

SC34-0642-0

Event Driven Executive Operation Guide

Version 5.0

**Library Guide and
Common Index**

SC34-0645

**Installation and
System Generation
Guide**

SC34-0646

**Operator Commands
and
Utilities Reference**

SC34-0644

**Language
Reference**

SC34-0643

**Communications
Guide**

SC34-0638

**Messages and
Codes**

SC34-0636

Operation Guide

SC34-0642

**Event Driven
Language
Programming Guide**

SC34-0637

**Reference
Cards**

SBOF-1625

**Problem
Determination
Guide**

SC34-0639

**Customization
Guide**

SC34-0635

**Internal
Design**

LY34-0354

SC34-0642-0

Event Driven Executive Operation Guide

Version 5.0

**Library Guide and
Common Index**

SC34-0645

**Installation and
System Generation
Guide**

SC34-0646

**Operator Commands
and
Utilities Reference**

SC34-0644

**Language
Reference**

SC34-0643

**Communications
Guide**

SC34-0638

**Messages and
Codes**

SC34-0636

Operation Guide

SC34-0642

**Event Driven
Language
Programming Guide**

SC34-0637

**Reference
Cards**

SBOF-1625

**Problem
Determination
Guide**

SC34-0639

**Customization
Guide**

SC34-0635

**Internal
Design**

LY34-0354

First Edition (December 1984)

Use this publication only for the purpose stated in the Preface.

Changes are made periodically to the information herein; any such changes will be reported in subsequent revisions or Technical Newsletters.

This material may contain reference to, or information about, IBM products (machines and programs), programming, or services that are not announced in your country. Such references or information must not be construed to mean that IBM intends to announce such IBM products, programming, or services in your country.

Publications are not stocked at the address given below. Requests for copies of IBM publications should be made to your IBM representative or the IBM branch office serving your locality.

This publication could contain technical inaccuracies or typographical errors. A form for readers' comments is provided at the back of this publication. If the form has been removed, address your comments to IBM Corporation, Information Development, 3406, P. O. Box 1328, Boca Raton, Florida 33432. IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation whatever. You may, of course, continue to use the information you supply.

Summary of Changes for Version 5

The following changes have been made to procedures in this document.

- The procedure “Creating a Stand-Alone or \$TRAP Dump Diskette” on page OP-207 was updated to reflect that the diskettes you create can be used for \$TRAP or stand-alone dumps.
- The procedure “Printing a Stand-Alone or Formatted \$TRAP Dump” on page OP-293 was updated to reflect that a \$TRAP dump can be on more than one diskette.
- The procedures “Saving the System on Tape” on page OP-325 and “Restoring from Tape” on page OP-338 have been updated to reflect the single-buffer and double-buffer functions. The double-buffer function stores to tape faster and allows the system to vary-on and initialize tapes automatically.

The following new procedures have been added to this document.

- The procedure “Transmitting Data from System to System” on page OP-148 shows you how to change source and target systems so that you are able to transfer data sets and volumes, using the new \$TRANS utility, from a remote-to-local or local-to-remote system.
- The procedure “Renaming a Diskette Volume Label and Owner Id” on page OP-134 lists instructions on how to rename a volume label and owner id on a diskette.
- The procedure “Listing Your Log Data Set on the Console” on page OP-297 lists instructions on how to list program and system checks, recorded in your log data set, on the console.
- The procedure “Copying a \$TRAP Data Set to Two Diskettes” on page OP-299 shows you how to copy a \$TRAP disk data set, which is greater than 512K, to 2 diskettes.

Summary of Changes for Version 5

Miscellaneous updates were made throughout this document to include the following new hardware:

- 4975-01A printer
- 4980 terminal
- 5219 printer.

About This Book

This book is a guide to operating your IBM Series/1 with the Event Driven Executive system installed. It contains procedures for doing many of the common operations jobs.

Audience

This book is intended for anyone who has to operate the IBM Series/1 with the Event Driven Executive. Readers should have a basic understanding of computer terminology and operation before using this book.

How this Book is Organized

The book is divided into the following 9 chapters and 2 appendixes.

- *Chapter 1. Introduction to Series/1 Operation* contains an overview of the topics covered in this book.
- *Chapter 2. Getting Started* explains how to get ready to use the Series/1.
- *Chapter 3. Operating Your Display Terminal* explains the basics of operating a display terminal, and how to change some of the operating characteristics.
- *Chapter 4. Using the Session Manager* shows you how to use the session manager facility to help you do some of your jobs on the Series/1.

About This Book

How this Book is Organized (*continued*)

- *Chapter 5. Storing and Handling Information* explains how information is stored on the Series/1 and contains procedures for handling the information and the media on which it is kept.
- *Chapter 6. Running Programs and Job Procedures* shows how to run and control EDX and application programs.
- *Chapter 7. Controlling Program Output* contains procedures for controlling printed output generated by application programs.
- *Chapter 8. Responding To Problems* is a guide to gathering information useful in solving problems with EDX and application programs.
- *Chapter 9. Keeping Records and Backing-up the System* contains some suggested records that should be kept about the system and data and programs on it. This chapter also has procedures for making a copy of the operating system and application programs as security against something happening to the Series/1 or copies of the programs that are on it.
- *Appendix A. Reference Material* contains a cross reference of the jobs that can be done using the system utilities and operator commands, a chart of the \$FSEDIT PF keys and edit commands, the syntax for the system operator commands and \$JOBUTIL commands, explanations of the spool status display screens, and helps for determining the size of a spool data set.
- *Appendix B. Blank Forms* contains blank copies of the record-keeping forms suggested in Chapter 9.

Aids to Using this Book

This book has the following aids to using the information it contains:

- A chart of the main jobs that are done on the Series/1 with EDX. This chart is found under “Job Cross Reference Chart” on page OP-346. It lists the operator command or system utility program that is used for the job, the session manager option numbers to help you access the utility with the session manager, and the prefix of the guide in the EDX library that contains instructions for doing the job.
- A table of contents that lists the main headings in this book.
- An index of the topics covered in this book.
- A glossary that defines terms and acronyms used in this book and in other EDX library publications.
- A guide to using the procedures presented in this book. This guide is found under “Reading Operation Procedures” on page OP-5.

If you are new to Series/1 Event Driven Executive operation, you should become familiar with the material in Chapters 1 through 5 before using the system.

A Guide to the Library

Refer to the *Library Guide and Common Index* for information on the design and structure of the Event Driven Executive library, for a bibliography of related publications, and for an index to the complete library.

Contacting IBM about Problems

You can inform IBM of any inaccuracies or problems you find when using this book by completing and mailing the **Reader's Comment Form** supplied at the end of the book.

If you have a problem with the Series/1 Event Driven Executive services, fill out an authorized program analysis report (APAR) form as described in the *IBM Series/1 Software Service Guide*, GC34-0099.



Contents

Chapter 1. Introduction to Series/1 Operation	OP-1
The Operation Jobs	OP-2
Getting Started	OP-2
Operating Your Display Terminal	OP-2
Using the Session Manager	OP-3
Storing and Handling Information	OP-3
Running Programs and Job Procedures	OP-3
Controlling Program Output	OP-4
Responding to Problems	OP-4
Keeping Records and Making Backup Copies of Your System	OP-4
Reading Operation Procedures	OP-5
Chapter 2. Getting Started	OP-7
Series/1 Hardware	OP-8
Switching Power On and Off	OP-9
Switching Power On	OP-9
Switching Power Off	OP-10
Switching Power Off in an Emergency	OP-10
Restoring Power After an Emergency	OP-11
Using Diskettes	OP-12
Handling Diskettes	OP-13
Inserting Individual Diskettes	OP-15
Inserting 4966 Magazines	OP-16
Removing Individual Diskettes	OP-19
Removing Diskette Magazines	OP-20
Loading the System (Initial Program Load)	OP-22
The Basic Console	OP-22
Manual IPL Procedure	OP-24
Automatic IPL Procedure	OP-25

Contents

Setting the Time and Date OP-26
Displaying the Date and Time OP-27

Chapter 3. Operating Your Display Terminal OP-29

Switching Power On OP-30
Adjusting Screen Brightness OP-30
Moving the Cursor OP-32
Entering and Changing Characters OP-34
 Entering New Characters OP-34
 Erasing with the Space Bar OP-34
 Replacing and Deleting Characters OP-35
 Inserting Characters OP-36
Erasing Data from the Screen OP-37
 Erasing to the End of the Field OP-37
 Erasing to the End of the Line OP-38
 Erasing the Screen OP-39
Printing the Terminal Display Screen OP-40
Entering Commands and Answering Prompts OP-41
 Entering Operator Commands OP-41
 Responding to Prompts OP-42
Clearing the Display Screen OP-43
Listing System Terminal Assignments OP-44
Sending a Message to Another Terminal OP-45
Renaming a Terminal OP-47
Reassigning a Terminal Address OP-48
Changing the Hard-Copy Device OP-49
Changing a Terminal's Partition Assignment OP-50
Changing the Display Screen Format OP-51
Varying a Terminal Offline OP-52
Varying a Terminal Online OP-54
Changing Display Terminal Keyboard Values OP-55
 Upper and Lowercase Characters OP-56
 Changing 4978 and 4980 Keyboard Values OP-62

Chapter 4. Using the Session Manager OP-71

Introduction to the Session Manager OP-72
 Menus OP-72
 Utilities Supported OP-74
Loading the Session Manager OP-77
 Loading with \$L OP-77
 Loading Automatically OP-78
Logging On OP-79
Selecting an Option OP-80
Suspending and Restarting the Session Manager OP-82
 Suspending OP-82
 Restarting OP-83
Clearing the Screen OP-83
Returning to the Previous Screen OP-83

Returning to the Primary Option Menu OP-83
Logging Off OP-84

Chapter 5. Storing and Handling Information OP-85

How Information is Stored OP-86

Records OP-86
Data Sets OP-87
Volumes OP-87
Directories OP-88
Volume Label and TAPEID OP-88

Diskette Formats OP-90

Preparing Disks, Diskettes, and Tapes for Use OP-91

Initializing Diskettes OP-92

Initialize for EDX Format OP-93
Initialize to IBM Standard for Information Interchange OP-97
Initialize to Bytes Per Sector Format OP-102
Initialize a Diskette Volume Directory OP-106
Initialize an H Exchange Volume OP-111

Initializing Tapes OP-112

Changing Tape Drive Attributes OP-118

Allocating Volumes and Data Sets OP-121

Allocate a Volume OP-121
Allocate a Data Set OP-124
Allocate an H Exchange Data Set OP-126
Allocate Tape Data Sets OP-127

Renaming Volumes and Data Sets OP-131

Rename a Volume OP-131
Rename a Data Set OP-132

Renaming a Diskette Volume Label and Owner Id OP-134

Deleting Volumes and Data Sets OP-136

Delete a Volume OP-136
Delete One Data Set OP-137
Delete Multiple Data Sets OP-139

Compressing Volumes and Devices OP-142

Copying Data on EDX Systems OP-148

Copy a Data Set with Allocation OP-152
Copy a Volume or Data Set without Allocation OP-162
Copy to or from a Tape Data Set OP-167

Copying Data to/from a Non-EDX System OP-171

Copy From a Basic Exchange Data Set OP-171
Copy to a Basic Exchange Data Set OP-173
Copy from an H Exchange Data Set OP-175
Copy to an H Exchange Diskette OP-178

Listing Volume and Data Set Directory Information OP-181

List All Volumes OP-181
List All Volumes on One Device OP-183
List All Data Sets on All Volumes OP-185
List One or More Data Sets on a Volume OP-187

Contents

- List Data- or Program-Type Data Sets on a Volume OP-191
- Listing the Contents of a Data Set OP-195
 - List the Contents of a Data-Type Data Set OP-195
 - Dump the Contents of a Data-Type Data Set OP-198
 - Dump the Contents of a Program-Type Data Set OP-202
- Determining Volume Size and Free Space OP-205
- Creating a Stand-Alone or \$TRAP Dump Diskette OP-207
- Entering/Editing Data with \$FSEEDIT OP-210

Chapter 6. Running Programs and Job Procedures OP-215

- Preparing to Run a Program OP-216
- Loading Programs OP-217
 - Using the \$L Operator Command OP-217
 - Using the Session Manager OP-219
 - Using the Command Stream Processor, \$JOBUTIL OP-222
- Checking and Controlling Programs OP-227
 - Listing Partitions and Active Programs OP-227
 - Controlling Job Queue Jobs OP-228
 - Responding to a \$JOBUTIL Pause OP-235
- Cancelling Programs OP-236
- Deleting Job Queue Jobs OP-237
 - Deleting One Job OP-237
 - Deleting All Job Queue Jobs OP-239
 - Ending Job Queue Processing OP-240
- Coding a \$JOBUTIL Procedure OP-241
 - Planning a Job Procedure OP-241
 - Entering a Procedure OP-243
 - Example Procedure OP-245
- Running the Job Queue Processor \$JOBQ OP-247

