	33B	2:0	0	0	0		400	48	0
7958B¯CF:1	33B	2:0	0	0	19200		943	48	0
7958B¯CF:2	33B	2:0	0	0	64750		640	64	0
7958B¯CF:3	33B	2:0	0	1	40174		640	64	0
7958B¯CF:4	33B	2:0	0	2	15598		2560	64	0
7958B¯CF:5	33B	2:0	0	4	48366		3125	64	0
7958B¯CF:6	33B	2:0	0	7	51758		1307	64	0
7959:0	33B	2:0	0	0	0		400	48	0
7959:1	33B	0	0	0	19200		943	48	0
7959:2	33B	0	0	0	64750		2560	64	0
7959:3	33B	0	0	3	31982		2560	64	0
7959:4	33B	0	0	5	64750		2560	64	0
7959:5	33B	0	0	8	31982		2560	64	0
7959:6	33B	0	0	10	64750		2560	64	0
7959:7	33B	0	0	13	31982		4757	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
9133¯CF:3	33B	2:0	0	1	12272		1622	48	0
* *model C2200 335MB Disk Drives*									
c2200c:0	33B	4:0	0	400	48				
c2200c:1	33B	4:0	19200	943	48				
c2200c:2	33B	4:0	64464	4096	64				
c2200c:3	33B	4:4	64464	4096	64				
c2200c:4	33B	4:8	64464	4096	64				
c2200c:5	33B	4:12	64464	4096	64				
c2200c:6	33B	4:16	64464	3075	64				
* *model C2202/3 670MB Disk Drives*									
c2202c:0	33B	4:0	0	400	48				
c2202c:1	33B	4:0	19200	943	48				
c2202c:2	33B	4:0	64464	4096	64				
c2202c:3	33B	4:4	64464	4096	64				
c2202c:4	33B	4:8	64464	4096	64				
c2202c:5	33B	4:12	64464	4096	64				
c2202c:6	33B	4:16	64464	4096	64				
c2202c:7	33B	4:20	64464	4096	64				
c2202c:8	33B	4:24	64464	8192	64				
c2202c:9	33B	4:32	64464	7159	64				
* *model 9122*									
M_FLOPPY_CF:0	33B		0	0	0	0	154	16	0
* *model 9122/9133*									
M_FLOPPY_CF:1	33B		400B	0	0	0	154	16	0

DD*36

Entry	TX	DX	QU	BL	PR
DD.36	25	8	–	–	–

Model	DT	DP						
500	36B	2:0	0	0	0	64	4	1

DDC00

Entry Point	TX	DX	QU	BL	PR
DDC00	11	4	–	BU,40,400	–

Model	DT	DP				
MHP_Term:A	05	1:2	0	0b	100001b	
MHP_Term:S	05	1:2	0	0b	100001b	TX:12
MHP_Telnet	05	1:2	0	20000b	100001b	TX:16
MHP_Term:0	05	1:2	0	170b	100001b	
MHP_Term:1	05	1:2	0	171b	100001b	
MHP_Term:2	05	1:2	0	172b	100001b	
MHP_Term:3	05	1:2	0	173b	100001b	
MHP_Term:4	05	1:2	0	174b	100001b	
MHP_Term:5	05	1:2	0	175b	100001b	
MHP_Term:6	05	1:2	0	176b	100001b	
MHP_Term:7	05	1:2	0	177b	100001b	
MTerm:A	0	1:0	0	0	100001b	
MTerm:S	0	1:0	0	0	100001b	TX:12
MTelnet	0	1:1	0	20000b	100001b	TX:16
MTerm:0	0	1:1	0	170b	100001b	
MTerm:1	0	1:1	0	171b	100001b	
MTerm:2	0	1:1	0	172b	100001b	
MTerm:3	0	1:1	0	173b	100001b	
MTerm:4	0	1:1	0	174b	100001b	
MTerm:5	0	1:1	0	175b	100001b	
MTerm:6	0	1:1	0	176b	100001b	
MTerm:7	0	1:1	0	177b	100001b	
MHP48000:A	0	1:44b	0	0	0	
MHP_Printer:A	12b	1:4402b	0	0	0	
MHP_Printer:S	12b	1:4402b	0	0	0	TX:12
MHP_Printer:0	12b	1:4402b	0	130b	0	
MHP_Printer:1	12b	1:4402b	0	131b	0	
MHP_Printer:2	12b	1:4402b	0	132b	0	
MHP_Printer:3	12b	1:4402b	0	133b	0	
MHP_Printer:4	12b	1:4402b	0	134b	0	
MHP_Printer:5	12b	1:4402b	0	135b	0	
MHP_Printer:6	12b	1:4402b	0	136b	0	
MHP_Printer:7	12b	1:4402b	0	137b	0	

DDC00 (continued)

Entry Point	TX	DX	QU	BL	PR
DDC00	11	4	–	BU,40,400	–

Model	DT	DP				
MPrinter:A	12b	1:4401b	0	0	0	
MPrinter:S	12b	1:4401b	0	0	0	TX:12
MPrinter:0	12b	1:4401b	0	130b	0	
MPrinter:1	12b	1:4401b	0	131b	0	
MPrinter:2	12b	1:4401b	0	132b	0	
MPrinter:3	12b	1:4401b	0	133b	0	
MPrinter:4	12b	1:4401b	0	134b	0	
MPrinter:5	12b	1:4401b	0	135b	0	
MPrinter:6	12b	1:4401b	0	136b	0	
MPrinter:7	12b	1:4401b	0	137b	0	
MPlotter:S	0	1:1	0	0		TX:12
MPlotter:0	0	1:1	0	130b	0	
MPlotter:1	0	1:1	0	131b	0	
MPlotter:2	0	1:1	0	132b	0	
MPlotter:3	0	1:1	0	133b	0	
MPlotter:4	0	1:1	0	134b	0	
MPlotter:5	0	1:1	0	135b	0	
MPlotter:6	0	1:1	0	136b	0	
MPlotter:7	0	1:1	0	137b	0	
MHP_2635:A	06	1:4002b	0	0	100001b	
MHP_2635:S	06	1:4002b	0	0	100001b	TX:12
MHP_2635:0	06	1:4002b	0	170b	100001b	
MHP_2635:1	06	1:4002b	0	171b	100001b	
MHP_2635:2	06	1:4002b	0	172b	100001b	
MHP_2635:3	06	1:4002b	0	173b	100001b	
MHP_2635:4	06	1:4002b	0	174b	100001b	
MHP_2635:5	06	1:4002b	0	175b	100001b	
MHP_2635:6	06	1:4002b	0	176b	100001b	
MHP_2635:7	06	1:4002b	0	177b	100001b	
MQTD_Port7	00	1:100004b	0	77b	0	
MHP_Ctu:L	20b	4:1 BL:UN:0:0				
MHP_Ctu:R	20b	4:2 BL:UN:0:0				
MHP_Slaved_Serial	12b	4:4 BL:UN:0:0				
MHP_Slaved_HPIB	12b	4:5 BL:UN:0:0				
MHP_Internal_Prtr	12b	4:4 BL:UN:0:0				

DDC12

Entry Point	TX	DX	QU	BL	PR
DDC12	98	8	–	–	–

Model	DT	DP						
500	12B	2:1	0	0	0	0	0	500
2608S		8:500						
2563		8:500						
2566		8:500						

DDQ30

The DVT entries for DDQ30 are given in source file DDQ30_GEN.MAC. You can list this file to see the default DVT entries (generation records) available.

DDM30

Entry Point	TX	DX	QU	BL	PR
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

GEN27

Entry Point	TX	DX	QU	BL	PR	TO	DT
–	–	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

G

Example System Generation Answer File

The example 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-IV (HP 91750A) and VC+ (HP 92078A) are also specified.

The example answer file can be used as the basis for a new system generation. The file is available on your Primary System under the name PRIMARY.ANS. Always use the most current version. 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: 92077-17326  REV.6200 <950406.1028>
*
* Primary.ans- RTE-A Primary System Generation Answer File to make
* primary.sys(#92077-16954), primary.snp (#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 file, and continue with     *
*     *   the instructions on the second half.                              *
*     *                                                                     *
*     ***********************************************************************
*
* 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:
*     LU 1:   ID100 at select code 20B (double-mapped if VCP port)
*     LU 110: ID400 at select code 77B (double-mapped if VCP port)
*     LU 120: ID800 at select code 30B (double-mapped if VCP port)
*     LU 130: IDM00 at select code 23B (double-mapped if VCP port)
*
* Note that if mapping occurs, do not use the previous LU.  (If 12100A is
*   the VCP port, LU 110 should not be used, since there are now two LUs
*   pointing to the same hardware.)
*
*$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$*
```

```
*                                                                           *
*                              DISCLAIMER                                   *
*                              ==========                                   *
*                                                                           *
*    The Primary System is designed to be extremely flexible and serves     *
* two main purposes.  First, it is used as a verification tool for all      *
* supported peripherals on the A-Series.  Second, it is used to generate    *
* a customized system for the particular needs of the customer.             *
*                                                                           *
*    To provide maximum flexibility the Primary System is NOT necessarily   *
* the configuration that HP recommends for the final system generation.     *
* Depending on the mix and cabling of the peripherals connected, the        *
* possibility exists of configuring the system in an unsupported manner,    *
* which can adversely impact system operation and performance.  It is the   *
* responsibility of the user to be aware of these limitations and not       *
* violate the maximum number or mix of devices on a given interface to      *
* avoid the possibility of data corruption or diminished system performance.*
*                                                                           *
*    For support and configuration information, contact your local sales    *
* representative or customer engineer for the information.                  *
*                                                                           *
*$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$*
*
*                     I/O    D E F I N I T I O N S
*
*  LU         DEVICE                         SELECT CODE        ADDRESS
* ----       --------                       -------------      ---------
*                                SCSI
*                                =====
*  44     SCSI DAT tape                          25B               3
*  35     SCSI 7980S                             25B               3
* 20-21   SCSI hard disk                         25B               6
* 22-23   SCSI 650A MO disk                      25B               5
*  60     SCSI floppy single sided               25B               0
*  61     SCSI floppy doubled sided              25B               0
*                                HP-IB
*                                =====
*  25     HP-IB controller                       26B              36B
* 26-27   HP-IB disk                             26B               2
*  40     HP-IB disk 20mb  (overlay)             26B               2
*  41     CS-80 flexible disk single sided       26B               2
*  42     CS-80 flexible disk doubled sided      26B               2
*  54     CS-80 compatible cartridge tape cache  26B               2
* 28-29   2nd HP-IB disk                         26B               0
*  39     9144/5 standalone CS-80 cartridge tape 26B               1
*  37     7974A/7978A streaming tape drive       26B               3
*  38     7970 HP-IB tape drive                  26B               4
*  36     2932A line printer                     26B               7
*
*  85     2608S,2563A,2566A line printer         26B               7
*
*                              SCSI/HP-IB
*                              ==========
*  14     Reserved for SCSI DAT tape             27B               3
*   5     Reserved for SCSI 7980S                27B               3
* 10-11   Reserved for SCSI hard disk            27B               6
* 12-13   Reserved for SCSI MO                   27B               5
* 50-51   Reserved for SCSI floppy               27B               0
```