Chapter 7. Controlling Program Output OP-251

- Aligning Forms OP-252
- Advancing Printer Forms OP-252
- Changing Page Formatting Parameters OP-253
- Spooling Program Output OP-258
 - Printer Spooling Terms OP-258
 - Preparing to Run Spool OP-260
 - Setting Spool Options OP-261
 - Running the Spool Facility OP-273

Chapter 8. Responding to Problems OP-287

- Recording Problem Symptoms OP-288
- Taking a Stand-Alone Dump of Storage OP-290
- Printing a Stand-Alone or Formatted \$TRAP Dump OP-293
- Listing Your Log Data Set on the Console OP-297
- Copying a \$TRAP Data Set to Two Diskettes OP-299
- Dumping Tapes to a Printer or Display Terminal OP-302
- Determining the Software Version Level OP-306

Determining Version Level with \$D OP-306
Determining the Version Level from a Dump of Partition 1 OP-308
Determining the Version Level of Programs OP-308
Restarting the System OP-311

Chapter 9. Keeping Records and Backing-up the System OP-313

Keeping Records OP-314
 Operations Log OP-314
 Job Instructions OP-315
 Diskette and Tape Log OP-316
 Backup Log OP-317
 Procedures Log OP-317
 Listing the Devices on Your System OP-318
Saving the EDX System OP-319
 Saving the System on Diskettes OP-319
 Saving the System on Tape OP-325
Restoring the System OP-334
 Restoring from Diskette OP-334
 Restoring from Tape OP-338

Appendix A. Reference Material OP-345

Job Cross Reference Chart OP-346
\$FSEDIT PF Keys and Edit Commands OP-353
 \$FSEDIT PF Keys OP-353
 \$FSEDIT Edit Commands OP-354
Operator Command Syntax OP-357
 Syntax Conventions OP-357
 \$A - List Partitions and Active Programs OP-357
 \$B - Blank Display Screen OP-358
 \$C - Cancel Program OP-358
 \$CP - Change Display Terminal's Partition Assignment OP-359
 \$D - Dump Storage OP-360
 \$E - Eject Printer Page OP-360
 \$L - Load a Program or Utility OP-361
 \$P - Patch Storage OP-362
 \$S - Control Spooled Program Output OP-362
 \$T - Set Date and Time OP-372
 \$U - Invoke Your Own Operator Command OP-372
 \$VARYOFF - Set Device Offline OP-373
 \$VARYON - Set Device Online OP-374
 \$W - Display Date and Time OP-375
\$JOBUTIL Command Syntax OP-376
 AL — Allocate Data Set OP-376
 DE — Delete Data Set OP-377
 DS — Identify Data Set OP-378
 EJECT — Start New Log Page OP-379
 EOJ — End of Job OP-379
 EOP — End of Procedure OP-380

Contents

EXEC — Execute Program	OP-380
JOB — Identify Job	OP-381
JUMP — Jump to Label	OP-382
LABEL — Identify Continuation Point	OP-383
LOG — Log Control	OP-384
NOMSG — No Message Logging	OP-385
PARM — Pass Parameter	OP-386
PAUSE — Await Manual Intervention	OP-387
PROC — Execute Procedure	OP-388
PROGRAM — Identify Program	OP-389
REMARK — Display Remark	OP-390
* — Comment	OP-390
Spool Status Display Formats	OP-391
\$SPLUT1 Status Display	OP-391
Spool Job Display Format	OP-392
Spool Writer Display Format	OP-393
Spool Resources Display Format	OP-394
Estimating Spool Job Print Time	OP-395
Estimating Spool Data Set Size	OP-395
Estimating \$EDXASM and \$S1ASM Output Listing Sizes	OP-395
Estimating \$EDXLINK Output Listing Size	OP-396
Estimating Application Program Output Listing Size	OP-396
Estimating COBOL, FORTRAN, and PLI Output Listing Sizes	OP-396
Appendix B. Blank Forms	OP-397
\$JOBUTIL Procedure Planning Guide	OP-399
\$JOBUTIL Procedures Log	OP-403
Backup Log	OP-407
Diskette Log	OP-411
Job Instructions	OP-415
Operations Log	OP-419
Problem Recording Form	OP-423
Tape Log	OP-431
Glossary of Terms and Abbreviations	OP-435
Index	OP-445

Figures

1. IBM Series/1 Hardware OP-8
2. 4977 Rack Enclosure Circuit Breaker Location OP-11
3. IBM Diskette Parts OP-12
4. Basic Processor Console OP-23
5. Series/1 Display Terminals OP-31
6. 3101 Setup Switches OP-57
7. 4978 Display Station Keyboard, RPQ D02056 OP-64
8. Control Chart for 4978 Display Station, RPQ D02056 OP-65
9. Session Manager Options by Utility OP-74
10. Series/1 Data Organization OP-89
11. Diskette Format Combinations OP-92
12. Data Set Directory Information OP-187
13. \$JOBUTIL Procedure Guide OP-242
14. Example \$JOBUTIL Procedure OP-246
15. Example Nested \$JOBUTIL Procedure OP-246
16. Problem Record Form OP-288
17. Storage Dump, EDXFLAGS OP-308
18. Operations Log OP-314
19. Job Instructions OP-315
20. Diskette Log OP-316
21. Tape Log OP-316
22. Backup Log OP-317
23. Procedures Log OP-317
24. Job Cross-reference Chart OP-346
25. \$FSEDIT Program Function Keys OP-353
26. \$FSEDIT Command Summary OP-354



Chapter 1. Introduction to Series/1 Operation

Operating your IBM Series/1 Event Driven Executive (EDX) system involves running the Series/1, using the EDX support programs, monitoring and controlling application programs, controlling the printing of program output, responding to problems, and keeping records.

This operation guide contains step-by-step instructions, or procedures, for doing most of the daily jobs that use the EDX program support. In some instances, it explains the basic concepts to help you understand the job better. It also contains enough information about the Series/1 computer hardware to help you get started.¹

The Series/1 with EDX installed is operated by people with many different backgrounds and job requirements. Therefore, this book should be kept near the Series/1 so the person using the Series/1 can refer to it as needed. If you need additional information not covered in this book, such as about a job it doesn't cover, refer to its companion book, the *Operator Commands and Utilities Reference*. The *Operator Commands and Utilities Reference* contains reference information about all the EDX utilities.

This chapter provides a general introduction to the operation jobs and explains how to use the procedures in this book under the following headings:

- “The Operation Jobs” on page OP-2
- “Reading Operation Procedures” on page OP-5

¹ Because of the variety of devices that can be attached to a Series/1, this book cannot contain all there is to know about each one. You must refer to the *IBM Series/1 Operator's Guide*, GA34-0039 or the hardware description manual for an individual device for additional information on that device.

Introduction to Series/1 Operation

The Operation Jobs

The individual operation procedures are divided into categories as follows. Each category corresponds to a chapter in this book.

- Getting Started
- Operating your display terminal
- Using the session manager
- Storing and handling information
- Running programs and job procedures
- Controlling program output
- Responding to problems
- Keeping records and making backup copies of your system.

Getting Started

If this is the first time you have used your Series/1, you should become familiar with the devices, or hardware, that make up your Series/1. There are many different combinations of devices that can make up a Series/1. Chapter 2, "Getting Started" on page OP-7 shows you an average system with the main devices that are common to most Series/1's. It also contains procedures for switching the electrical power to these devices on and off. If you need more detailed information about a specific device, you must refer to the *IBM Series/1 Operator's Guide*, GA34-0039 or the hardware description manual for the device.

Getting started also includes starting (loading) the EDX operating system so you can use the Series/1, using diskettes, putting paper into the printers and, if you have one, loading tape onto the tape drive. Chapter 2, "Getting Started" on page OP-7 shows you how to do the first two. Refer to the *Operator's Guide* or the hardware description manual for each individual device to learn how to put paper in the printers and load tapes onto a tape drive.

Operating Your Display Terminal

The display terminal is the main tool for operating the Series/1. Operation of your display terminal requires adjusting the intensity of the screen, moving the cursor, setting the PF keys, and changing the display terminal options that are under program control.

Chapter 3, "Operating Your Display Terminal" on page OP-29 shows you how to use your display terminal.

The Operation Jobs (*continued*)

Using the Session Manager

You can use the session manager to help you operate the Series/1. It is one way you can access system utilities and application programs from a display terminal. (System utilities are EDX programs that do everyday jobs. Programs are collections of computer instructions that tell the Series/1 what to do. Application programs are programs written to make the Series/1 do a specific job for your installation.)

Chapter 4, "Using the Session Manager" on page OP-71 contains an introduction to the session manager and shows you how to use it.

Storing and Handling Information

Storing and handling information (data) involves preparing the device that will store the information, setting up a data "filing" system, moving the data around, making copies of it, and entering and/or editing the data.

Chapter 5, "Storing and Handling Information" on page OP-85 explains how data is stored on the Series/1 and contains procedures for handling stored data.

Running Programs and Job Procedures

The main activity you do when using the Series/1 is to run programs and job procedures. You are running programs even when you do most of the other jobs described in this book.

A program is a collection of computer language instructions that tell the Series/1 what to do. A job procedure is a series of special commands that you set up to run multiple programs for you.

The job of running programs and procedures involves loading them into storage, checking and controlling their progress, and, if necessary, cancelling them. For procedures, you also have to write them. Chapter 6, "Running Programs and Job Procedures" on page OP-215 shows you how to do these jobs.

Running programs also includes controlling the output. The job of controlling program output is described in Chapter 7, "Controlling Program Output" on page OP-251.

Introduction to Series/1 Operation

The Operation Jobs (*continued*)

Controlling Program Output

Program output can be listed on either a printer or a display terminal. Controlling that output includes aligning the paper in the printer, changing page formatting parameters on either the printer or display terminal screen (a display screen is sometimes called a page), and spooling the output. Spooling the output means directing it to a data set for printing by an EDX system program at a later time. This allows the program generating the output to finish sooner because EDX can store output in a data set faster than it can be printed or displayed.

Chapter 7, "Controlling Program Output" on page OP-251 contains procedures for controlling printer output.

Responding to Problems

You will occasionally have problems when you are operating your computer. These problems may be simple, such as a typing error made when you enter an operator command, or more complex, such as an error in an application program. Whenever you encounter problems when operating the EDX system on the Series/1, you should gather as much information about the problem as possible. Chapter 8, "Responding to Problems" on page OP-287, helps you gather that information.

Keeping Records and Making Backup Copies of Your System

Keeping accurate records helps you improve the efficiency of the operation of your computer and helps you recover should an error occur. Making regular copies of your system and important data provides a point of reference should you experience a problem that destroys the copy of EDX or data that is on the Series/1. Chapter 9, "Keeping Records and Backing-up the System" on page OP-313, contains some suggestions on records that should be kept of your system and a procedure for making a backup copy of your EDX system and data.

Reading Operation Procedures

The procedures in this book are designed to be used step-by-step or for reference. Each procedure has an appropriate introduction. The introduction either provides background information and concepts, or simply tells you what the procedure is used for.

If there are any requirements or conditions, such as obtaining the address of a device or the name of a data set, they are listed under the heading "Before You Start". The "Before You Start" information should always be reviewed before starting the procedure.

Each step of the instructions contains an explanation of what must be done. This may simply be the command to enter, or a question that you must consider followed by the possible responses. When necessary, an explanation is included to help you decide which response to use.

The first time you use a procedure, you should read the complete procedure first, then follow the instructions for each step as you do the job. Once you are familiar with the job, you may want to just scan the example screens, referring to the explanations only when you are unsure of a particular step. As you become more experienced and do the jobs without using the procedures, you can scan the example screens to find your place should you encounter any problems.

The following example procedure includes an explanation of the display screen examples used in this book.

Example Procedure

1. Description of the first action to be done or decision to be made.

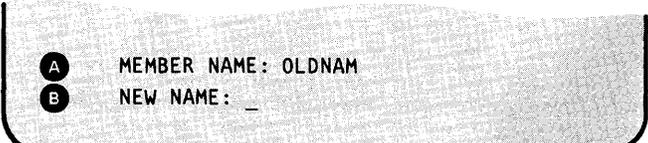
The display screen contents shown with each step shows the system's prompt (request) for information **A**, an example of the information you are entering **B**, and the system's response after you press the enter key **C**. The examples of information that you enter, your responses, are always shown in red.



A terminal window with a light gray background. On the left side, there are three red circular markers labeled A, B, and C. A: COMMAND (?): B: RE C: MEMBER NAME: _

2. Description of the second action to be done or decision to be made.

If the system's response is another prompt, the next example repeats it with an example of the information that should be entered **A**, followed by the next system response **B**.



A terminal window with a light gray background. On the left side, there are two red circular markers labeled A and B. A: MEMBER NAME: OLDNAM B: NEW NAME: _

Chapter 2. Getting Started

Before you can use your Series/1 to do a job, you must be familiar with the various devices that make up a Series/1 system, and you must know how to switch on the electrical power, how to handle diskettes, and how to do an initial program load (IPL). These jobs are described in this chapter under the following headings:

- “Series/1 Hardware” on page OP-8
- “Switching Power On and Off” on page OP-9
- “Using Diskettes” on page OP-12
- “Loading the System (Initial Program Load)” on page OP-22.

You may also have to put paper in the printer and load a tape onto the tape drive. For instructions on how to do these two tasks, refer to the *IBM Series/1 Operator's Guide*, GA34-0039.

You must also refer to the *IBM Series/1 Operator's Guide*, GA34-0039 and the manuals for the individual devices for hardware operation instructions not included in this book.

Getting Started

Series/1 Hardware

The IBM Series/1 is a compact, general-purpose computer system. It consists of different compact units, or hardware devices. Some of these devices, such as the 4955 processor and the 4964 diskette unit, fit into a common frame called a 4997 rack enclosure. Others, such as the 4979 display station and the 4974 printer, are self-contained. They are either free-standing or are placed on a table. See Figure 1 for a picture of a typical Series/1 hardware arrangement.

Because the Series/1 can consist of different hardware devices, you should become familiar with the operation of the devices attached to your system by referring to the *IBM Series/1 Operator's Guide*, GA34-0039, and the individual manuals for those devices.

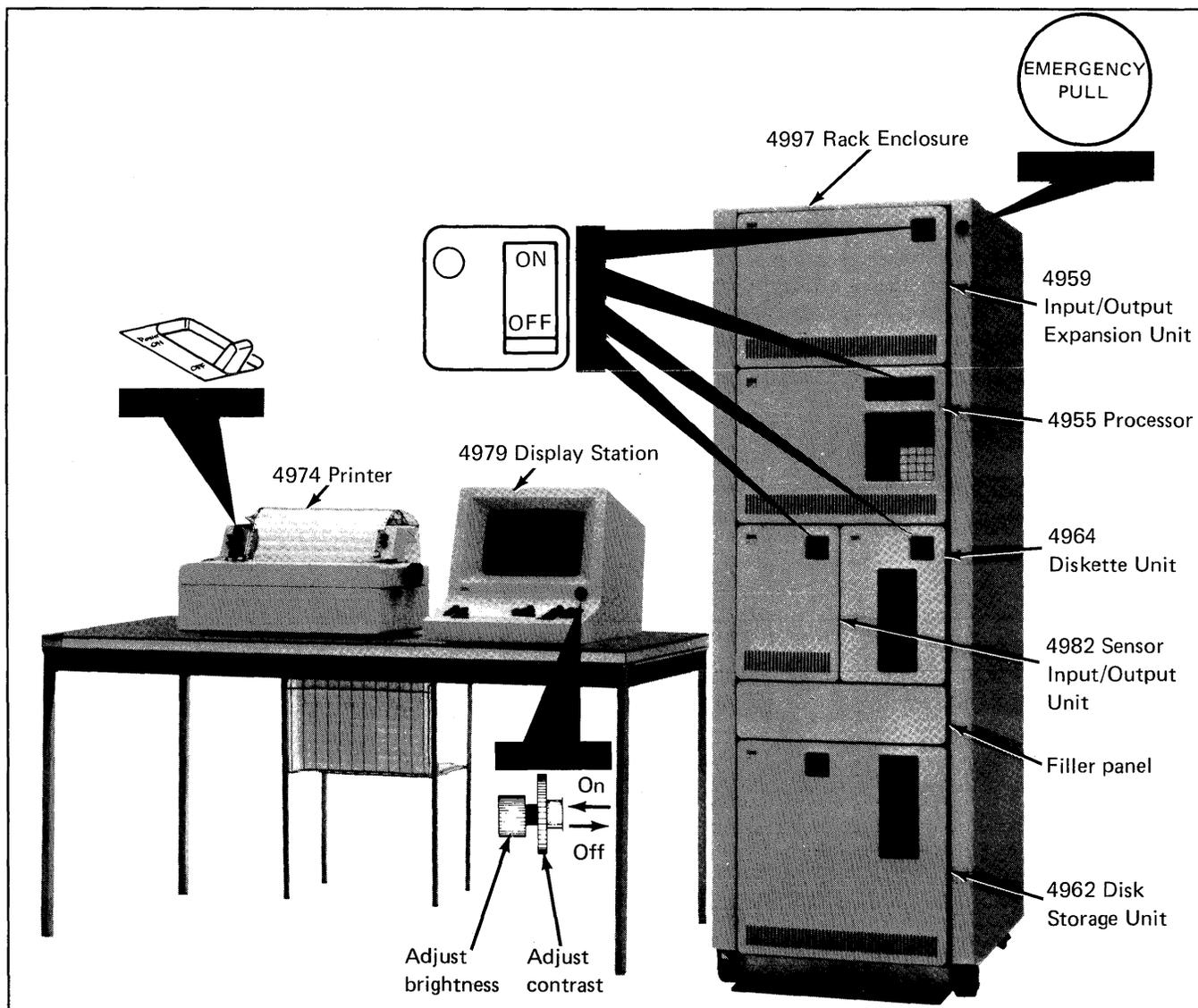


Figure 1. IBM Series/1 Hardware

Switching Power On and Off

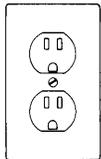
Each hardware device has its own electrical power switch. In an emergency, you can turn off all electrical power to the rack enclosure by pulling the Emergency Pull switch.

This section contains the following four procedures:

- Switching the power on
- Switching the power off
- Switching the power off in an emergency
- Restoring power after an emergency.

Switching Power On

To switch on electrical power, place each power switch in the **ON** position. Figure 1 on page OP-8 shows some typical power switch locations. Switch the power on in the following sequence:



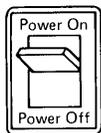
1. Make sure all the power cords from the rack enclosure(s), table top units, and free-standing units are plugged into the wall outlets.



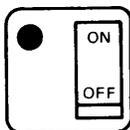
2. Make sure the Emergency Pull switch is pushed in. If it is not, see "Switching Power Off in an Emergency" on page OP-10 before proceeding.



3. If you have a 4993-1 in the system enclosure, set the Enable/Disable switch to the Disable position.



4. Switch on all display terminal, printers, and other table-top and free-standing devices. They can be switched on in any order.



5. Switch on the devices in the rack enclosure(s). You must switch on the processor unit last, unless you have a 4993-1 Series/1-System/370 Termination Enclosure; switch it on after the processor unit.

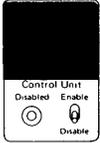
Getting Started

Switching Power On and Off (*continued*)

Switching Power Off

Use this procedure to switch off electrical power.

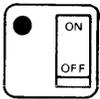
Note: Remember, use the Emergency Pull switch **only in an Emergency**.



1. If you have a 4993-1 Series/1-System/370 Termination Enclosure on your system:
 - a. Notify the System/370 operator that you are taking the unit off-line.
 - b. Set the Enable/Disable switch to the Disable position. You will see the Disable Indicator light come on, indicating that the unit is off-line.

Note: If the Disable Indicator light does not come on, **do not continue**. Refer to the *IBM Series/1 Operator's Guide*, GA34-0039 for the appropriate action.

- c. Set the Power switch to the **OFF** position.



2. Switch off the remaining units in the rack enclosure(s) and the table-top and stand-alone units.



Switching Power Off in an Emergency



You can switch off the power to all units mounted in a rack enclosure by pulling the Emergency Pull switch located at the top right of the rack enclosure. If you have more than one rack enclosure, you must pull the Emergency Pull switch on each enclosure.

Switching Power On and Off (*continued*)

Restoring Power After an Emergency

1. Place all power switches for the rack-mounted units in the **OFF** position.
2. Correct the problem that required you to pull the switch.
3. Open the 4997 Rack Enclosure rear cover.
4. Find the circuit breaker in the upper left corner of the rack enclosure. Figure 2 shows you the location of the circuit breaker.
5. Push the circuit breaker to the **ON** position to reset the Emergency Pull switch.
6. Close the 4997 Rack Enclosure rear cover.
7. Switch power on as described under "Switching Power On" on page OP-9.

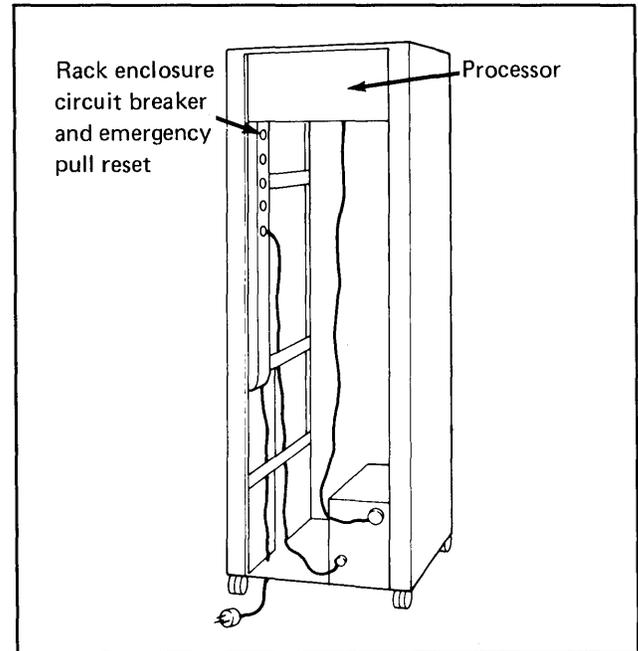


Figure 2. 4997 Rack Enclosure Circuit Breaker Location

Getting Started

Using Diskettes

A diskette consists of two parts: a thin, circular, flexible disk, and a protective jacket. If you could see the flexible disk out of its protective jacket, it would look much like a 45-rpm record. The flexible disk has a magnetic recording surface similar to an audio cassette tape. (See Figure 3 for a picture of a diskette.)

A diskette is used to store information for use by your Series/1. When you insert a diskette into a diskette unit, the Series/1 can read information stored on the diskette. It can also write (store) new information onto the diskette.

Reading information from a diskette or writing information onto one is much like recording on an audio cassette tape. Like the audio cassette tape, information on the diskette does not change until you write new information over the old information.

If you look at a diskette, you can see the recording surface through the head slot. The large center hole is used to spin the diskette past the recording head (called a read/write head). A diskette should be kept in a paper envelope when it is not in the diskette drive. The envelope protects the recording surface exposed at the head slot.

The Event Driven Executive recognizes three types of diskettes: the Diskette 1, the Diskette 2, and the Diskette 2D.

Diskette 1 The *Diskette 1*, also called a one-sided diskette, has a recording surface on one side only and is 8 inches in size.

Diskette 2 The *Diskette 2*, also called a two-sided diskette, has a recording surface on each side and is 8 inches in size.

Diskette 2D The *Diskette 2D* is a two-sided double density diskette; it can hold twice as much information as a Diskette 2.

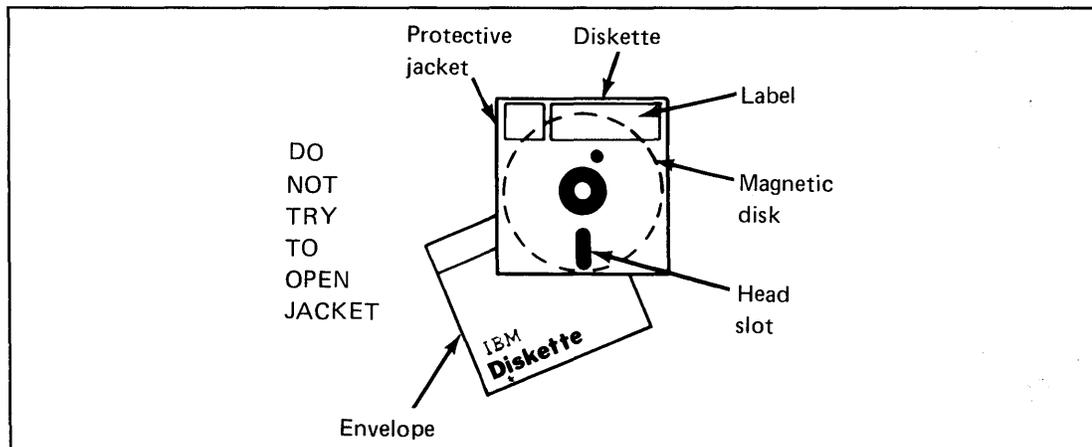


Figure 3. IBM Diskette Parts

Using Diskettes (*continued*)

Handling Diskettes

Because your diskette can be damaged easily, you should be very careful when handling it. If you handle a diskette carelessly, you can lose information stored on it. *A damaged (torn, folded, creased) diskette cannot be used in a diskette drive because it might cause errors.*

Do not touch these surfaces



Take the following precautions to avoid damaging your diskette:

- When you remove a diskette from its envelope, grasp the diskette by its upper edge and pull it out straight.
- **Do not** touch the head slot or any other unprotected surface of the diskette. Body oils and perspiration can damage the recording area.
- Return the diskette to its envelope when you remove the diskette from the diskette drive to keep dust from getting into the diskette.
- If you mail a diskette, place it in a box or heavy cardboard mailer.