```
*
*  15     Reserved for HP-IB controller          27B               36B
* 16-17   Reserved for HP-IB hard disk           27B                2
*  30     Reserved for HP-IB hard disk           27B                2
*  31     Reserved for single sided floppy       27B                2
*  32     Reserved for doubled sided floppy      27B                2
*  24     Reserved for CS/80 cartridge tape      27B                2
* 18-19   Reserved for 2nd HP-IB disk            27B                0
*   9     Reserved for 9144/45 cartridge tape    27B                1
*   7     Reserved for 7974/78 streaming tape    27B                3
*   8     Reserved for 7970 tape                 27B                4
*   6     Reserved for 2932A line printer        27B                7
*
*                              TERMINALS
*                              =========
* 100     12005 ASIC #1                          20B
* 110-113 12100 A400 OBIO 4-channel MUX          77B
* 120-127 12040 Rev D 8-channel MUX              30B
* 130-137 12040 Rev A,B,C 8-channel MUX          23B
*
*  You can load the Primary system from either a SCSI or an HP-IB disk on
*  select code 27B.  After the Primary system is loaded, the select codes
*  25B through 27B look like:
*
*  SC  ADDR  LU      SCSI system      HP-IB system
*  --- ---- -----  ----------------- ----------------
*  25b  5   22-23     --               SCSI MO disk
*       3   44/35     --               SCSI DAT/7980S
*       6   20-21     --               SCSI hard disk
*       0   60-61     --               SCSI floppy
*
*  26b  2  26-27/40   HP-IB disk       --
*       2    41       floppy (single)  --
*       2    42       floppy (doubled) --
*       2    54       HP-IB cartridge  --
*       0   18-19     HP-IB disk       --
*       1    39       9144/45 cartridge--
*       3    37       7974/78 tape     --
*       4    38       7970 tape        --
*       7   36/85     printer          --
*      36b   25       HP-IB controller --
*
*  27b  2   16-17/30  --               HP-IB disk
*       2    31       --               floppy (single)
*       2    32       --               floppy (doubled)
*       2    24       --               HP-IB cartridge
*       0   18-19     --               2nd HP-IB disk
*       1     9       --               9144 cartridge
*       3   7/14/5    DAT/7980S        7974/78 tape
*       4     8       --               7970 tape/DAT
*       5   12-13     SCSI MO disk     --
*       6   10-11     SCSI hard disk   --
*       0   50-51     SCSI floppy      --
*       7     6       --               2392A printer
*      36b   15       --               HP-IB controller
*
**********************************************************************
links,cp,,                Use current page links
```

```
* =======================
* System Relocation Phase
* =======================
le,off,                   Do not list the module entry points
er,,                      Echo errors to the terminal
re %vctr:92570,,          Entry points
tg 700                    Number of tags required
re %mapos:92077,,         Partitioned OS tag routines
re %rpl70:92077,,         No CDS no double precision floating point
re %exec:92077,,          EXEC request processing
re %rtioa:92077,,         Real–Time I/O control
re %maps:92570,,          Dynamic mapping system routines
re %progs:92570,,         Program state processing
re %util:92570,,          System variables and utilities
re %sam:92077,,           System available memory
re %sched:92570,,         Programmatic program scheduling
re %strng:92077,,         Runstring passing
re %erlog:92570,,         Error Logging
re %opmsg:92077,,         Operating system messages
re %sycom:92570,,         Operator commands
re %iomod:92570,,         I/O module
re %id*43:92077,,         Power fail driver
re %signl:92570,,         Signals module
re %$idrpl:92570,,        System ID dup IDRPL
*
se $syslb:92570,,         Search the system library
*
* define partitionable modules
*
pa perr,xcmnd,stat,dsq,vema,lock,load,memry,iorq
pa time,class,abort,alarm
*
ms $sysa:92570,,          Search the system dummy library
*
end,                      End system relocation phase
* ================================
* OS module/Driver partition Phase
* ================================
*
re %load:92570,,          Program loading and swapping
end
*
re %memry:92570,,         Memory management module
end
*
re %iorq:92570,,          I/O request processing
end
*
re %lock:92570,,          LU locking and resource numbers
re %time:92077,,          Time scheduling of programs
end
*
re %class:92570,,         Class I/O module
end
*
re %xcmnd:92570,,         Operator command extension module
re %stat:92077,,          Status command module
re %dsq:92570,,           NS/1000 module
```

```
end
*
re %perr:92570,,          Parity error handler
re %vema:92570,,          Virtual memory module
end
*
re %alarm:92077,,         Timer signal processing module
end
*
re %abort:92570,,         Abort processing module
end
*
* Driver partitions
*
re ddq30.rel:92077,,      Device driver for SCSI disks
re id100.rel:92077,,      Interface driver for 12005 ASIC
end
*
re ddq24.rel:92077,,      Device driver for SCSI DAT tape
re %dd*24:92077,,         Device driver for 7974/78 straming tape
end
*
re idq35.rel:92077,,      SCSI interface drvier
end
*
re %dd*33:92077,,         Device driver for CS/80 disks
end
*
re %id*37:92077,,         Interface driver for CS/80 disks
re,pri27.rel:92077,,      Driver for the Primary system only
end
*
re id400.rel:92077,,      Interface driver for A400 OBIO
end
*
re %idm00:92077,,         Interface driver for 12040 A/B/C MUX
re %dd*23:92077,,         Device driver for magnetic tape
end
*
re ddc00.rel:92077,,      Device driver for 100 series interfaces
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 %ddc12:92077,,         Device driver for HP–IB line printer
end
*
re %dd*00:92077,,         Device driver for A/B/C MUX
end
*
end,,                     End OS module/Driver partition phase
* ======================
* Table Generation Phase
* ======================
*
* SCSI –– hard disk, MO and DAT tape                Select Code = 25b
```

```
*
ift,idq35.rel:92077,SC:25B
*
*  SCSI hard disk                 SCSI address 6,     LU 20–21
*
dvt,ddq30_gen.rel:92077, m64mb:0, lu:20 dp:1:6 dp:8:1
dvt,ddq30_gen.rel:92077, m64mb:1, lu:21 dp:1:6 dp:8:1
*
*  SCSI 650A MO disk              SCSI address 5,     LU 22–23
*
dvt,ddq30_gen.rel:92077, m64mb:0, lu:22 dp:1:5 dp:8:3
dvt,ddq30_gen.rel:92077, m64mb:1, lu:23 dp:1:5 dp:8:3
*
*  SCSI DAT/7980S                 SCSI address 3,     LU 44/35
*
dvt,ddq24_gen.rel:92077,,lu:44,dp:1:3
dvt,ddq24_gen.rel:92077,M7980,lu:35,dp:1:3
*
* SCSI floppy  single sided                   LU 60
*
dvt,ddq30_gen.rel:92077,m16mb:0,lu:60,dp:1:0:0:0:0:0,–
  dp:6:66:16:0
*
* SCSI floppy  double sided                   LU 61
*
dvt,ddq30_gen.rel:92077,m16mb:0,lu:61,dp:1:0:0:0:0:0,–
  dp:6:154:16:0
*
* HP–IB  –– disks, magnetic tape and printer        Select Code = 26b
*
ift,%id*37:92077,SC:26B
*
* Bus Controller LU               HP–IB address 36b,  LU 25
*
dvt,,,LU:25,TO:2000,DT:37B,TX:0,DX:1,DP:1:36B,PR:0
*
* HP–IB disk 64MB/20MB            HP–IB address 2,    LU 26–27/40
*
dvt,%dd*33:92077 m7908_cf:0 lu:26 dp:1:2 dp:2:0:0:00:0:4096 dp:7:64
dvt,%dd*33:92077 m7908_cf:0 lu:27 dp:1:2 dp:2:0:0:04:0:4096 dp:7:64

dvt,%dd*33:92077 m7908_cf:0 lu:40 dp:1:2 dp:2:0:0:00:0:1215 dp:7:64
*
*  Flexible disk single/doubled sided     HP–IB address 2,    LU 41/42
*
dvt,%dd*33:92077 m_floppy_cf:1 lu:41 dp:1:2:400b:0:0:0,–
  dp:6:66:16:0
dvt,%dd*33:92077,m_floppy_cf:1,lu:42,dp:1:2
*
* Cartridge tape with cache    HP–IB address 2,    LU 54
*
dvt,%dd*33:92077,mtape,lu:54,dp:1:2 dp:5:768
*
* 2nd HP–IB disk                  HP–IB address 0,    LU 28–29
*
dvt,%dd*33:92077 m7908_cf:0 lu:28 dp:1:2 dp:2:0:0:00:0:4096 dp:7:64
dvt,%dd*33:92077 m7908_cf:0 lu:29 dp:1:2 dp:2:0:0:04:0:4096 dp:7:64
*
```

```
* 9144/45 CS–80 cartridge tape    HP–IB address 1,    LU 39
*
dvt,%dd*33:92077,M9144:0,LU:39,DP:1:1
*
* 7974/78 streaming tape drive     HP–IB address 3,   LU 37
*
dvt,%dd*24:92077,M7974:0,LU:37,DP:1:3
*
* 7970 tape drive                  HP–IB address 4,   LU 38
*
dvt,%dd*23:92077,M7970E:0,LU:38,DP:1:4,PR:1
*
* 2932A Line Printer               HP–IB address 7,   LU 36
*
dvt,%dd*12:92077,M2932A,LU:36,DT:12B,DP:1:7
*
*    2608S,2563A,2566A line printer          HP–IB address = 7  LU 85
*
dvt,%ddc12:92077,,lu:85,dp:1:7
*
*
* Primary driver for HP–IB/SCSI                     Select Code = 27b
*
ift,%id*37:92077 sc:27b
dvt,pri27.rel:92077,,lu:15 dp:1:36b
dvt,pri27.rel:92077,,lu:16 dp:1:2 dt:33b dp:2:0:0:00:0:4096 dp:7:64
dvt,pri27.rel:92077,,lu:17 dp:1:2 dt:33b dp:2:0:0:04:0:4096 dp:7:64
dvt,pri27.rel:92077,,lu:30 dp:1:2 dt:33b dp:2:0:0:00:0:1215 dp:7:64
dvt,pri27.rel:92077,,lu:31 dp:1:2 dt:33b dp:2:400b:0:0:0,dp:6:66:16:0
dvt,pri27.rel:92077,,lu:32 dp:1:2 dt:33b dp:2:400B:0:0:0:154,dp:7:16:0
dvt,pri27.rel:92077,,lu:24 dp:1:2 dt:26b
dvt,pri27.rel:92077,,lu:9  dp:1:1 dt:26b
dvt,pri27.rel:92077,,lu:18 dp:1:0 dt:33b dp:2:0:0:00:0:4096 dp:7:64
dvt,pri27.rel:92077,,lu:19 dp:1:0 dt:33b dp:2:0:0:04:0:4096 dp:7:64
dvt,pri27.rel:92077,,lu:7  dp:1:3 dt:24b
dvt,pri27.rel:92077,,lu:8  dp:1:4 dt:23b

dvt,pri27.rel:92077,,lu:14 dp:1:3 dt:24b
dvt,pri27.rel:92077,,lu:5  dp:1:3 dt:24b
dvt,pri27.rel:92077,,lu:12 dp:1:5 dt:30b
dvt,pri27.rel:92077,,lu:13 dp:1:5 dt:30b
dvt,pri27.rel:92077,,lu:10 dp:1:6 dt:30b
dvt,pri27.rel:92077,,lu:11 dp:1:6 dt:30b
dvt,pri27.rel:92077,,lu:50 dp:1:0 dt:30b
dvt,pri27.rel:92077,,lu:51 dp:1:0 dt:30b

dvt,pri27.rel:92077,,lu:6  dp:1:7
*
*  Terminal LUs
*
* NOTE: There is no LU defined as LU 1.  This points to the bit bucket.
*
* ASIC #1  Terminal                                 Select Code = 20B
*                                                     LU 100
ift,id100.rel:92077,sc:20B
dvt,ddc00.rel:92077,MHP_Term:A,lu:100
*
*   12100A A400 OBIO 4–channel MUX ports A–D        Select Code = 77B
*                                                     LU 110–113
*
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 Revision D 8–channel MUX port 0–7        Select Code = 30B
*                                                     LU 120–127
*
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
*
* 12040 A–C 8–channel MUX port 0–7 select code = 23B
*
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:1:20040B,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
* =================
node,16,17,30,31,32,24
node,26,27,40,41,42,54
node,18,19
node,28,29
node,10,11
node,20,21
node,12,13
node,22,23
node,50,51
node,60,61
end,                      End node list
```

```
*
end,                      End table generation phase
* =================
* 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 schedule table
mb 0,                     Number of memory blocks for NS/1000 (0 if no NS)
us 1,                     Number of concurrent users (1 if no VC+)
lb,,                      LOGOF buffer limit (use defaults)
* ================================
* Labeled System Common Relocation
* ================================
*
* Add the system common table for HpMdm.run
*
re hpmdm_table.rel:92077
*
*
end,                      End labeled system common relocation
*
* Unlabeled (Blank) Common
*
com 1024,                 Number of words of memory to use
* ===============
* System Messages
* ===============
re %msgtb:92089,,         Message table
end
*
re %$m000:92089,,         Message module
end
*
end,                      End system messages
* ================
* System Libraries
* ================
lib sec1000.lib
lib $biglb.lib
end,                      End system libraries
*
end
*
*
*%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%*
*                                          /|\                                *
*                                           |                                 *
*       FIRST HALF, HP PRIMARY ONLY! -------+                                 *
*    SECOND HALF, SAMPLE ANSWER FILE -------+                                 *
*                                           |                                 *
*                                          \|/                                *
```

```
*%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%*
*
*
*     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, refer to the RTE–A System Generation     *
*     *   and Installation Manual, part no. 92077–90034.                    *
*     *                                                                     *
*     *   1.  Uncomment all required modules by removing the string         *
*     *       ”*!REMOVE!”.  Refer to the EDIT/1000 User’s Guide, part no.   *
*     *       92074–90001, and the note at the bottom of these instructions *
*     *       for information on how to use EDIT/1000 to edit this file.    *
*     *                                                                     *
*     *   2.  To select your system disk, remove *!REMOVESCSI! for SCSI     *
*     *       disk, remove *!REMOVEHPIB! and *!REMOVEHPIB64! for 64MB HP–IB *
*     *       disk, or remove *!REMOVEHPIB! and *!REMOVEHPIB20! for 20MB    *
*     *       HP–IB disk.                                                   *
*     *                                                                     *
*     *   3.  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 with the HP 92078A Virtual Code+ package,     *
*     *       a product which allows a multiuser environment and spooling.  *
*     *                                                                     *
*     *   4.  If you have NS/1000 and LAN/1000 in your system, uncomment    *
*     *       all NS/1000 and LAN/1000 modules by removing the string       *
*     *       ”*!YES_NS!”;  otherwise, uncomment the non–NS modules by      *
*     *       removing the string ”*!NO__NS!”.  If you have NS/1000 but not *
*     *       LAN/1000 in your system, refer to the NS subsystems’ answer   *
*     *       file.                                                         *
*     *                                                                     *
*     *   5.  If you have APRA/1000 in your system, rename the ”NS1000”     *
*     *       directory to ”APRA1000” and comment the %adv00 driver and LU  *
*     *       mapping (lu 81 – 82) out.                                     *
*     *                                                                     *
*     *   6.  If you do not use an ASIC card as LU 1 (as generated in       *
*     *       this system) comment it out and change one of the             *
*     *       terminal LUs generated into the system (such as the           *
*     *       8–channel multiplexer) as LU 1.                               *
*     *                                                                     *
*     *   7.  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_!”.  If you are using the A900 and *
*     *       NS/1000, choose %RPL91 then either rpl_a900_rev4.rel (rev.4   *
*     *       firmware or later) or xmb.rel (pre rev.4 firmware).           *
*     *                                                                     *
*     *   8.  If you don’t have NS/1000 in your system and you want to use  *
*     *       the DDC00/DDC01 modem handler, HPMDM.RUN, uncomment the       *
*     *       ”*!YES_NS!” before HPMDM_TABLE.REL.  If you want to save      *
*     *       space, and not use terminal cassette tapes and slaved         *
*     *       devices, use DDC00.REL wherever DDC01.REL is specified.       *
*     *                                                                     *
*     *   9.  Run the RTE-A generator (RTAGN) on the answer file,           *
*     *       producing a new system and snap file.  Place them on          *
*     *       the /SYSTEM directory (on the LU mounted by BOOTEX).          *
*     *                                                                     *
*     *  10.  Create a new welcome command file (or copy WELCOME1.CMD).     *
*     *       Modify it to copy your new snap file to /SYSTEM/SNAP.SNP.     *
*     *                                                                     *
*     *  11.  Create a new boot command file (or copy BOOT.CMD).  Modify    *
*     *       this boot command file to designate the system and snap       *
*     *       files just created.  Also modify the startup program          *
*     *       (st,,nn) to indicate your new welcome file (welcomenn.cmd),   *
*     *       where nn is between 1 and 99.                                 *
*     *                                                                     *
*     *  12.  Boot the new system as before, specifying the new boot        *
*     *       command file in the VCP> string.  In this way, you can save   *
*     *       the Primary System as a backup system.                        *
*     *                                                                     *
*     *   NOTE: EDIT/1000 can perform the string removal with the           *
*     *         command ”.,$x/*!REMOVE!//” (this edit command removes       *
*     *         all occurences of ”*!REMOVE!” from the current line         *
*     *         until the end of the file.  Be sure you are at the          *
*     *         first line you want to modify when you execute this         *
*     *         command.                                                    *
*     *                                                                     *
*     ***********************************************************************
*
*
*$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$*
*                                                                           *
*                             DISCLAIMER                                    *
*                             ==========                                    *
*                                                                           *
*    The Primary System is designed to be extremely flexible and serves     *
* two main purposes.  First, it is used as a verification tool for all      *
* supported peripherals on the A-Series.  Second, it is used to generate    *
* a customized system for the particular needs of the customer.             *
*                                                                           *
*    To provide maximum flexibility the primary system is NOT necessarily   *
* the configuration that HP recommends for the final system generation.     *
* Depending on the mix and cabling of the peripherals connected, the        *
* possibility exists of configuring the system in an unsupported manner,    *
* which can adversely impact system operation and performance.  It is the   *
* responsibility of the user to be aware of these limitations and not       *
* violate the maximum number or mix of devices on a given interface to      *
* avoid the possibility of data corruption or diminished system performance.*
*                                                                           *
*    For support and configuration information, contact your local sales    *
* representative or customer engineer for the information.                  *
*                                                                           *
*$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$*
*
*                   I/O    D E F I N I T I O N S
*
*  LU         DEVICE                         SELECT CODE        ADDRESS
* ----       --------                       -------------      ---------
```