- **Do not** try to clean the surface of the diskette.



- **Do not** write on the diskette protective jacket outside the label area.
- **Do not** write on the diskette label with a pencil. Mark the diskette label with a fiber-tip pen. Mark the label **before** attaching it to the diskette.



- **Do not** erase a diskette label or make erasures on the diskette. Eraser crumbs on the diskette surface can make the diskette unusable.

Getting Started

Using Diskettes (*continued*)



- Do not lay the diskette near food, drinks, or ashtrays where foreign materials can contaminate the diskette surface.



- Do not allow the diskette to be near anything that might be magnetized. A magnetic field can cause data loss. Magnetic fields can be found near items such as: fluorescent lights, electronic calculators, dictation equipment, and telephones.



- Do not place heavy objects on the diskette. The weight can seriously damage the surface.



- Do not use clips nor rubber bands on the diskette or bend it in any way.



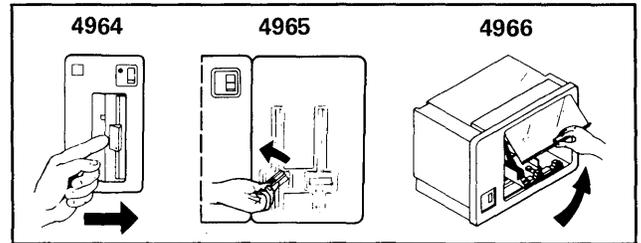
- Do not expose the diskette to heat greater than 51°C (125°F) or to direct sunlight.

Note: If a diskette has been exposed to temperatures outside the range of 10°C to 51°C (50°F to 125°F), keep the diskette at room temperature for about five minutes before inserting it into a diskette drive.

Using Diskettes (*continued*)

Inserting Individual Diskettes

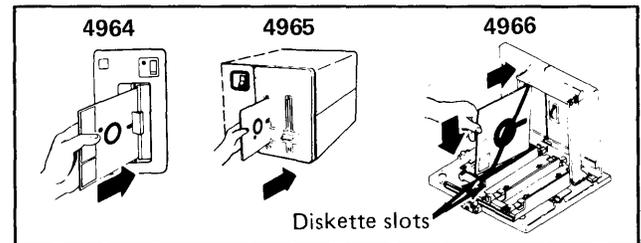
1. Open the diskette unit door.



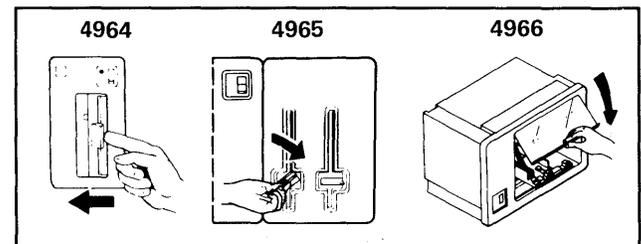
2. Grasp the diskette by the label (upper edge) and remove it from the envelope.



3. Insert the diskette squarely into the diskette unit opening as shown. Make sure that the diskette is completely in the slot, but do not force or bend it. On the 4965 Diskette Storage Unit the label on the diskette faces to the left, and on the 4964 and 4966 diskette units it faces to the right.



4. Close the unit door.



Getting Started

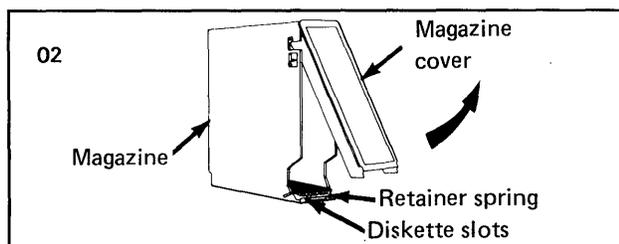
Using Diskettes (*continued*)

Inserting 4966 Magazines

If you are using a 4966 Diskette Magazine Unit use the following procedure. It shows you how to insert diskettes into the magazine, and how to insert the magazine into the 4966.

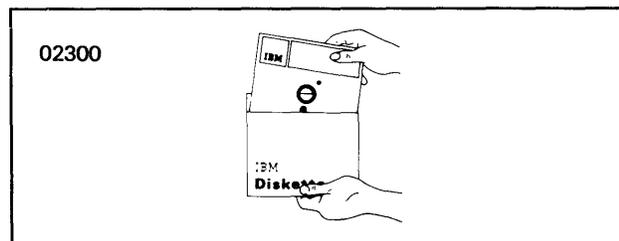
1. Remove the magazine cover.

- a. Hold the retainer spring back.
- b. Lift the cover off.



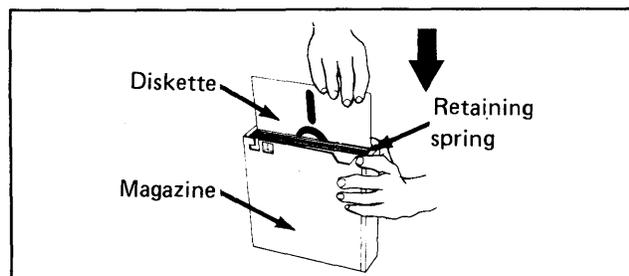
2. If the magazine already contains diskettes, skip to step 5 on page OP-17 .

3. Remove each diskette from its envelope.



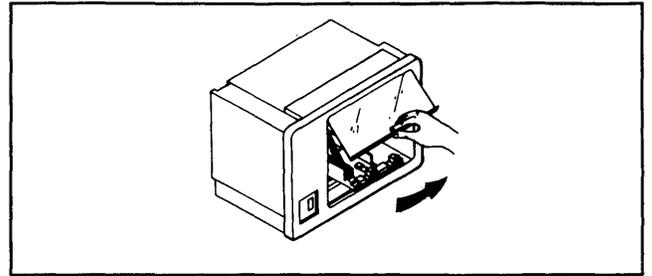
4. Insert each diskette into the magazine:

- a. Grasp the magazine in one hand.
- b. Hold the retainer spring back with your thumb and forefinger.
- c. Place each diskette into an available slot in the magazine. The diskette labels should face to the right when you place the magazine into the unit. The top of the diskette (the end with the external label) is inserted first.

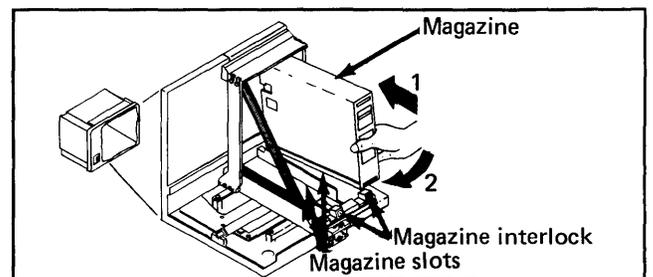


Using Diskettes (*continued*)

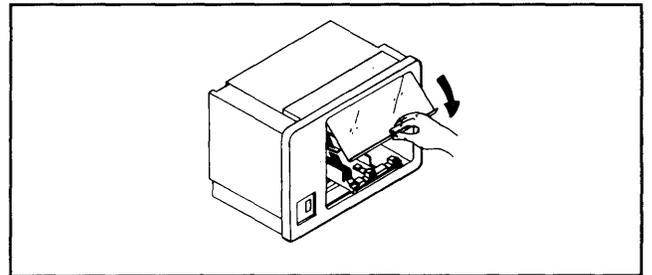
5. Open the door on the magazine unit by lifting the door outward and up. Slide the door back into the top of the unit.



6. Place the magazine into one of the two magazine slots. Check the magazine interlock/indicator to make sure the magazine is locked in place.



7. Close the magazine unit door.



Getting Started

Using Diskettes (*continued*)

Varying the 4966 Diskette Magazine Unit Online

If you are not going to re-IPL your system, you must tell it about the new diskettes in the 4966 Diskette Magazine Unit by entering the \$VARYON operator command as shown in the following procedure. If it was not IPL'd, you do not have to vary the diskette online, EDX will find out about the new diskette when you IPL the Series/1. (See "Entering Commands and Answering Prompts" on page OP-41 for more information about entering operator commands.)

Before You Start:

Observe the following considerations:

- Familiarize yourself with the display terminal and operator commands. (See Chapter 3, "Operating Your Display Terminal" on page OP-29 and "Entering Commands and Answering Prompts" on page OP-41 for instructions on using the display terminal and entering commands.)
- Determine the device address of the 4966 diskette unit and decide whether you want to vary all diskettes in the unit online, or just certain ones. You can vary online the diskette in slot 1, 2, or 3, or those in one of the magazines. (See "Listing the Devices on Your System" on page OP-318 if you do not know the address of your magazine unit.)

1. Press the attention key, type \$VARYON, and press the enter key.

```
> $VARYON  
ENTER DEVICE ADDRESS: _
```

2. Enter the device address.

```
ENTER DEVICE ADDRESS: 22  
ENTER SLOT OR MAGAZINE NO. 1,2,3,A,B(0=ALL):
```

3. Enter the number representing the diskette or magazine that you are varying online. The valid numbers are:

```
ENTER SLOT OR MAGAZINE NO. 1,2,3,A,B(0=ALL): 1  
EDXVOL ONLINE SLOT 1
```

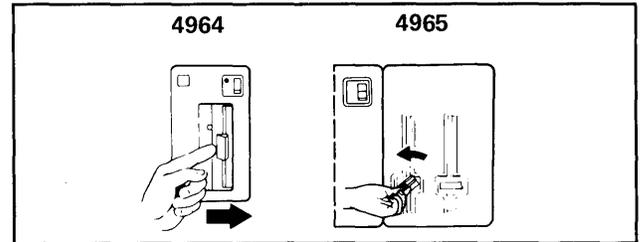
- 0 for the entire 4966 magazine unit
- 1 for Slot 1
- 2 for Slot 2
- 3 for Slot 3
- A for Magazine 1
- B for Magazine 2

Using Diskettes (*continued*)

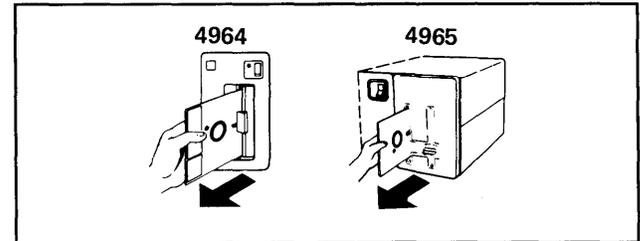
Removing Individual Diskettes

Use the following procedure to remove diskettes from a diskette device with only one diskette. (See "Removing Diskette Magazines" on page OP-20 to remove diskettes or magazines from a 4966 Diskette Magazine Unit.)

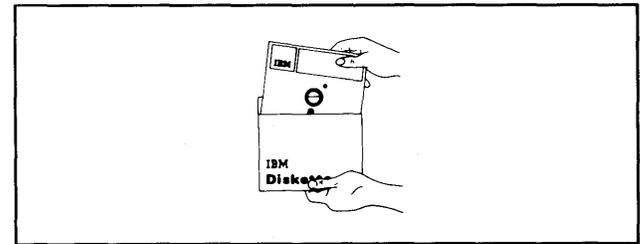
1. Open the diskette door.