```
*
*                              SCSI  #1
*                              ========
*   5        SCSI 7980S                         27B              4
*  14        DAT tape                           27B              3
* 12–13      650A                               27B              5
* 10–11      SCSI hard disk                     27B              6
*  50        SCSI floppy(single sided)          27B              0
*  51        SCSI floppy(double sided)          27B              0
*
*                              HP–IB #1
*                              ========
*         High–speed disk interface bus (load resistors installed),
*         maximum cable length 12 meters, 1 meter per device–load.
*                              ========
*         Maximum of 4 devices (up to four disks, four 9144/45 cartridge
*         tape units, or three of the above and one 7974/79/80).
*                              ========
*         If a 7974/79/80 is on the bus, the maximum cable length is
*         8 meters.
*                              ========
*         If this interface is connected to the 12122A internal disk
*         in the 248X computer, no other device can be connected to this
*         interface.
*
*  15     HP–IB #1 controller                   27B             36B
*
* 18–19   HP–IB disk                            27B              0
*   9     9144/5 standalone CS–80 cartridge tape 27B             1
* 16–17   HP–IB disk                            27B              2
*  30     HP–IB disk (20MB)                     27B              2
*  24     CS–80 compatible cartridge tape cache  27B             2
*  53     3.5” single–sided disk (9153/4)       27B              2
*  54     3.5” double–sided disk (9153/4)       27B              2
*   7     7974A/7978A streaming tape drive      27B              3
* 44–47   9133XV/4XV 5.25” Winchester hard disk  27B             4
* 36–37   9122D double–sided floppy             27B              5
* 32–33   9121 flexible disk                    27B              6
* 41–43   9133H/34H 5.25” Winchester fixed disk  27B             7
*  61     9133H 3.5” flexible disk              27B              7
*
*                              HP–IB #2
*                              ========
*         Low speed magnetic tape bus (load resistors installed),
*         a maximum of 2 7970 tape drives are supported.
*                              ========
*         Maximum cable length is 20 meters.
*
*  40     HP–IB #2 controller                   26B             36B
*   8     7970 HP–IB tape drive                 26B              4
*  28     7970 HP–IB tape drive                 26B              6
*
*                              HP–IB #3
*                              ========
*         High speed magnetic tape bus (load resistors installed),
*         a maximum of 2 7974/79/80 tape drives are supported.
*                              ========
*         Maximum cable length of 10 meters.
```

```
*
*  48     HI–IB #3 controller                    31B              36B
*  34     7974A/7980A streaming tape drive       31B               5
*  35     7974A/7980A streaming tape drive       31B               7
*
*                                HP–IB #4
*                                ========
*         Low speed peripheral / instrument bus.
*         Maximum cable length of 2 meters per device.
*                                ========
*         A maximum of two 256X printers are supported on one interface;
*         no other devices are supported with them
*
*  91     HP–IB #4 controller                    25B              36B
*  85     2608S,2563A,2566A line printer         25B               1
*   6     2932A line printer                     25B               2
*  92     device #1                              25B               3
*  93     device #2                              25B               4
*  94     device #3                              25B               5
*  95     device #4                              25B               6
*
*         248X INTEGRATED DISKS (Micro/1000 with 12022A controller)
*         =========================================================
*
* 55,62   15MB hard disk                         32B
* 59,60   20MB hard disk                         32B
*  39     single–sided 3.5” flexible disk        32B
*  63     double–sided 3.5” flexible disk        32B
*
*                NETWORK LINKS (NS/1000 and IEEE802.3 LAN)
*                ========================================
*
* 79,80   Telnet LU
* 81,82   NS/1000 LU mapping                     37B
* 96      12076A 802.3 LAN card                  37B
*
*                          PARALLEL INTERFACE CARD
*                          =======================
*  84    Parallel Interface card                 35B
*
*                                 TERMINALS
*                                 =========
*
*   Terminal configuration (NOTE: The 12005 is configured as system console
*     LU 1.  LUs may be swapped to designate another device as the new
*     system console.  You should have a system console device.)
*
*   1     12005 ASIC #1                          20B
*  100    12005 ASIC #2                          21B
* 110–113 12100 A400 OBIO 4–channel MUX          77B
*  211    slaved device(printer) port B
*  212    left CTU               port B
*  213    right CTU              port B
* 120–127 12040D 8–channel MUX                   30B
*  221    slaved device(printer) port 1
*  222    left CTU               port 1
*  223    right CTU              port 1
* 130–137 12040A/B/C 8–channel MUX               23B
```

```
*
*  Free lus:
*    2–4, 20–23, 25–27, 29, 31, 38, 49, 56–58,
*    64–69, 70–78, 83, 86–89, 90, 97–99, 101–109, 114–119, 140–210,
*    214–219, 220, 224–255.
*
*!REMOVE!links,cp,,                Use current page links
*
* System Relocation
*
*!REMOVE!er,,                      Echo errors to the terminal
*!REMOVE!le,off,                   Do not list the module entry points
*!REMOVE!re /rte_a/%vctr,,         Entry points
*!REMOVE!tg 950                    Number of tags required
*!REMOVE!re /rte_a/%mapos,,        Partitioned OS tag routines
*
*     ###################################################################
*     #                                                                 #
*     #   If you use an RPL file for a hardware configuration with      #
*     #   fewer features than yours, your system will not be performing #
*     #   as well as it could.  You may not use an RPL file for a       #
*     #   hardware configuration with more features than yours.         #
*     #                                                                 #
*     #   Choose the correct RPL file for your system from the          #
*     #   choices below and remove the ”*” in front of it.              #
*     #                                                                 #
*     ###################################################################
*
*                                                     DOUBLE PRECISION
*    RPL FILE            PROCESSOR TYPE     CDS        FLOATING POINT
*   ----------          ----------------   -----     ------------------
*re /rte_a/%rpl40,,            A400          NO              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
*re /rte_a/rpl_a990.rel,,      A990          NO              YES
*re /vcplus/rpl_a990_cds.rel,, A990          YES             YES
*
*     ###########################################################
*     #                                                         #
*     #  For A900 with either NS/1000 or %envrn, choose either  #
*     #  rpl_a900_rev4.rel (rev.4 firmware or later) or         #
*     #  xmb.rel (pre rev.4 firmware)                           #
*     #                                                         #
*     ###########################################################
*
*re /rte_a/rpl_a900_rev4.rel
*re /rte_a/xmb.rel,mb02         For NS/1000 with A900 pre rev.4 firmware only
*re /rte_a/xmb.rel,mb12         For either NS/1000 or %envrn with
```

```
*                                  A900 pre rev.4 firmware only
*re /rte_a/xmb.rel,mb01         For %envrn with A900 pre rev.4 firmware only
*re /rte_a/xmb.rel,mb10         For %envrn with A900 pre rev.4 firmware only
*re /rte_a/xmb.rel,mb21         For %envrn with A900 pre rev.4 firmware only
*
*!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/%$IDRPL,,       System ID dup IDRPL
*!REMOVE!re /rte_a/%signl,,        Signal processing module
*!YESVC+!re /vcplus/secos.rel,,    Security/1000 module
*!YESVC+!re /vcplus/check.rel,,    Security/1000 module
*!YESVC+!re /vcplus/%spool,,       Spooling module
*
*
*     #################################################################
*     #                                                               #
*     #   If you are using an A600 (not an A600+) WITHOUT CDS, then   #
*     #   the following paragraph applies.  If you are using any      #
*     #   other type of A-Series processor, (A400 with or without     #
*     #   CDS, A600+, A700 or A900) then the following paragraph      #
*     #   does NOT apply.                                             #
*     #                                                               #
*     #   The two routines .DMP and .DDI must be relocated here       #
*     #   because they are required by code that is in the O.S.       #
*     #   partitions.  You cannot do library searches for code in     #
*     #   O.S. partitions.  This is required ONLY when using an       #
*     #   A600 (using %RPL60).  If you are using any other RPL,       #
*     #   these modules will cause duplicate entry point errors       #
*     #   if included in the generation.                              #
*     #                                                               #
*     #################################################################
*
*!RPL60_!re /rte_a/$math,.dmp   Required for A600 (not A600+)   (RPL60)
*!RPL60_!re /rte_a/$math,.ddi   Required for A600 (not A600+)   (RPL60)
*
*!REMOVE!se /rte_a/$syslb,,         Search the system library
*
* define partitionable modules
*
*!REMOVE!pa perr,xcmnd,stat,dsq,vema,lock,load,memry,iorq
*!REMOVE!pa time,class,abort,alarm
*!YESVC+!pa cdsfh,envrn
*!YES_NS!pa nsabp
*
*    #################################################################
*    #                                                               #
*    #  If you want to include the dummy version of a partitionable  #
*    #  module from $SYSA.LIB, remove the module from the PA module  #
*    #  list above as well as from the OS partition relocation phase #
*    #  below.  For example, if you do not want NS/1000, you can     #
*    #  remove DSQ from the first PA command above and leave the     #
*    #  relocation statement of DSQ commented out below.             #
*    #                                                               #
*    #################################################################
*
*!REMOVE!ms /rte_a/$sysa,,     Search the system dummy library
*
*!REMOVE!end,                      End system relocation phase
*
* OS module/Driver partition phase
*
*!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!re /rte_a/%time,,         Time scheduling of programs
*!REMOVE!end
*
*!REMOVE!re /rte_a/%class,,        Class I/O module
*!REMOVE!end
*
*!REMOVE!re /rte_a/%xcmnd,,        Operator command extension module
*!REMOVE!re /rte_a/%stat,,         Status command module
*!REMOVE!re /rte_a/%dsq,,          NS/1000 module
*!REMOVE!end
*
*!YESVC+!re /vcplus/%cdsfh,,       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
*
*!YESVC+!re /vcplus/%envrn,,       EXEC interface to Environment Var Block
*!REMOVE!re /rte_a/%alarm,,        Timer signal processing module
*!REMOVE!end
*
*!REMOVE!re /rte_a/%abort,,        Abort processing module
*!YES_NS!re /ns1000/rel/nsabp.rel,,NS–ARPA abort processor
*!REMOVE!end
*
* Driver partitions
*
*!REMOVE!re /rte_a/ddq30.rel,,      Device driver for SCSI disks
*!REMOVE!end
*
*!REMOVE!re /rte_a/ddq24.rel,,      Device driver for SCSI DAT tape
*!REMOVE!end
*
*!REMOVE!re /rte_a/idq35.rel,,      SCSI interface drvier
```

```
*!REMOVE!end
*
*!REMOVE!re /rte_a/%dd*33,,        Device driver for CS/80 disks
*!REMOVE!end
*
*!YES_NS!re /rte_a/idz00.rel,,            Telnet driver
*!YES_NS!end
*
*!YES_NS!re /rte_a/%id*67,,        LAN: interface driver for IEEE802.
*!YES_NS!end
*
*!REMOVE!re /rte_a/%id*37,,        Interface driver for CS/80 disks
*!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
*!REMOVE!end
*
*!REMOVE!re /rte_a/%dd*00,,        Device driver for %idm00
*!REMOVE!re /rte_a/%dd*30,,        Device driver for disks (ICD)
*!REMOVE!end
*
*!YES_NS!re /ns1000/dsrel/%adv00,,       device driver for NS/1000
*!YES_NS!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
*
*    ###################################################################
*    #                                                                 #
*    #  If you want to use slaved devices (such as printers) or CTUs   #
*    #  connected to your terminal, use DDC01.REL instead of DDC00.REL.#
*    #  Refer to the RTE–A System Generation and Installation manual,  #
*    #  part no. 92077–90034.                                          #
*    #                                                                 #
*    ###################################################################
*
*!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 disk
*!REMOVE!re /rte_a/%dd*24,,        Device driver for 7974A/78A streaming tape
*!REMOVE!end
*
*!REMOVE!re /rte_a/id100.rel,,     Interface driver for 12005 ASIC card
*!REMOVE!re /rte_a/%id*50,,        Interface driver for parallel card
*!REMOVE!end
*
*!REMOVE!end,,                     End OS module/Driver partition phase
*
*
```

```
* Table Generation phase – configure LU tables
*
*
* HP–IB #1 –– High speed disk interface bus         Select Code = 27b
* SCSI  #1 –– SCSI disk/DAT tape interface bus
*
*!REMOVEHPIB!ift,/rte_a/%Id*37,SC:27B
*
*!REMOVESCSI!ift,/rte_a/idq35.rel,SC:27B
*
* SCSI DAT tape                                      LU 14
* SCSI 7980S                                         LU 5
*
*!REMOVESCSI!dvt,/rte_a/ddq24_gen.rel,,lu:14,dp:1:3
*!REMOVESCSI!dvt,/rte_a/ddq24_gen.rel,M7980,lu:5,dp:1:4
*
* SCSI hard disk
*
*!REMOVESCSI!dvt,/rte_a/ddq30_gen.rel,m64mb:0,lu:10,dp:1:6 dp:8:1
*!REMOVESCSI!dvt,/rte_a/ddq30_gen.rel,m64mb:1,lu:11,dp:1:6 dp:8:1
*
* SCSI 650A MO disk
*
*!REMOVESCSI!dvt,/rte_a/ddq30_gen.rel,m64mb:0,lu:12,dp:1:5 dp:8:3
*!REMOVESCSI!dvt,/rte_a/ddq30_gen.rel,m64mb:1,lu:13,dp:1:5 dp:8:3
*
* SCSI floppy  single sided                   LU 50
*
*!REMOVESCSI!dvt,/rte_a/ddq30_gen.rel,m16mb:0,lu:50,dp:1:0:0:0:0:0,–
*!REMOVESCSI!  dp:6:66:16:0
*
* SCSI floppy  double sided                   LU 51
*
*!REMOVESCSI!dvt,/rte_a/ddq30_gen.rel,m16mb:0,lu:51,dp:1:0:0:0:0:0,–
*!REMOVESCSI!  dp:6:154:16:0
*
*
*    ###################################################################
*    #                                                                 #
*    #  The bus controller dvt is needed only if Direct I/O to the     #
*    #  device is being done.                                          #
*    #                                                                 #
*    ###################################################################
*
* Bus Controller LU                                 HP–IB address 36b
*                                                     LU 15
*
*!REMOVEHPIB!dvt,,,LU:15,TO:2000,DT:37B,TX:0,DX:1,DP:1:36B,PR:0
*
*  HP–IB disk                                       HP–IB address 0
*                                                     LU 18–19
*
*!REMOVEHPIB!dvt,/rte_a/%dd*33,M7908_CF:0,LU:18,DP:1:0,–
*!REMOVEHPIB!  DP:2:0:0:00:0:4096 DP:7:64
*!REMOVEHPIB!dvt,/rte_a/%dd*33,M7908_CF:0,LU:19,DP:1:0,–
*!REMOVEHPIB!  DP:2:0:0:04:0:4096 DP:7:64
*
*
```

```
* 9144/5 standalone CS–80 cartridge tape            HP–IB address 1
*                                                     LU 9
*
*!REMOVEHPIB!dvt,/rte_a/%dd*33,M9144:0,LU:9,DP:1:1
*
*  HP–IB disk (20MB)                                HP–IB address 2
*                                                     LU 30
*
*!REMOVEHPIB20!dvt,/rte_a/%dd*33 m7908_cf:0 lu:30 dp:1:2 dp:2:0:0:00:0:1215,–
*!REMOVEHPIB20!  dp:7:64
*
*      Single–sided floppy                          HP–IB address 2
*                                                     LU 53
*
*!REMOVEHPIB20!dvt,/rte_a/%dd*33,M_FLOPPY_CF:1,LU:53,DP:1:2:400b:0:0:0,–
*!REMOVEHPIB20!  DP:6:66:16:0
*
*      Double–sided floppy                          HP–IB address 2
*                                                     LU 54
*
*!REMOVEHPIB20!dvt,/rte_a/%dd*33,M_FLOPPY_CF:1,LU:54,DP:1:2
*
* CS–80 compatible cartridge tape with cache        HP–IB address 2
*   size of disk cache is 256 physical blocks         LU 24
*   starting block address of disk cache is 77760
*
*!REMOVEHPIB20!dvt,/rte_a/%dd*33,mtape,lu:24,dp:1:2 dp:4:1:12224
*
*
*
*      HP–IB disk(64MB)                             HP–IB address 2
*                                                     LU 16–17
*
*!REMOVEHPIB64!dvt,/rte_a/%dd*33,M7908_CF:0,LU:16,DP:1:2,–
*!REMOVEHPIB64!  DP:2:0:0:00:0:4096 DP:7:64
*!REMOVEHPIB64!dvt,/rte_a/%dd*33,M7908_CF:0,LU:17,DP:1:2,–
*!REMOVEHPIB64!  DP:2:0:0:04:0:4096 DP:7:64
*
* CS–80 compatible cartridge tape with cache        HP–IB address 2
*   size of disk cache is 256 physical blocks         LU 24
*   starting block address of disk cache is 262144
*
*!REMOVEHPIB64!dvt,/rte_a/%dd*33,mtape,lu:24,dp:1:2 dp:4:4:0
*
*
* 7974A/7978A streaming tape drive                  HP–IB address 3
*                                                     LU 7
*
*!REMOVEHPIB!dvt,/rte_a/%dd*24,M7974:0,LU:7,DP:1:3
*
* 5.25” Winchester fixed disk (9133/4 XV)           HP–IB address 4
*                                                     LU 44–47
*
*!REMOVEHPIB!dvt,/rte_a/%dd*30,M9134X:0,LU:44,DP:1:4
*!REMOVEHPIB!dvt,/rte_a/%dd*30,M9134X:1,LU:45,DP:1:4
*!REMOVEHPIB!dvt,/rte_a/%dd*30,M9134X:2,LU:46,DP:1:4
*!REMOVEHPIB!dvt,/rte_a/%dd*30,M9134X:3,LU:47,DP:1:4
*
```

```
* flexible disk (9122D)                             HP–IB address 5
*                                                     LU 36–37
*
*!REMOVEHPIB!dvt,/rte_a/%dd*33,m_floppy_cf:0,lu:36,DP:1:5
*!REMOVEHPIB!dvt,/rte_a/%dd*33,m_floppy_cf:1,lu:37,DP:1:5
*
* 5.25” and 3.5” flexible disks (9121)              HP–IB address 6
*                                                     LU 32–33
*
*!REMOVEHPIB!dvt,/rte_a/%dd*30,M9121:0,LU:32,DP:1:6,TO:3000
*!REMOVEHPIB!dvt,/rte_a/%dd*30,M9121:1,LU:33,DP:1:6,TO:3000
*
* 3.5” flexible disk (9133H)                        HP–IB address 7
*                                                     LU 61
*
*!REMOVEHPIB!dvt,/rte_a/%dd*33,M_FLOPPY_CF:1,LU:61,DP:1:7
*
* 5.25” Winchester fixed disk (9133H/9134H)         HP–IB address 7
*                                                     LU 41–43
*
*!REMOVEHPIB!dvt,/rte_a/%dd*33,M9133_CF:0,LU:41,DP:1:7
*!REMOVEHPIB!dvt,/rte_a/%dd*33,M9133_CF:1,LU:42,DP:1:7
*!REMOVEHPIB!dvt,/rte_a/%dd*33,M9133_CF:2,LU:43,DP:1:7
*
*
* HP–IB #2 –– Low speed mag tape bus                Select Code = 26b
*
*
*!REMOVE!ift,/rte_a/%Id*37,SC:26B
*
* Bus Controller LU                                 HP–IB address 36b
*                                                     LU 40
*
*!REMOVE!dvt,,,LU:40,TO:2000,DT:37B,TX:0,DX:1,DP:1:36B,PR:0
*
* 7970 tape drive                                   HP–IB address 4
*                                                     LU 8
*
*!REMOVE!dvt,/rte_a/%dd*23,M7970E:0,LU:8,DP:1:4,PR:1
*
* 7970 tape drive                                   HP–IB address 6
*                                                     LU 28
*
*!REMOVE!dvt,/rte_a/%dd*23,M7970E:0,LU:28,DP:1:6,PR:1
*
* HP–IB #3 –– High speed mag tape bus               Select Code = 31b
*
*
*!REMOVE!ift,/rte_a/%Id*37,SC:31B
*
* Bus Controller LU                                 HP–IB address 36B
*                                                     LU 48
*
*!REMOVE!dvt,,,LU:48,TO:2000,DT:37B,TX:0,DX:1,DP:1:36B,PR:0
*
*
* 7974A/7978A streaming tape drive                  HP–IB address 5
*                                                     LU 34
```