2. Grasp the diskette by the label and pull it straight out.



3. Slide the diskette into its envelope.



Getting Started

Using Diskettes (*continued*)

Removing Diskette Magazines

Use the following procedure to remove a diskette magazine:

Before You Start:

- If you plan to IPL the system after you remove the diskettes, skip to step 4 on page OP-21.
- If you are not going to re-IPL the system after you remove the diskettes, observe the following and start with step 1.
 - Know how to use your display terminal and enter operator commands. (See Chapter 3, “Operating Your Display Terminal” on page OP-29 and “Entering Commands and Answering Prompts” on page OP-41 for instructions on using the display terminal and entering commands.)
 - Determine the device address of the 4966 diskette unit and decide whether you want to vary all diskettes in the unit offline, or just certain ones. You can vary offline the diskette in slot 1, 2, or 3, or those in one of the magazines. (See “Listing the Devices on Your System” on page OP-318 if you do not know the address of your magazine unit.)

- 1.** Press the attention key, type \$VARYOFF, and press the enter key.

```
> $VARYOFF  
ENTER DEVICE ADDRESS: _
```

- 2.** Enter the device address.

```
ENTER DEVICE ADDRESS: 22  
ENTER SLOT OR MAGAZINE NO.1,2,3,A,B(0=ALL):
```

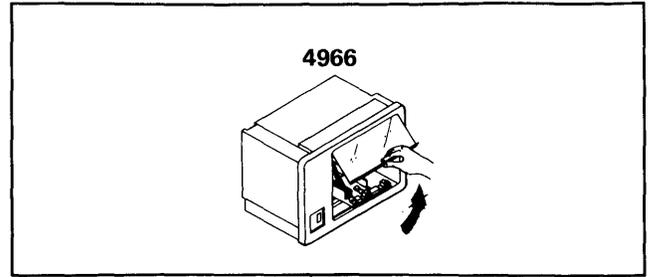
- 3.** Enter the number representing the diskette or magazine that you are varying offline. The valid numbers are:

```
ENTER SLOT OR MAGAZINE NO.1,2,3,A,B (0=ALL): 1  
EDXVOL OFFLINE SLOT 1
```

- 0** for the entire 4966 magazine unit
- 1** for Slot 1
- 2** for Slot 2
- 3** for Slot 3
- A** for Magazine 1
- B** for Magazine 2

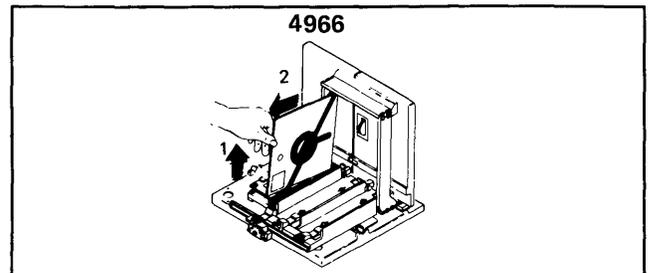
Using Diskettes (*continued*)

4. Open the magazine unit door.

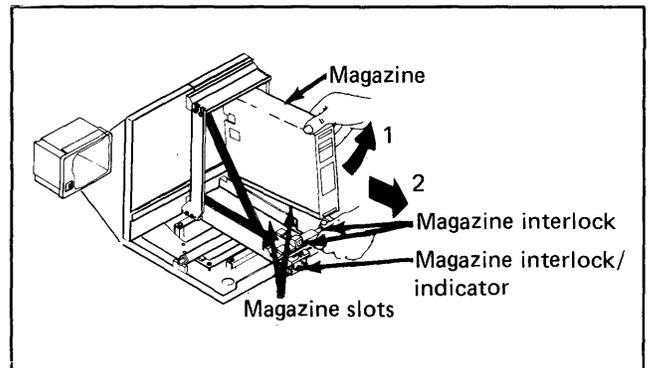


5. Remove the diskette or magazine from the unit:

- To remove a diskette from one of the slots, grasp the diskette by the label, lift it up slightly, and pull it straight out.



- To remove a diskette magazine:
 - a. Grasp the magazine with one hand.
 - b. Using your other hand, press the magazine interlock and unlock the magazine. If the magazine is positioned above the magazine interlock/indicator, push the interlock/indicator down.
 - c. Lift the magazine and remove it from the unit.

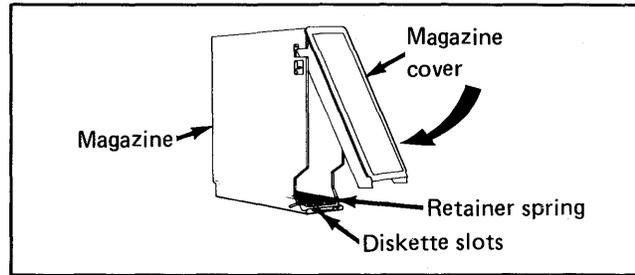


Getting Started

Using Diskettes (*continued*)

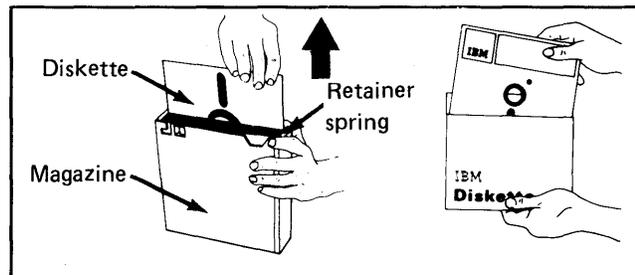
6. Are the diskettes going to be stored in the unit?

- If so, put the magazine cover on the magazine and return the magazine to its storage area.



- If not, remove the diskettes from the magazine:

- a. Hold the retainer spring down and pull each diskette straight out of the magazine.
- b. Slide each diskette into its envelope and return it to a clean storage area.



Loading the System (Initial Program Load)

The procedure for starting (loading) the system is called *initial program load (IPL)*. When you do an IPL, the system loads specific Event Driven Executive programs into storage to execute. These programs prepare (initialize) the devices and control the operation of programs. You must IPL the system before it can do any work.

You can IPL the system in two ways; manually and automatically. In the manual IPL procedure, you start the IPL process after switching on all the devices. In the automatic IPL procedure, the system loads itself automatically after you switch on the processor unit.

You should be familiar with the *basic console* before IPLing the system because it is used in the IPL procedure.

The Basic Console

You should become familiar with the processor consoles before using your Series/1. The consoles are located on the front of the processor. All processors come with a basic console. Your processor may also have the optional programmer console shown in Figure 4 on page OP-23.

Loading the System (Initial Program Load) *(continued)*

The following IPL procedures show you how to use the basic console for IPL. If you need to know more about the basic console or how to use the programmer console, refer to the *IBM Series/1 Operator's Guide, GA34-0039* or the *Event Driven Executive Problem Determination Guide*.

The purpose of the four switches and indicator lights on the basic console are:

Switch	Purpose
On/Off	Switches electrical power to the processor on or off.
IPL Source	Selects the disk or diskette device that contains the operating system you are loading.
Mode	Selects automatic, normal, or diagnostic IPL operation mode. Automatic IPL mode is for automatic IPL as described under "Automatic IPL Procedure" on page OP-25; Normal mode is for manual IPL as described under "Manual IPL Procedure" on page OP-24; and Diagnostic is for problem determination. Diagnostic mode allows the processor to recognize special programming instructions called STOP instructions. The programmer uses these instructions to help solve problems with the programs. (See the <i>Problem Determination Guide</i> for more information about diagnostic mode.)
Load	Resets the system and starts initial program load.
Indicator lights	Purpose
Power On	When on, shows that the electrical power to the processor is switched on.
Load	When on, shows that initial program load is in progress.
Wait	When on, shows that programs are not running.
Run	When on, shows that programs are running.

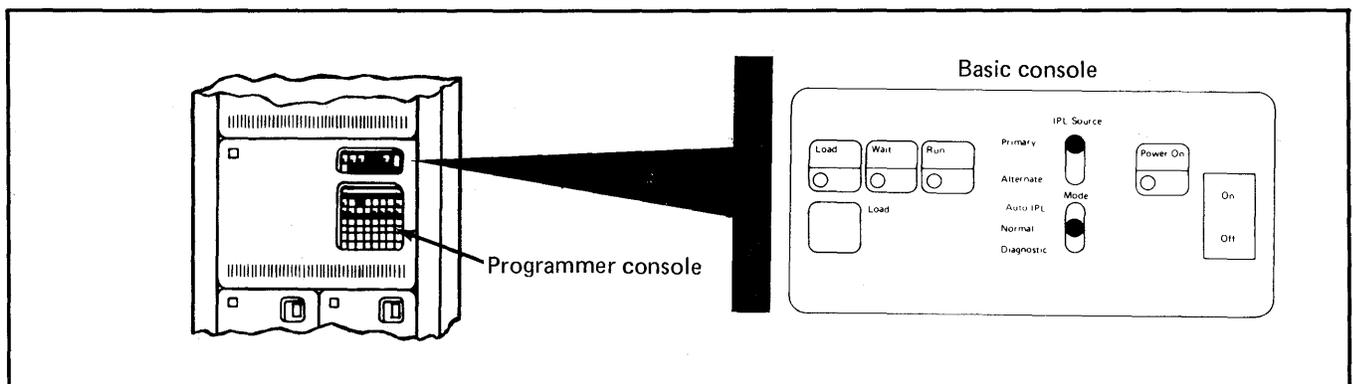


Figure 4. Basic Processor Console

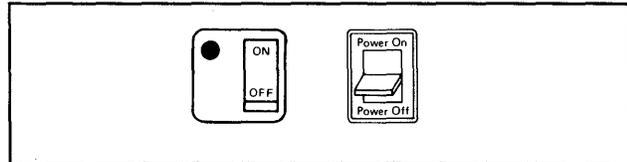
Getting Started

Loading the System (Initial Program Load) *(continued)*

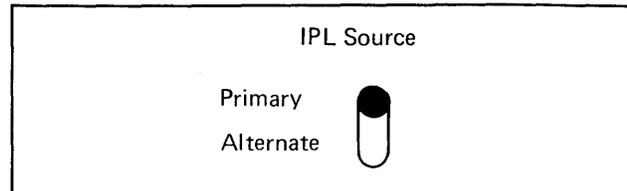
Manual IPL Procedure

Use this procedure to IPL your Series/1 after switching on the electrical power:

1. Place the processor power-on switch to the ON position.



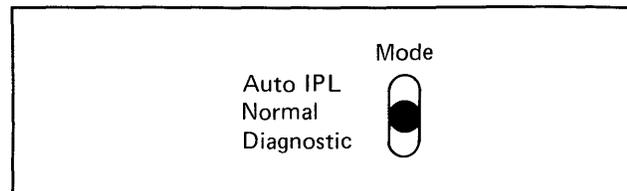
2. Set the IPL source switch to the **Primary** or **Alternate** position, depending on which device you are using. The IPL source can be a disk unit or a diskette unit, and either one can be wired to either switch position.



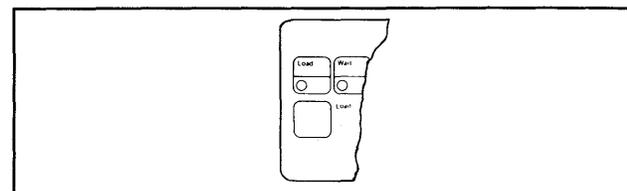
Note: If the IPL device is a diskette unit, insert the correct diskette into the unit.

3. Set the Mode switch to Normal.

If your Series/1 processor also has a programmer console, you may want to set the Mode switch to Diagnostic. The Diagnostic setting allows the processor to recognize special programming instructions called STOP instructions. These instructions are used by a programmer to help solve problems with the programs. (See the *Problem Determination Guide* for further information on diagnostic mode.)



4. Press the Load switch.



Loading the System (Initial Program Load) *(continued)*

5. If you are using the starter system (diskette XS4001 or a copy on the disk), the system displays the illustrated message on the system logging device. The system logging device is the display terminal or printer named \$SYSLOG. If you are using a customized supervisor, the display may be different.

```

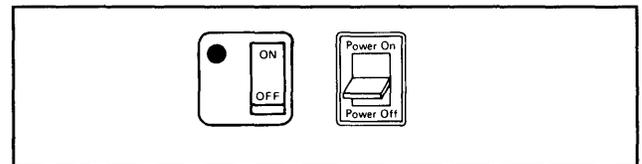
*** EVENT DRIVEN EXECUTIVE *** V5.0
***           XPS           ***
IPL = $EDXNUC ,XS4001
           XPS SYSTEM STORAGE MAP
-----
      USER  USER COMMON  SUPV  SUPV  USER  USER  TOTAL
PART  START  SIZE  SIZE  START  SIZE  START  SIZE  SIZE
#    (DEC) (DEC) (HEX) (HEX) (HEX) (HEX) (HEX) (HEX)
  1   47104 18432    0    0  B800  B800  4800 10000
  2     0    0    0    0    0    0 10000 10000

TOTAL(HEX):           B800    74800
UMAPPED STORAGE =    0 (HEX) 2K BLOCKS
EDX INITIALIZATION COMPLETE
    
```

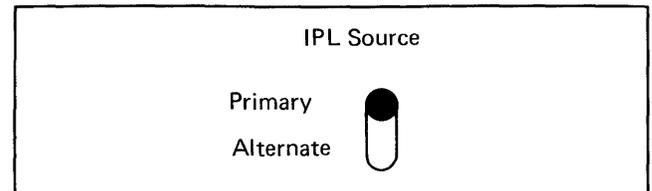
Automatic IPL Procedure

Use the following procedure to IPL your system automatically.