```
*
*!REMOVE!dvt,/rte_a/%dd*24,M7974:0,LU:34,DP:1:5
*
* 7974A/7978A streaming tape drive                  HP–IB address 7
*                                                     LU 35
*
*!REMOVE!dvt,/rte_a/%dd*24,M7974:0,LU:35,DP:1:7
*
*
* HP–IB #4 –– Low speed peripheral/instrument bus   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:37B,TX:0,DX:1,DP:1:36B
*
* 2608S,2563A,2566A Line Printer                    HP–IB address 1
*                                                     LU 85
*
*!REMOVE!dvt,/rte_a/%ddC12,,LU:85,DP:1:1
*
* 2932A Line Printer                                HP–IB address 2
*                                                     LU 6
*
*!REMOVE!dvt,/rte_a/%dd*12,M2932A,LU:6,DT:12B,DP:1:2
*
* Four devices                                      HP–IB address 3–6
*                                                     LU 92–95
*
*!REMOVE!dvt,,,LU:92,TO:500,DT:77B,DX:1,DP:1:3
*!REMOVE!dvt,,,LU:93,TO:500,DT:77B,DX:1,DP:1:4
*!REMOVE!dvt,,,LU:94,TO:500,DT:77B,DX:1,DP:1:5
*!REMOVE!dvt,,,LU:95,TO:500,DT:77B,DX:1,DP:1:6
*
*
* 248x INTEGRATED DISK INTERFACE (MICRO/1000)       Select Code = 32b
*
*
*!REMOVE!ift,/rte_a/%Id*27,SC:32B
*
* Hard disk (15Mb)
*
*!REMOVE!dvt,/rte_a/%GEN27,M2480:15,LU:55
*!REMOVE!dvt,/rte_a/%GEN27,M2480:16,LU:62
*
* Hard disk (20Mb)
*
*!REMOVE!dvt,/rte_a/%gen27,m2480:11,lu:59
*!REMOVE!dvt,/rte_a/%gen27,m2480:12,lu:60
*
* 3.5” flexible disk  (singled–sided)
*
*!REMOVE!dvt,/rte_a/%GEN27,M2480:3,LU:39
*
* 3.5” flexible disk (double–sided)
```

```
*
*!REMOVE!dvt,/rte_a/%GEN27,M2480:14,LU:63
*
*
*
* IEEE 802.3 Local Area Network                     Select Code = 37b
*                                                     LU 96
*
*    ##################################################################
*    #                                                                #
*    #  The following are the table entries for ID*67, the LAN/1000  #
*    #  driver for the 12076A card.  The IFT extension area           #
*    #  defaults to 72 words.                                         #
*    #                                                                #
*    ##################################################################
*
*!YES_NS!ift,/rte_a/%id*67,sc:37B
*!YES_NS!dvt,,,lu:96,DT:67b,TO:200
*
* LU mapping                                          LU 81–82
*
*!YES_NS!ift,/ns1000/dsrel/%ADV00,EIDV00,QU:FI,TX:2,AL:DY
*
*!YES_NS!dvt,,,LU:81,EddV00,TX:0
*!YES_NS!dvt,,,LU:82,EddV00,TX:5
*
*  Telnet pseudo terminal driver/LUs                  LU 79–80
*
*!YES_NS!ift,/rte_a/idz00.rel
*!YES_NS!dvt,/rte_a/ddc01.rel,MHP_TELNET,lu:79
*!YES_NS!dvt,/rte_a/ddc01.rel,MHP_TELNET,lu:80
*
*
* Parallel Interface Card                           Select Code = 35b
*                                                     LU 84
*
*!REMOVE!ift,/rte_a/%Id*50,SC:35B
*
*!REMOVE!dvt,,,LU:84,TO:5000,TX:2,DX:3,DP:1:0:0:0,DT:55B
*
*
*  Terminal LUs
*
*      ##############################################################
*      #                                                            #
*      # For systems with non–12005 ASIC system consoles:          #
*      #   change the ’lu:1’ below to ’lu:yy’, where yy is an      #
*      #   unused LU, and change the desired LU:XX to LU:1.        #
*      #                                                            #
*      ##############################################################
*
* ASIC #1 (default system console)                  Select Code = 20B
*                                                     LU 1
*
*!REMOVE!ift,/rte_a/id100.rel,sc:20B
*!REMOVE!dvt,/rte_a/ddc01.rel,MHP_Term:A,lu:1
*
* ASIC #2  Terminal, slaved device (printer),       Select Code = 21B
*          left CTU and right CTU.                    LU 100, 200–202
*
*!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 A400 OBIO 4–MUX portA–D, portB’s          Select Code = 77B
*     slaved device (printer), left CTU, right CTU.   LU 110–113,211–213
*
*    #################################################################
*    #                                                               #
*    #  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:0,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 8–channel MUX, ports 0–7, port 1 is      Select Code = 30B
*     slaved device (printer), left CTU, right CTU.   LU 120–127,221–223
*
*!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 Pre–Revision D 8–MUX, port 0–7.        Select Code = 23B
*                                                     LU 130–137
*
*!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!end,,                     End of dvt generation phase
*
*!REMOVE!end,,                     End of ift generation phase
*
*
* Define Node Lists
*
* SCSI disks
*
*!REMOVESCSI!node,10,11
*!REMOVESCSI!node,12,13
*
* HP–IB disk

*!REMOVEHPIB!node,18,19
*
* 5.25” and 3.5” Flexible disks (9121)
*
*!REMOVEHPIB!node,32,33
*
* 9122D 3.5” flexible disks
*
*!REMOVEHPIB!node,36,37
*
* HP–IB disk(64MB)
*
*!REMOVEHPIB64!node,16,17,24
*
* HP–IB disk(20MB)
*
*!REMOVEHPIB20!node,30,24,53,54
*
* 5.25” Winchester fixed disk LUs (9133/4XV)
*
*!REMOVEHPIB!node,44,45,46,47
*
* 5.25” Winchester fixed disk LUs (9133/4H) and 9133H flexible disk
*
*!REMOVEHPIB!node,41,42,43,61
*
* 248x Integrated disk LUs
*
*!REMOVE!node,55,62,59,60,39,63
*
* 4–channel MUX port B/Rev. D 8–channel MUX CTUs 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 80,                    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 schedule table
*
*!YES_NS!mb 512,                   Number of memory blocks for NS/1000
*!NO__NS!mb 0,                     Number of memory blocks for NS/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_NS!re /ns1000/dsrel/%resa
*
*    #################################################################
*    #                                                               #
*    #  If you are using the serial modem handler HPMDM, uncomment   #
*    #  the following line.                                          #
*    #                                                               #
*    #################################################################
*!YES_NS!re /rte_a/hpmdm_table.rel
*
*!YES_NS!se /ns1000/lib/nssys.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,                      End system messages
*
* System Libraries
*
*    #################################################################
*    #                                                               #
*    #   add here any other library routines which you often use     #
*    #                                                               #
*    #################################################################
*
*!REMOVE!lib sec1000.lib,,         Security/1000 library
*!YES_NS!lib bigns.lib,,           Merged NS–ARPA/1000 user libraries
*!REMOVE!lib $FNDLB,,              FORTRAN non–DS library
*!YES_NS!lib pascal.lib,,          Required for NS–ARPA/1000
*!REMOVE!lib $biglb.lib,,
*!REMOVE!end,                      End system libraries
*
* CDS Libraries
*
*    #################################################################
*    #                                                               #
*    #  add here any other CDS library routines which you often use  #
*    #                                                               #
*    #################################################################
*
*!YESVC+!lib sec1000cds.lib,,      Security/1000 CDS library
*!YES_NS!lib bigns_cds.lib,,       Merged NS–ARPA/1000 user libraries
*!YES_NS!lib pascal_cds.lib,,      Required for NS–ARPA/1000
*!YESVC+!lib $bgcds.lib,,
*!YESVC+!lib $FNDLB,,              FORTRAN non–DS library
*!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 `%B`*device*`[`*ffbusc*`][`*file*`][,SS]`
   or
   `%L`*device*`[`*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`	Disk drive or cartridge tape drive (CTD) via HP-IB, or SCSI disk drive
`DS`	HDLC link or LAN link
`MT`	Magnetic tape drive, or DAT tape drive
`DI`	Integrated disk

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 disks, 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 disk (cylinder 0) when the boot extension file is loaded. If any other value is given, the disk head moves 256 blocks times this value before reading from the disk.

  If you are booting over DS (HDLC) or LAN, the *ff* parameter is an octal number from 00000 to 77777. It is converted to ASCII to form the P*fffff* file name.

- *b* = Bus address (or disk address). The disk ROM loader uses this value to specify the HP-IB or SCSI address (0-7) of the disk unit from which to load the system file. For the 248x integrated disk, the disk ROM loader uses this value to specify the integrated disk address from which to load the system file. The unit number must be 0 for both the fixed disks and the micro-floppy disk driver.

- *u* = Unit or head number from which to load the file (0 or 1). For the 7902 disk, this parameter is 0 for the left drive or 1 for the right. In the case of the 7906 disk, this parameter is the head number. In the case of the CS/80 disk drive with CTD, this parameter is 0 for the disk drive or 1 for the cartridge tape. In the case of the SCSI disk drive, this parameter is 0.

- *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 (disk-based system). The file for a disk volume must be in directory SYSTEM. The default name for both of these files is BOOT.CMD::SYSTEM for a file disk 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
Disk via HP-IB/SCSI	2027/6027	/SYSTEM/BOOT.CMD or SYSTEM
PROM Module	22	none
DS (HDLC) Link	24	none
LAN Link	24	none
Magnetic/DAT Tape	4027	none
248x Integrated Disk Controller	32	/SYSTEM/BOOT.CMD 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.

The rest of this appendix explains how to boot your system from various media. For booting a memory-based system over a LAN, refer also to Chapter 11.

Loading Your System from Disk

Before loading your new disk-based system into memory from disk, make sure you do the following:

- Ensure that the boot extension (BOOTEX) is physically stored beginning at cylinder zero of the disk drive from which the system is to be loaded. Refer to the INSTL program in the *RTE-A User's Manual*, part number 92077-90034, 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 or SCSI address, unit number, head number, and the interface card select code of the disk (or the select code of the 248x integrated disk 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 disk on the HP-IB/SCSI bus, enter:

   `%bdc`*buscfile*`[,ss]`

   Default is BOOT.CMD::SYSTEM (or system)

   Default is 2027 (HP-IB) or 6027 (SCSI)

   where:

   `%bdc` Execute the disk bootstrap loader program for the disk 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/SCSI address of the disk drive where your target system file resides. Default address is 2 for HP-IB, 6 for SCSI.

   u Unit number or head number of the disk drive (0-7) where the system file resides. Default value is 0.

   sc Select code of the HP-IB/SCSI card to which the disk 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 disk 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 disk 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 disk, 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 disk a system with a boot command file called BOOT.CMD::SYSTEM and with a hierarchical file system:

   Bus Address = 2 (HP-IB), or 6 (SCSI)
   Unit Number = 0
   Select Code = 27   (hierarchical file disk 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
   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 disk.

2. To load the diagnostic program called HPIB::SYSTEM from the disk 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

   12009A HP-IB INTERFACE DIAGNOSTIC

   READY
   ```

   In the above example, all parameters must be entered and the diagnostic program specified, because none of them are the default values.

3. To cold load a system with a boot command file called MUTIGR::SYSTEM from the disk with:

   Bus Address = 2 (HP-IB) or 6 (SCSI)
   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 because 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 disk with a boot command file called BOOT.CMD::SYSTEM, using a hierarchical file disk 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 because the default file name is used. This command also could be given as

   ```
   VCP> bdc7
   ```

   because 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 disk 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 disk 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 disk with:

Bus Address	= 7	
Unit Number	= 0	
Select Code	= 22	(hierarchical file disk 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 because the default file name is used.

7. To load and execute a system from disk, entering the boot commands interactively, with BOOTEX and the system and snap files described by

   Bus Address = 2 (HP-IB), or 6 (SCSI)
   Unit Number = 0
   Select Code = 27

   The following sequence of commands should be entered at the VCP (user input is underlined):

```
CI>   (Strike the BREAK key)
P 002675 A 077774 B 000000 RW 100003  M 002674 T 026655

VCP> %bdclu1

BOOTEX:  sy,systmc                 system file on directory SYSTEM
BOOTEX:  sn,snapc                  snap file name
BOOTEX:  mc,13                     mount disk 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 disk 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:

  `CI> 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: `%BDC`*ffb*`1`*sc*

   where:

   `%BDC` Executes the bootstrap loader program for disk 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 disks, 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 *
         :
         :
```

Because 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 `%BRM00`*sc*

   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 (HDLC) 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 disk-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 disk at the source. Give the system file one of the default DS file descriptors P*nnnnn*:0:0, where *nnnnn* is the file number you will use in the bootstring at the destination system. If this file descriptor can be P00000:0:, you will be able to default the file number in the bootstring, 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 bootstring 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 bootstring.
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 `%BDS`*ff*`00`*sc*

   where:

   `%BDS` Executes the DS/LAN boot loader program and begins execution of the system when the entire merged system file has been loaded into memory.

   ff File number. The *ff* parameter is an octal number from 00000 to 77777. It is converted to ASCII to form the P*fffff* file name. Refer to "DS Link" and "LAN Link" in Chapter 10.

`00` Use 00 as placeholders for the *bu* parameters in the bootstring if the *ff* parameter is not zero.

sc is the select code of the target system DS (HDLC) 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 over a LAN Link

Before loading your new system over a LAN link, perform the steps given below. Refer to Chapter 11 for details.

1. Create the merged system file using BUILD.
2. Select a server computer system. The server must be disk-based and must be running LAN VCP software (VCPMT). The server computer and the target computer must be connected by a LAN link.
3. Put the merged system file on disk at the server. Give the system file one of the default LAN file descriptors /FILES802/P*nnnnn*, where *nnnnn* is the file number you will use in the bootstring at the client system.

   The default file number 00000 will be translated according to the ipl_table.txt file instead of using P000000. Refer to Chapter 11.

With the above steps completed, you can boot the target system at the client computer VCP terminal. The boot procedure is the same if the VCP terminal is itself remotely located at a LAN node running RMVCP and VCPMT. When you are at the keyboard of the VCP terminal, you are effectively at the client 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 `%BDS`*ff*`00`*sc*

   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 File number. The *ff* parameter is an octal number from 00000 to 77777. It is converted to ASCII to form the P*fffff* file name. Refer to “DS Link” and “LAN Link” in Chapter 10.

   `00` Use 00 as placeholders for the *bu* parameters in the bootstring if the *ff* parameter is not zero.

   sc is the select code of the target system LAN interface card that is directly connected to the server computer system.

Loading Your System from Magnetic or DAT Tape

Before loading your system from magnetic or DAT 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: `%BMT`*ffbusc*

   where:

   `%BMT` Executes the bootstrap loader program for magnetic tape.

   ff 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.

   b HP-IB/SCSI bus address of the unit from which to load system file. The magnetic tape is preset to address 4 (the default value) in the Primary System. You can change this address to a value between 0 and 7.

   u Unit number of the magnetic tape, normally 0 (the default value).

   sc Select code of HP-IB/SCSI 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 *
        :
        :
```

Because 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/SCSI 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 243x/8x Integrated Disks

Before loading your disk-based system into memory from disk, do the following:

- Ensure that the boot extension (BOOTEX) is physically stored beginning at cylinder zero of the disk drive from which the system is to be loaded. Refer to the INSTL program in the *RTE-A User's Manual*, part number 92077-90002, 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 disk 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 `%bdi`*buscfile*`[,SS]`

   where:

   `%bdi` Execute the disk bootstrap loader program for the 248x Integrated disk. 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 Disk address of the integrated disk where the target system file resides. The default address is 0.

b = 0	First fixed disk
b = 3	Microfloppy

u Unit number of the microfloppy drive. The default is 0.

u = 0	First microfloppy

sc Select code of the integrated disk 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.CMD (or SYSTEM for a FMGR disk 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 disk 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:

Disk Address	= 0	
Unit Number	= 0	
Select Code	= 27	(disk 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 disk.

2. To load the diagnostic program called /SYSTEM/HPIB from the disk with:

   Disk 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, because neither is the default value.

3. To cold load a system with a boot command file called /SYSTEM/MUTIGR from the disk with:

   Disk 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 disk with a boot command file called /SYSTEM/BOOT.CMD:

   Disk 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:

   Disk Address = 0
   Unit Number = 0
   Select Code = 22   (disk 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 disk, entering the boot commands interactively. BOOTEX and the system and snap files are on the disk described by:

   Disk Address = 0
   Unit Number = 0
   Select Code = 32

   Enter the following sequence of commands from the VCP (user input is underlined):

```
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 disk 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 disk 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 BUILD program 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).

Note Refer to Chapter 10 for additional information on the BUILD program.

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[,+D[,command[,list[,output[,snapshot[,system[,op]]]]]]]
```

where:

`+D` An optional parameter indicating you want to include an initialized RAM disk.

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` or `A` Abort and purge the BUILD system file created.
`END` or `E` Exit, saving the system file created.
`CONTINUE` or `C` Continue processing subsequent commands on all but unexpected command file EOF errors, or errors associated with runstring files.

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[IORITY],*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 disk-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 disk-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 underlined 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.

```
CI> 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.

```
CI> 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 Files

The following is an example 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 (for example, 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.

```
CI>  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,FI,LE,,,       make APPLN the startup program; pass parameters
/E,                  complete the BUILD process
```

The following is an example BUILD command file that includes RAM disk initialization.

```
client_mb.sys:::1:16384
client.snp
client.sys
yes,,            Automatic partitioning
2048,,           System size in pages
```

```
rp,/programs/drtr.run,d.rtr,,   Built in programs
rp,/programs/derr.run,d.err
rp,/programs/ci.run,cm
rp,/programs/ci.run,ci.01
rp,/programs/ci.run,ci.s
st,,1
rp,/programs/inpro.run
rp,/programs/outpro.run
rp,/programs/uplin.run
rp,/programs/inetd.run
rp,/programs/evmon.run
rp,/programs/qclm.run
rp,/programs/grpm.run
rp,/programs/ifpm.run
rp,/programs/matic.run
rp,/programs/#send.run
rp,/programs/execm.run
rp,/programs/execw.run
rp,/programs/ptopm.run
rp,/programs/dsrtr.run
rp,/programs/trfas.run
rp,/programs/iomap.run
rp,/programs/lumap.run
rp,/programs/luque.run
/e
2,mc,,           RAM disk LU 2, mounted at boot
/system/welcome1.cmd
/system/client.nsin
/system/nsinit.msg
/system/nserrs.msg
/e,,             Above files go in /system
system
/programs/ci.run
/programs/cix.run
/programs/io.run
/programs/dl.run
/programs/edit.run
/programs/li.run
/programs/wh.run
/programs/nsinit.run
/programs/mminit.run
/programs/nspars.run
/programs/nspr1.run
/programs/nspr2.run
/programs/nspr3.run
/programs/telnet.run
/programs/tnsrv.run
/e,,             Above files go in /programs
programs
/e
/e
```

J

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 disk 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 disk 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 (for example, %BDCffbuscFILENAME). The boot command file must be on the disk LU from which you are booting. If the boot command file is not found on that disk, 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:
%BDC*ffbusc*--

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 causes it to print an error message to the terminal. If an error is not fatal, BOOTEX continues 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 disks; 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
```

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 number 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 overwrites the SY file before booting. This prevents 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,[`*p1*`][,`*p2*`]`

where:

p1 = number of words of SAM (if null, size of SAM is unchanged).

p2 = number of words of XSAM (if null, size of XSAM is unchanged).
If *p2* is a dash (−), then XSAM is made identical to SAM; that is, the two are no longer separate and both are allocated from SAM.

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 disk LU the operating system uses for VMA backing store scratch files.

You may specify that the VMA backing store scratch files are to be put into the /SCRATCH directory by using the command:

```
SC,0
```

You may specify that the VMA backing store scratch files are to be put onto a FMGR cartridge using:

`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, /SCRATCH is used. The LU must be a FMGR LU; otherwise, an error VM209 (no such directory) occurs 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 skips 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 (that is, *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 is 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 file name, *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 disk 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 nor :d is assigned, the default is :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 are swapped.