1. Place the processor power-on switch to the ON position.

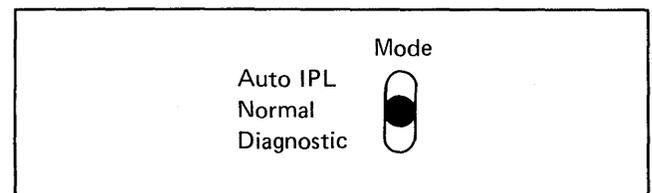


2. Set the IPL source switch to the **Primary** or **Alternate** position, depending on which device you are using. The IPL source can be a disk unit or a diskette unit and either one can be wired to either switch position.



Note: If the IPL device is a diskette unit, insert the correct diskette into the unit.

3. Set the Mode switch to the **Auto IPL** position.

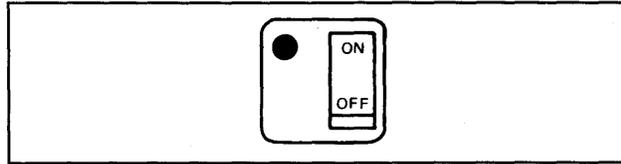


Getting Started

Loading the System (Initial Program Load) *(continued)*

4. Switch on the electrical power to the system. (See "Switching Power On" on page OP-9 for the correct procedure.)

Your system loads itself automatically.



5. If you are using the starter system (diskette XS4001 or a copy on the disk), the system displays the illustrated message on the system logging device. The system logging device is the display terminal or printer named \$SYSLOG. If you are using a customized supervisor, the display may be different.

```
*** EVENT DRIVEN EXECUTIVE *** V5.0
*** XPS ***
IPL = $EDXNUC ,XS4001
XPS SYSTEM STORAGE MAP
-----
PART  USER  USER COMMON  SUPV  SUPV  USER  USER TOTAL
#    START  SIZE  SIZE  START  SIZE  START  SIZE  SIZE
1    47104 18432   0     0    B800  B800  4800 10000
2     0     0     0     0     0     0    0 10000 10000
.
TOTAL(HEX):          B800      74800
UMAPPED STORAGE = 0 (HEX) 2K BLOCKS
EDX INITIALIZATION COMPLETE
```

Setting the Time and Date

Use this procedure to set the time and date on your system either after IPL or during system operation.

If you are not familiar with using your display terminal or entering operator commands, see Chapter 3, "Operating Your Display Terminal" on page OP-29 and "Entering Commands and Answering Prompts" on page OP-41 before using this procedure.

Notes:

1. \$T can only be entered from the system logging device. The system logging device is a terminal with the name \$SYSLOG. If \$SYSLOG is not working, the alternate system logging device, the terminal named \$SYSLOGA, can be used. (See "Listing the Devices on Your System" on page OP-318 for the procedure for determining the names of the terminals attached to your system.) If you enter \$T from a terminal with another name, EDX only displays the time as if you had entered a \$W command.
2. The \$T command can only be entered in prompt-reply format.

Loading the System (Initial Program Load) *(continued)*

- 1 Press the attention key, type \$T, and press the enter key.

```
> $T  
DATE (M.D.Y): _
```

- 2 Enter the month, day, and year separated by either a period (.), a colon (:), or a blank.

```
DATE (M.D.Y): 05.21.82  
TIME (H.M): _
```

- 3 Enter the hours and minutes separated by either a period (.) , a colon (:), or a blank. The system starts the seconds at zero (0).

```
TIME (H.M): 08:30  
DATE = 05/21/82 TIME = 08:30:00
```

Displaying the Date and Time

Use this procedure to display the date and time. This procedure can be entered from any display terminal.

If you are not familiar with using you display terminal or entering operator commands, see Chapter 3, "Operating Your Display Terminal" on page OP-29 and "Entering Commands and Answering Prompts" on page OP-41 before using this procedure.

Press the attention key, type \$W, and press the enter key.

```
> $W  
DATE = 05/21/82 TIME = 08:30:00
```


Chapter 3. Operating Your Display Terminal

On the Series/1, display terminals, printers, and other devices that can send and receive information are all considered *terminals*. This chapter explains the basics of operating and controlling display terminals. However, it does not contain all there is to know about them. For more information on how to operate your terminal, refer to the hardware operation manual.

A display terminal is much like a television with a typewriter keyboard attached. Instead of a picture, the display tube shows what is being typed on the keyboard. It also displays messages sent to you by the Series/1.

You use a display terminal to give instructions (commands, data) to the EDX operating system, and communicate with your application programs. The operating system and application programs use a display terminal to send messages that either tell you what the system is doing or request information from you. These messages can be a single line request from EDX, called a *prompt*, or a full, formatted display called a *screen image*. A screen image contains information for you, and/or prompts.

Your Series/1 can have one or more display terminals attached to it. You can have a 4978 display station, 4979 display station, 4980 display station or 3101 display terminal. Throughout this chapter these are all called *display terminals*. (Figure 5 on page OP-31 shows four display terminals supported by EDX.)

Operating Your Display Terminal

This chapter contains the following procedures:

- “Switching Power On”
- “Adjusting Screen Brightness”
- “Moving the Cursor” on page OP-32
- “Entering and Changing Characters” on page OP-34
- “Erasing Data from the Screen” on page OP-37
- “Printing the Terminal Display Screen” on page OP-40
- “Entering Commands and Answering Prompts” on page OP-41
- “Clearing the Display Screen” on page OP-43
- “Listing System Terminal Assignments” on page OP-44
- “Sending a Message to Another Terminal” on page OP-45
- “Renaming a Terminal” on page OP-47
- “Reassigning a Terminal Address” on page OP-48
- “Changing the Hard-Copy Device” on page OP-49
- “Changing a Terminal’s Partition Assignment” on page OP-50
- “Changing the Display Screen Format” on page OP-51
- “Varying a Terminal Offline” on page OP-52
- “Varying a Terminal Online” on page OP-54
- “Changing Display Terminal Keyboard Values” on page OP-55

Switching Power On

Place the power switch in the **ON** position to switch on the electrical power to your display terminal. Figure 5 on page OP-31 shows the location of the power switches.

Adjusting Screen Brightness

Adjust the brightness and contrast of the display screen by turning the brightness and contrast controls. Figure 5 on page OP-31 shows the location of these controls.

Adjusting Screen Brightness (continued)

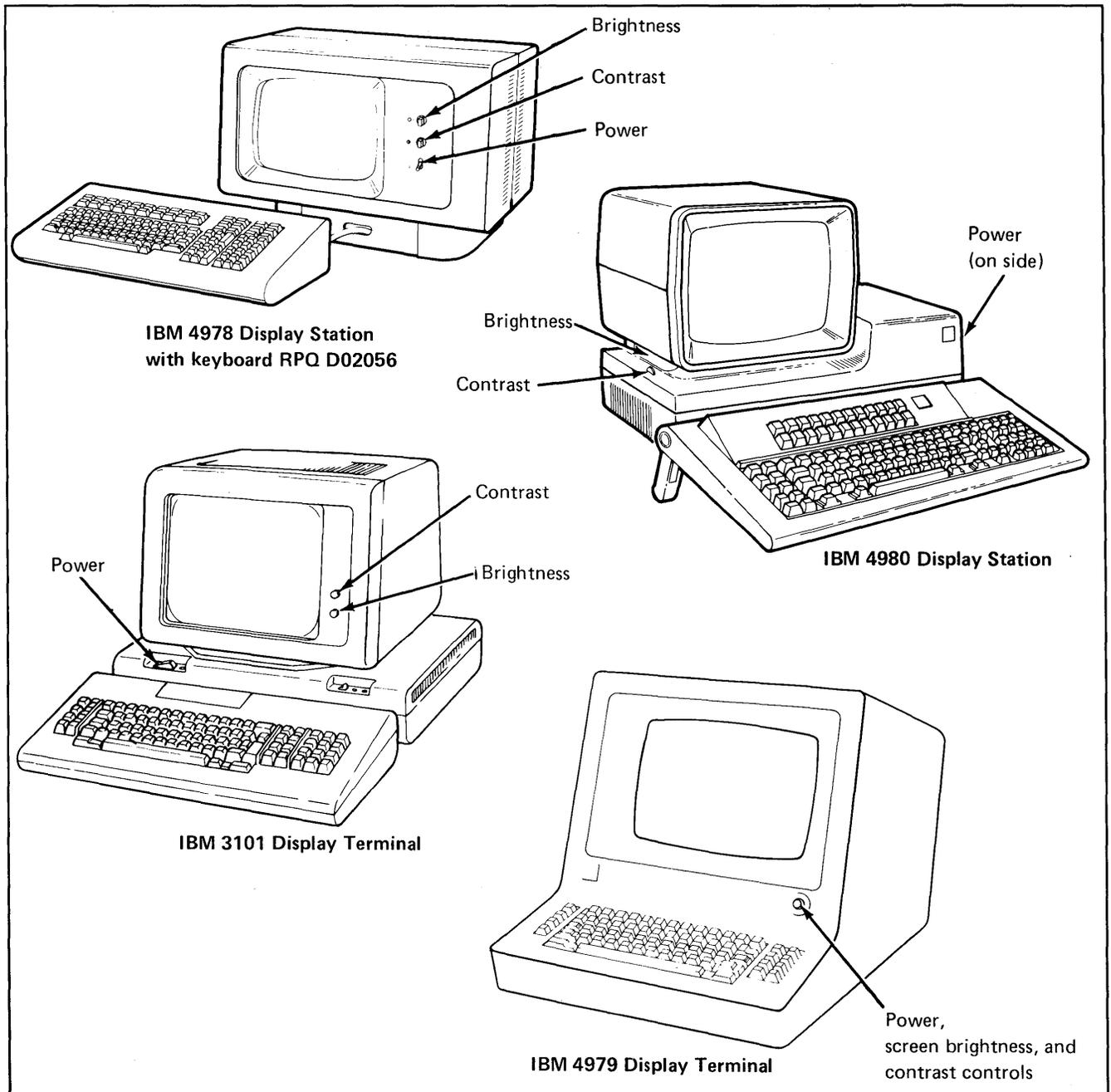


Figure 5. Series/1 Display Terminals

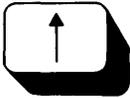
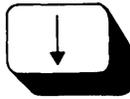
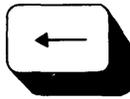
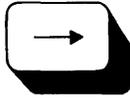
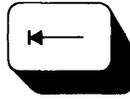
Operating Your Display Terminal

Moving the Cursor

The cursor is a special symbol displayed on the screen. It indicates where you should enter or change a character or number. On the 4978 and 4979 display stations, the cursor is a short underscore that appears as a dash of light on the screen. The cursor is block-shaped on the 3101 display terminal. On the 4980 display station, the cursor can be either underscored or block-shaped.

The keyboards on the display terminals have special keys that control the cursor. These keys move the cursor to any position on the display screen without affecting the information displayed. Most of these keys are *typomatic* keys - the cursor continues moving until you release the key. The following chart lists the keys and their direction of movement:

Moving the Cursor (*continued*)

Cursor Direction	Key used
Move toward the top of the screen.	Cursor up 
Move toward the bottom of the screen.	Cursor down 
Move toward the left side of the screen.	Cursor left 
Move toward the right side of the screen.	Cursor right 
Move to the home position. (This is the first unprotected character or the upper leftmost character position on the screen). The 4979 does not have a home key.	Cursor home 
Move to the home position. On the 4980, press the alternate key (Alt) and the blank key (located between cursor left and cursor right) to move to the cursor to home.	Cursor home 
Move to position one of the next line.	New line 
Tab to the right one field	Forward tab 
Tab to the left one field	Backward tab 

Operating Your Display Terminal

Entering and Changing Characters

To enter or change one or more characters on the display screen, move the cursor to the correct position. Then, type new characters or delete existing characters.

Some display screen images, called static screens, are formatted with protected and unprotected areas, or fields. The protected fields contain system messages or operator prompts indicating the type of information that should be entered in the corresponding unprotected field. You can enter or change characters in unprotected fields only. Unprotected fields are recognized by the fact that characters in those fields are brighter than characters in protected fields.