```
SW,filedescriptor
```

If BOOTEX finds the file given in the command, that file is purged and a swap file with the same name is created by using the 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 words. (Refer to the *RTE-A Programmer's Reference Manual*, part number 92077-90007, 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 creates 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 disk (or the first disk mounted if CRN/LU not specified.) If not found, it is created (SWAP::0 on a FMGR disk, /SYSTEM/SWAP.SWP on a disk volume). If this command is not given, a default swap file, SWAP, is created on the system disk. If the disk is a FMGR disk, 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 disk LUs is maintained in the first two blocks of the swap file. If an error occurs during creation of the swap file, no disk LUs are 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 are those set at generation.

`BG,`*boundary* — Set the boundary where priority swapping is to begin. If this command is not given, the value is that set at generation.

`SS,`*newcode* — Change the system security code from the value in the system file to the newcode.

`AC[,ON|OFF|OA]` — For systems with VC+ only. Turns session accounting on (ON), off (OFF), or on without logoff end-of-session information displayed (OA). AC with no parameters turns accounting on. If this command is not given, the session accounting state carries over from the last time the system booted.

Note that the system utility METER uses information gathered by the session accounting system; session accounting must be enabled for METER to function properly.

EN (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.

Example of a 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 disk the system file that is being booted. After the first time through the boot process, the system file has 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 is not 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 disk 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.

K

Installing Localizable Programs

This appendix discusses the system generation and installation considerations for systems or applications designed to be localizable, such as HP 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 disk-based message files. In one method, the program accesses the disk-based message catalog file directly. Access to the disk-based binary messages is slow, but the memory requirement is reduced. The other disk-based method uses the message monitor (MSG.M) system utility program to access the disk-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 disk-based access. For each supported native language, there is a different disk-based message catalog in the /CATALOGS directory.

The custom formatting tables are included in the LANGDEF files, the disk-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):

.S*nnn* – source message catalogs

.R*nnn* – relocatable binary message catalogs

.C*nnn* – disk-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 disk-based binary catalog files.

System Generation and Installation

This section discusses the generation and installation of disk-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.

Relocating the Appropriate Driver

In the Driver Partitions phase of the answer file, relocate the serial interface drivers and the device drivers appropriate to your system and terminal configuration.

Configuring Terminals for 8-Data Bit Support

In the Table Generation phase of the answer file, you need to modify the DVT command for the serial device driver for each terminal to be configured for any lingual configuration other than Native/1000. All lingual configurations other than Native/1000 require 8 bit data support. Refer to the *RTE-A Driver Reference Manual,* part number 92077-90011, for a description of the terminal configuration formats for each device driver. In addition, refer to the Table Generation Phase Chapter in this manual.

Relocating the System Message Relocatables

In the memory allocation phase, in the section headed by system messages, you must relocate %MSGTB, the system message module containing all of the system message pointers. Following this, you must relocate the message relocatable %$M000 containing the system error messages in Native/1000. At this point, you may relocate up to seven additional lingual system message relocatables.

The following example shows the relocation of %$M003, the system message relocatable for Danish.

(the first column of system generation answer file)

```
|
|
|
*
re /rtea/%msgtb,,
end
re /rtea/%$m000,,
end
re /rtea/%$m003,,
end
end
*
```

Place an END statement after each module, and a second END statement after the last module to signal the end of the system message relocatable section. The order in which the message catalogs are relocated defines the language for each system language number (SLN). In the above example, %$M000 defines SLN 0 to Native/1000, and %$M003 defines SLN 1 to be Danish. SLNs 2 through 7 default to the same language as SLN 0.

The system message catalog in languages other than Native/1000 are localized versions of the catalog <$M000. Contact HP about the availability of localized system message catalogs.

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*.H*nnn* into /SYSTEM directory (where *nnn* is the language ID number).

4. Set up a directory /HELP*xxx* and place localized help messages into it (where *xxx* is the language ID number).

5. Set up a directory /PROGRAMS*xxx* 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*.C*nnn* into a directory /CATALOGS.

The LANGDEF files (>LG*xxx*) are the binary message catalogs that define table-driven language-dependent information read by a library routine NlInfo. The binary message catalog >MS*xxx* is an error message catalog for the message monitor MSG.M.

Both the >LG*xxx* and >MS*xxx* 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*.H*nnn* in the /SYSTEM Directory

Localized help files with the format *xxxxx*.H*nnn*, 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*.H*nnn* are localized versions of Native/1000 files with the form *xxxxx*.HLP.

Creating Directory /HELP*xxx* for Localized Help Files

Create a directory /HELP*xxx*, 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 /PROGRAMS*xxx*

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 /PROGRAMS*xxx* rather than in the directory /PROGRAMS, with some exceptions. The following programs must be in the directory, /PROGRAMS:

1. Programs PM*xxx*.RUN which are localized versions of PROMT.RUN.

2. A program LOGON.RUN which does not have an independent local version.

3. Programs that 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] [languageID]]
```

parameters:

n LU number

languageID 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 /PROGRAMS*xxx* (where *xxx* represents the language ID) is searched before the directory /PROGRAMS. The directory /PROGRAMS*xxx* contains corresponding lingual versions of the programs. Refer to the *RTE-A User's Manual*, part number 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 K-1 lists the localizable system and subsystem files, the names of their source and binary catalogs, and their proper locations in various directories.

Table K-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	
MSG.M	cat	<MS000	>MS000	/programs	0)
NLSID	rel	<ID000	%ID000	/programsxxx	
NLTM	rel	<TM000	%TM000	/programsxxx	
OS ERLOG	sys	<$M000	%$M000	(OS message map)	
CI	rel	CI.S000	CI.R000	/programsxxx	CDS (VC+)
		CINC.S000	%CI000		non-cds
		CISUB.S000	CISUB.R000		for CISUB
		CROUT.S000	%CR000		for CI LIB
CIX	rel	CIX.S000	CIX.R000	/programsxxx	CDS (VC+)
		CIXNC.S000	%CX000		non-CDS
DL	rel	<DL000	%DL000	/programsxxx	
IO	rel	<IO000	IO.R000	/programsxxx	
LI	rel	LI.S000	LI.R000	/programsxxx	
OUTPT	rel	<PT000	%PT000	/programsxxx	SPOOL
RESTR	rel	<RE000	%RE000	/programsxxx	
RS	rel	<RS000	%RS000	/programsxxx	
SAM	rel	SAMU.S000	SAMU.R000	/programsxxx	
SP	rel	<SP000	%SP000	/programsxxx	SPOOL
WH	rel	<WH000	WH.R000	/programsxxx	
GRUMP	rel	GRUMP.S000	GRUMP.R000	/programsxxx	
KILLSES	rel	<KI000	%KI000	/programsxxx	
SESLU	rel	<SL000	%SL000	/programsxxx	
RINFO	rel	<RI000	%RI000	/programsxxx	
SINFO	rel	<SI000	%SI000	/programsxxx	
LOGON	rel	LOGON.S000	LOGONSEC.R000	/programsxxx	1)
PROMT	rel	<PM000	%PM000	/programs	2)
PMxxx	rel	<PMxxx	%PMxxx	/programs	3)
D.ERR	rel	<DE000	%DE000	/programs	1)
CLOSE	rel	<CL000	%CL000	/programs	
LINK	cat	LINK.S000	LINK.C000	/programs	
EDIT	rel&hlp	<ED000	%ED000	/programs	
FTN7x	rel	<FX000	%FX000	/programsxxx	
DEBUG	cat	DEBUG.S000	DEBUG.C000	/programs	
XDB	cat	XDB.S000	XDB.C000	/programs	
MAIL	cat	MAIL.S000	MAIL.C000	/programs	
NEWMAIL	rel	NEWMAIL.S000	NEWMAIL.R000	/programsxxx	

0) This program has a system attribute and multiple lingual versions of catalogs should be implemented.

1) This program has a system attribute and multiple lingual versions of relocatable catalogs must be relocated.

2) Localized version of PROMT should be PMxxx which should be placed into directory /programs.

Table K-1. Localizable System and Subsystem Files (continued)

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	FOWN.R000	/programsxxx	
FPACK	rel	<FP000	%FP000	/programsxxx	
FREES	cat	<FR000	FREES.C000	/programs	
FST	rel &	<FS000	%FS000	/programsxxx	
	cat		>FS000	/programs	
FVERI	rel	<FV000	%FV000	/programsxxx	
MERGE	rel	MERGE.S000	MERGE.R000	/programsxxx	
MPACK	rel	MPACK.S000	MPACK.R000	/programsxxx	
PRINT	rel	<PR000	%PR000	/programsxxx	PRINT
PRIN0	rel	<P0000	%P0000	/programsxxx	PRINT
SCOM	cat	SCOM.S000	SCOM.C000	/programs	
TF	cat	<TF000	>TF000	/programs	
IMAGE/1000-II					
QUERY	rel & cat	QY000.CAT	QY000.REL & QY000.C000	/programsxxx	5)
DBDS	rel	DB000.CAT	DB000.REL	/programsxxx	
DBUTL	rel&hlp	UT000.CAT	UT000.REL	/programsxxx	
DBLOD	rel	L0000.CAT	L0000.REL	/programsxxx	
DBSTR	rel	ST000.CAT	ST000.REL	/programsxxx	
DBRBR	rel	RB000.CAT	RB000.REL	/programsxxx	
DBRFR	rel	RF000.CAT	RF000.REL	/programsxxx	
DBARC	rel	AR000.CAT	AR000.REL	/programsxxx	
DBSPA	rel	SA000.CAT	SA000.REL	/programsxxx	
DBBLD	rel	BL000.CAT	BL000.REL	/programsxxx	
DBULD	rel			/programsxxx	4)
DBRST	rel			/programsxxx	4)
DBMX.LIB	rel	LB000.CAT	LB000.REL		4)
DBMON			DBERR.C000	/programs	3)
DBCLN				/programs	3)

3) Not localizable.

4) Necessary message modules must be relocated before DBMX.LIB which includes LB000.REL.

5) The sources message catalog contains two target types, RELOCATABLE and CATALOG.

Error Messages

NLSID: NlInfo error number <*error number*>.

This message is issued when no language ID parameter is given, but NlInfo 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: NlInfo error <*error number*>.

This message is issued when NlInfo 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 NlInfo 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 NlInfo 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 NlTermCap file has a line that starts with NL but does not have a ":*lu*" after it.

NLSID: Language number <*lang#*> not supported on LU <*lu #*>.

This messages is issued then the NlTermCap 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