Entering New Characters

1. Place the cursor under a blank, unprotected, screen position.

```
SOURCE INPUT (NAME,VOLUME) ==>
OBJECT OUTPUT (NAME,VOLUME) ==>
OPTIONAL PARAMETERS ==>
```

2. Type the desired character(s) on the keyboard and it will be entered in that position on the screen.

```
SOURCE INPUT (NAME,VOLUME) ==> MYSRCE,MYVOL_
OBJECT OUTPUT (NAME,VOLUME) ==>
OPTIONAL PARAMETERS ==>
```

Erasing with the Space Bar

1. Place the cursor under the first unprotected character to be erased.

```
SOURCE INPUT (NAME,VOLUME) ==> MYSRCE,MYVOL
OBJECT OUTPUT (NAME,VOLUME) ==>
OPTIONAL PARAMETERS ==>
```

2. Press the space bar to erase the character(s). (The space bar is a typomatic key and can be held down to erase multiple characters.)

```
SOURCE INPUT (NAME,VOLUME) ==> _E,MYVOL
OBJECT OUTPUT (NAME,VOLUME) ==>
OPTIONAL PARAMETERS ==>
```

Entering and Changing Characters (*continued*)

Replacing and Deleting Characters

1. Place the cursor under an unprotected character.

Continue with Step 2(a) to replace characters, or Step 2(b) to delete characters.

```
SOURCE INPUT (NAME,VOLUME) ==> MYSRCE,MYVOL
OBJECT OUTPUT (NAME,VOLUME) ==> MYOUTDB,MYVOL
OPTIONAL PARAMETERS ==>
```

- 2(a). Replace the character by typing a new one. In the example the characters DB have been replaced by the characters DS.

```
SOURCE INPUT (NAME,VOLUME) ==> MYSRCE,MYVOL
OBJECT OUTPUT (NAME,VOLUME) ==> MYOUTDS,MYVOL
OPTIONAL PARAMETERS ==>
```

- 2(b). Delete characters by pressing the delete key once for each character. When you delete characters, the other characters in the unprotected area are shifted to the left, creating one or more *null* characters in the unprotected area. (Null characters are special characters that define the unprotected fields.)

```
SOURCE INPUT (NAME,VOLUME) ==> MYSRCE,MYVOL
OBJECT OUTPUT (NAME,VOLUME) ==> MYOUT,MYVOL
OPTIONAL PARAMETERS ==>
```

Operating Your Display Terminal

Entering and Changing Characters (*continued*)

Inserting Characters

1. Place the cursor under an unprotected character.

```
SOURCE INPUT (NAME,VOLUME) ==> MYSRCE,MYVOL  
OBJECT OUTPUT (NAME,VOLUME) ==> MYOUT,MYVOL  
  
OPTIONAL PARAMETERS ==>
```

2. Press the **INS MOD** (insert mode) key.

On the 4978, insert mode is indicated by a small light on the right side of the screen. On the 4979, insert mode is indicated by a blinking cursor. On the 3101, insert mode is indicated by the word **INSERT** at the bottom of the screen. On the 4980, insert mode is indicated on the status line by the symbol (/ \) .

INS
MODE

3. Type the desired character(s). The terminal inserts the new character(s) before the character above the cursor. The other characters in the unprotected area shift to the right. You can insert characters until you use up all the null characters in the unprotected area. The cursor goes to the next unprotected field or back to the start of the current field when the end of the current field is reached.

```
SOURCE INPUT (NAME,VOLUME) ==> MYSRCE,MYVOL  
OBJECT OUTPUT (NAME,VOLUME) ==> MYOUTDS,MYVOL  
  
OPTIONAL PARAMETERS ==>
```

4. Press the **RESET** key to end insert mode on all terminals except the 3101. On the 3101, press the insert key to reset.

RESET

Erasing Data from the Screen

The next three procedures show you how to do the following operations:

- Erase all unprotected characters within one field.
- Erase all unprotected characters on one line.
- Erase all unprotected characters on a display screen.

Erasing to the End of the Field

1. Place the cursor under the first character to be erased. For illustration purposes, the example screen has a two line unprotected field for the OBJECT OUTPUT line.

```
SOURCE INPUT (NAME,VOLUME) ==> MYSRCE,MYVOL
OBJECT OUTPUT (NAME,VOLUME) ==> _MYOUTDS,MYVOL
                                YOURDS,YRVOL
OPTIONAL PARAMETERS ==> LIST $$SYSPRTR CO
```

2. Press the **Erase EOF** (end of field) key to erase all characters from the cursor to the end of that unprotected field. The erase end of field function is slightly different on each terminal. Instructions for each terminal are listed below:

- On the 4978, 4979, and 3101, press the EOF key.
- On the 4980, press the erase EOF key.

EOF

```
SOURCE INPUT (NAME,VOLUME) ==> MYSRCE,MYVOL
OBJECT OUTPUT (NAME,VOLUME) ==> _
OPTIONAL PARAMETERS ==> LIST $$SYSPRTR CO
```

Operating Your Display Terminal

Erasing Data from the Screen (*continued*)

Erasing to the End of the Line

1. Place the cursor under the first character to be erased. For illustration purposes, the example screen has a two line unprotected field for the OBJECT OUTPUT line.

```
SOURCE INPUT (NAME,VOLUME) ==> MYSRCE,MYVOL
OBJECT OUTPUT (NAME,VOLUME) ==> _MYOUTDS,MYVOL
                                     YOURDS,YRVOL
OPTIONAL PARAMETERS ==> LIST $$SYSPRTR CO
```

2. On all terminals, press the EOL (end of line) key to erase all unprotected characters from the cursor to the end of the line.



```
SOURCE INPUT (NAME,VOLUME) ==> MYSRCE,MYVOL
OBJECT OUTPUT (NAME,VOLUME) ==>
OPTIONAL PARAMETERS ==> LIST $$SYSPRTR CO
```

Erasing Data from the Screen *(continued)*

Erasing the Screen

Press the **Erase EOS** (end of screen) key to erase all unprotected fields on the screen. The erase end of screen function is slightly different on each terminal. Instructions for each terminal are listed below:

- On the 4978, hold the shift key down and press the EOS key.
- On the 4979, hold the shift key down and press the EOS key.
- On the 4980, press the EOS key.
- On the 3101, press the EOS key.

To erase all protected and unprotected fields, see "Clearing the Display Screen" on page OP-43.

```
SOURCE INPUT (NAME,VOLUME) ==> MYSRCE,MYVOL
OBJECT OUTPUT (NAME,VOLUME) ==> _MYOUTDS,MYVOL

OPTIONAL PARAMETERS ==> LIST $SYSRTR CO
```

EOS

```
SOURCE INPUT (NAME,VOLUME) ==>
OBJECT OUTPUT (NAME,VOLUME) ==> _
OPTIONAL PARAMETERS ==>
```

Operating Your Display Terminal

Printing the Terminal Display Screen



Press the **PF6** key on a 4978, 4979, or 4980 display terminal keyboard to print the contents of the screen. The system prints the screen on the printer (hard-copy device) designated for this purpose when your system was set up.

Notes:

1. The EDX system does not support printing the display screen with a program function (PF) key on the 3101 display terminal. However, you can write your own program to do this, using the `ATTNLIST` statement as described in the *Language Reference*.
2. If printer spooling is using the same printer, EDX does not print the display screen. If this occurs, see “Changing the Hard-Copy Device” on page OP-49 for instructions on how to change the assigned printer.
3. If the hard-copy device has not been defined, or is busy, the screen is not printed.
4. The **PF6** key should not be pressed while the screen is being changed or when I/O is directed to the screen.
5. While the screen is being printed, other keys (such as other PF keys, the attention key, or the enter key) should not be pressed.

Entering Commands and Answering Prompts

You can communicate with EDX by entering system operator commands, utility commands, and answering utility prompts. You can also communicate with some application programs by entering data in response to a prompt. The following two procedures show you how to enter operator commands and respond to prompts from utilities and application programs.

Entering Operator Commands

1. Press the attention key.

The attention key function varies slightly from terminal to terminal. The following list provides instructions for each Series/1 terminal.

- On the 4978, 4979, or 4980, press the ATTN key.
- On the 3101, press the ALT and PF8 keys at the same time.



or



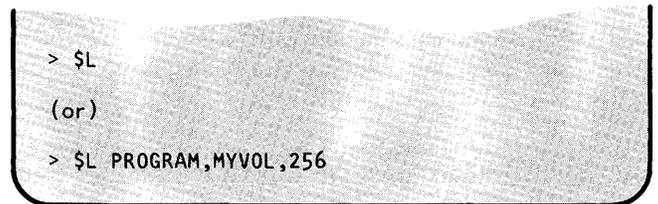
After you press the attention key, EDX responds by displaying a greater-than sign (>) on the display screen and placing the cursor under the first unprotected field following the > sign.



Note: The generic term *attention* is used throughout this book for the ATTN and PF8/ALT keys.

2. Type the operator command. Operator commands can be entered in either prompt-reply or single-line format. With prompt-reply format, you enter the command name, and continue with step 3 on page OP-42.

With the single-line format, you enter the command name followed by all parameters before going on to step 3 on page OP-42 .



Operating Your Display Terminal

Entering Commands and Answering Prompts (*continued*)

3. Press the enter key to tell EDX to read the command you typed on the display screen. On 4978, 4979, and 4980 display stations, use the key labeled ENTER. On a 3101 display terminal running in block mode², use the key labeled SEND. On a 3101 display terminal running in character mode², use the *new line* key. (If a terminal is defined in the system as a full- or static-screen device, EDX also expects you to press the enter key whenever the screen is filled with output.)



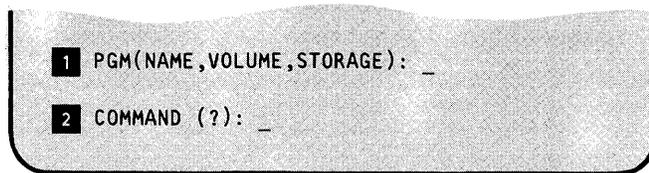
If you are entering a command using the prompt-reply format, continue with the procedure "Responding to Prompts."

Note: The generic term *enter* is used throughout this book for the ENTER, SEND, and new line keys.

Responding to Prompts

1. An operator command entered in the prompt-reply format, a system utility, and some application programs will prompt you for commands or data as shown in the example. When EDX issues a prompt, the cursor is placed under the first unprotected character following the prompt.

The example screen shows two possible prompts. The first **1** is from the \$L operator command entered in the prompt-reply format and the second **2** is from a utility.



² When running in *block mode*, a 3101 display terminal sends a block of characters to the Series/1 when you press the SEND key. When running in *Character mode* a 3101 display terminal sends each character to the Series/1 when you press the character key. The 3101 displays the current mode at the bottom of the display screen.

Entering Commands and Answering Prompts (*continued*)

2. Type the requested command or data. (If you enter a '?' in response to a utility COMMAND prompt, EDX lists the commands for that utility.)

```
PGM(NAME,VOLUME,STORAGE): PROGRAM,MYVOL,256  
COMMAND (?): LA
```

3. Press the enter key to tell EDX to read the information you typed on the display screen. EDX reads the information entered and either performs the operation or issues another prompt. (The enter key is described under "Entering Operator Commands" on page OP-41.)



Clearing the Display Screen

Use the following procedure to erase the entire display terminal screen (both protected and unprotected areas). If you want to erase unprotected areas only, see "Erasing the Screen" on page OP-39.

Press the attention key and enter **\$B**.

```
> $B
```

Operating Your Display Terminal

Listing System Terminal Assignments

Use the following procedure to find out what name, address, type, and partition is assigned to each terminal in your Series/1 configuration. By using this procedure to display your terminal configuration, you can also tell which terminals are online and available for use. You can also find out which terminals are connected to a printer (hardcopy).

1. Load the utility \$TERMUT1:

- Using the \$L operator command, press the attention key and enter \$L \$TERMUT1.
- Using the session manager, select option 4.1.

```
> $L $TERMUT1
LOADING $TERMUT1 7P,01:01:18, LP= 9100, PART=1

*** TERMINAL CONFIGURATOR ***

COMMAND(?): _
```

2. Enter LA to list terminal assignments.

```
COMMAND(?): LA

      NAME  ADDR TYPE      PART  HARDCOPY  ON-LINE
      CDRVTA **  VIRT      1      YES  CONNCTD  CDRVTB
      CDRVTB **  VIRT      1      YES  CONNCTD  CDRVTA
===> $SYSLOG 04  4979      1  $SYSRTR  YES
      MPRTR  01  4974 PTR    1      YES
      TERM2  24  4978      2  MPRTR    YES
      $SYSRTR 21  4973 PTR    1      YES

COMMAND(?): _
```

The headings on the listing have the following meanings:

- NAME** The symbolic name of the terminal.
- ADDR** The hexadecimal address where the terminal is attached to the Series/1.
- TYPE** The numeric IBM machine name of the terminal.
- PART** The number of the storage partition to which the terminal is assigned.
- HARDCOPY** The name of the printer associated with the display terminal.
- ON-LINE** Shows whether the terminal is available for use by the system.
- ===>** Points to the information for the terminal that you are currently using.

Listing System Terminal Assignments (*continued*)

3. Do you want to use \$TERMUT1 for another procedure?

- If so, continue with that procedure.
- If not, enter **EN** to end \$TERMUT1:

```
COMMAND(?): EN
$TERMUT1 ENDED AT 01:03:53
```

Sending a Message to Another Terminal

Use the following procedure to send a message to another display terminal, or a printer, that is attached to your Series/1.

Before You Start:

Determine the name of the terminal to which you are sending a message. If you don't know the name, see "Listing System Terminal Assignments" on page OP-44 for instructions on how to get the name.

1. Was \$TERMUT3 suspended after sending a previous message?

- If so, continue with Step 2(a).
- If not, continue with Step 2(b).

2(a). Press the attention key, enter **SEND**, and continue with Step 3.

```
> SEND
ENTER TERMINAL NAME
```

2(b). Load the utility \$TERMUT3:

- Using the \$L operator command, press the attention key and enter **\$L \$TERMUT3**.
- Using the session manager, select option 4.3.

```
> $L $TERMUT3
LOADING $TERMUT3 4P,00:04:54, LP= 0000,...
TERMINAL BROADCAST UTILITY
ENTER TERMINAL NAME: _
```

Operating Your Display Terminal

Sending a Message to Another Terminal (*continued*)

3. Enter the name of the terminal to which you are sending the message. If you do not enter a terminal name, the message is sent back to the display terminal you are using.

```
ENTER TERMINAL NAME: TERM1
ENTER MESSAGE TO SEND:
_
```

4. Enter the message. The message cannot be longer than one line, but you can send a second line as shown in Step 5.

```
ENTER MESSAGE TO SEND
THIS IS MESSAGE ONE
ANOTHER MESSAGE (Y/N)? _
```

5. Do you want to send another message (or continuation of the same message) to the same terminal?

- If so, enter a Y and continue with Step 4.
- If not, enter an N and continue with Step 6.

```
ANOTHER MESSAGE (Y/N)? N
ANOTHER TERMINAL (Y/N)? _
```

6. Do you want to send a message to another terminal?

- If so, enter a Y and continue with Step 3.
- If not, enter an N and continue with Step 7.

```
ANOTHER TERMINAL (Y/N)? N
DETACH PGM,( WAIT FOR 'ATTN SEND')(Y/N)? _
```

7. Do you want to suspend \$TERMUT3 so you can enter another message later? (If you suspend \$TERMUT3, you can do other operations from your terminal; you will not have to reload \$TERMUT3 when you want to send another message.)

- If so, enter a Y and continue with other operations.
- If not, enter an N to end \$TERMUT3.

```
DETACH PGM,( WAIT FOR 'ATTN SEND')(Y/N)? N
$TERMUT3 ENDED AT 00:12:47
```

Renaming a Terminal

Use the following procedure to temporarily change the symbolic name of a terminal. This change remains in effect until you IPL the system. To make a permanent change, you must regenerate your EDX system as described in the *Installation and System Generation Guide*.

Before You Start:

Determine the name of the terminal you are currently using and the new name you are assigning to it. If you do not know what the current name is, see “Listing System Terminal Assignments” on page OP-44.

Note: Two terminals cannot have the same name.

1. Load the utility \$TERMUT1:

- Using the \$L operator command, press the attention key and enter \$L TERMUT1.
- Using the session manager, select option 4.1.

```
> $L $TERMUT1
LOADING $TERMUT1 20P,01:01:18, LP= 9100,...
*** TERMINAL CONFIGURATOR ***
COMMAND(?): _
```

2. Enter RE.

```
COMMAND(?): RE
ENTER OLD TERMINAL NAME: _
```

3. Enter the current terminal name.

```
ENTER OLD NEW TERMINAL NAMES: CURNAME
ENTER NEW TERMINAL NAME: _
```

4. Enter the new terminal name.

```
ENTER NEW TERMINAL NAME: NEWNAME
COMMAND (?): _
```

5. Do you want to rename another terminal?

- If so, continue with Step 3.
- If not, enter EN to end \$TERMUT1.

```
COMMAND(?): EN
$TERMUT1 ENDED AT 01:01:08
```

Operating Your Display Terminal

Reassigning a Terminal Address

The address specified on the `TERMINAL` statement for a particular terminal is the the address to which EDX thinks the terminal is attached. If this address is incorrect, you can use this procedure to temporarily correct it. This change is effective only until you IPL the Series/1. If you want to make it permanent, you must regenerate your system as described in the *Installation and System Generation Guide*.

Note: You cannot change the address of the display terminal you are currently using.

Before You Start:

Determine the following:

- The name of the terminal you are currently using.
- The new address you are assigning the terminal.

If you do not know the current name of the terminal, see “Listing System Terminal Assignments” on page OP-44.

1. Load the utility \$TERMUT1:

- Using the `$L` operator command, press the attention key and enter `$L TERMUT1`.
- Using the session manager, select option 4.1.

```
> $L $TERMUT1
LOADING $TERMUT1 20P,01:01:18, LP= 9100,...
*** TERMINAL CONFIGURATOR ***
COMMAND(?): _
```

2. Enter RA.

```
COMMAND(?): RA
ENTER TERMINAL NAME: _
```

3. Enter the name of the terminal.

```
ENTER TERMINAL NAME: TERM2
ENTER NEW ADDRESS: _
```

4. Enter new address.

```
ENTER NEW ADDRESS: 24
COMMAND (?): _
```

Reassigning a Terminal Address (*continued*)

5. Do you want to change another terminal address?

- If so, continue with step 2 on page OP-48.
- If not, enter **EN** to end \$TERMUT1.

```
COMMAND(?): EN
$TERMUT1 ENDED AT 01:01:08
```

Changing the Hard-Copy Device

Use the following procedure to temporarily change the printer specified as a hard-copy device for the 4978, 4979, or 4980 display terminal that you are currently operating. This is the printer that EDX uses to print the contents of the display screen when you press the PF6 key. You can also use this procedure to change the program function (PF) key that prints the screen.

These changes are in effect until you IPL your system. To make a permanent change, you must regenerate your EDX system as described in the *Installation and System Generation Guide*.

Before You Start:

Obtain the following:

- The name of the printer you want to use.
- The number of the PF key you want to use.

(See “Listing System Terminal Assignments” on page OP-44 to get the name of the printer as it is known to EDX.)

1. Load the utility \$TERMUT1:

- Using the \$L operator command, press the attention key and enter **\$L \$TERMUT1**.
- Using the session manager, select option 4.1.

```
> $L $TERMUT1
LOADING $TERMUT1 20P,01:01:18, LP= 9100,...
*** TERMINAL CONFIGURATOR ***
COMMAND(?): _
```

2. Enter RH.

```
COMMAND(?): RH
HARDCOPY TERMINAL NAME: _
```

Operating Your Display Terminal

Changing the Hard-Copy Device (*continued*)

3. Enter the new printer name.

```
HARDCOPY TERMINAL NAME: MPRTR
KEY CODE (DECIMAL): _
```

4. Enter the decimal number of the PF key. If you do not enter a PF key number, EDX uses the old key.

```
KEY CODE (DECIMAL): 4
HARDCOPY KEY IS      4
COMMAND (?): _
```

5. Do you want to use \$TERMUT1 for another job?

- If so, continue with that procedure.
- If not, enter EN to end \$TERMUT1.

```
COMMAND(?): EN
$TERMUT1 ENDED AT 01:01:08
```

Changing a Terminal's Partition Assignment

Use this procedure to temporarily change the partition in which your terminal runs. This change stays in effect until you IPL your system. To make it permanent, you must regenerate your EDX system as described in the *Installation and System Generation Guide*.

Before You Start:

Obtain the new partition number. (See "Listing Partitions and Active Programs" on page OP-227 to find out what is running in each storage partition.)

1. Press the attention key and enter \$CP.

```
> $CP
PARTITION # ? _
```

Changing a Terminal's Partition Assignment (*continued*)

2. Enter the new partition number.

Note: If your system has a 4952 Processor, you are limited to partitions 1 and 2.

```
PARTITION # ? 2
PROGRAMS AT 09:30:50
IN PARTITION #2
MYPROG 0000 *
PARTITION ADDRESS: 0000 HEX; SIZE: 65536 DECIMAL BYTES
```

Changing the Display Screen Format

Use the procedure "Changing Page Formatting Parameters" on page OP-253 to change the way EDX displays (formats) the display screen. The parameters that you can change are the same as those for a page of printed output. If you want to design a special display screen, including text, see the \$IMAGE utility in the *Operator Commands and Utilities Reference*.

Operating Your Display Terminal

Varying a Terminal Offline

Use the following procedure to vary a terminal offline. When a terminal is offline, the system receives no input from that terminal and that terminal sends no data to the system. In other words, no communications between system and terminal take place.

You can vary the terminal you are currently using offline. However, once \$TERMUT1 varies it offline, it is inactive and you will have to use another terminal to vary it back online.

Before You Start:

Determine the symbolic name or the hexadecimal address of the terminal you plan to use. See "Listing System Terminal Assignments" on page OP-44 if you don't know the name or address.

1. Load the utility \$TERMUT1:

- Using the \$L operator command, press the attention key and enter **\$L TERMUT1**.
- Using the session manager, select option 4.1.

```
> $L $TERMUT1
LOADING $TERMUT1 20P,01:01:18, LP= 9100,...
*** TERMINAL CONFIGURATOR ***
COMMAND(?): _
```

2. Enter **OF** to vary a terminal offline.

```
COMMAND(?): OF
ENTER TERMINAL NAME, OR ADDRESS IN ( ): _
```

3. Enter the symbolic name, or the hexadecimal address, of the terminal. If you enter the hexadecimal address, place it in parenthesis ().

If the terminal is in use, \$TERMUT1 asks you if you still want to vary it offline. If you do not receive this message, continue with step 5 on page OP-53.

```
ENTER TERMINAL NAME, OR ADDRESS IN ( ): MPRTR
WARNING - TERMINAL IN USE
CONTINUE VARYOFF PROCESSING? _
```

Varying a Terminal Offline (*continued*)

4. If you receive the `TERMINAL IN USE` message, do you want to continue and vary it offline? (If you decide to have `$TERMUT1` continue processing your request, normal processing of any program using that terminal is not guaranteed. Also, if you are varying offline the terminal you are using, `$TERMUT1` ends after processing your request. You will have to use another terminal to vary yours back online.)

- If so, enter a **Y**.
- If not, enter an **N**.

```
WARNING - TERMINAL IN USE
CONTINUE VARYOFF PROCESSING? Y

TERMINAL SUCCESSFULLY VARIED OFF
COMMAND(?): _
```

5. Do you want to use `$TERMUT1` to vary another terminal offline, or do you want to end it?

- To vary another terminal offline, continue with step 2 on page OP-52.
- To end `$TERMUT1`, enter **EN**.

```
COMMAND(?): EN

$TERMUT1 ENDED AT 01:01:08
```

Operating Your Display Terminal

Varying a Terminal Online

Use the following procedure to place a terminal online. After completing this procedure, the terminal can send and receive messages.

Before You Start:

Determine the symbolic name or the hexadecimal address of the terminal you plan to use. (See "Listing System Terminal Assignments" on page OP-44 if you don't know the name or address.)

1. Load the utility \$TERMUT1:

- Using the \$L operator command, press the attention key and enter **\$L \$TERMUT1**.
- Using the session manager, select option 4.1.

```
> $L $TERMUT1
LOADING $TERMUT1 20P,01:01:18, LP= 9100,...
*** TERMINAL CONFIGURATOR ***
COMMAND(?): _
```

2. Enter ON to vary a terminal online.

```
COMMAND(?): ON
ENTER TERMINAL NAME, OR ADDRESS IN (): _
```

3. Enter the name or the hexadecimal address of the terminal. If you enter the hexadecimal address, enter it between parentheses ().

```
ENTER TERMINAL NAME, OR ADDRESS IN (): MPRTR
TERMINAL SUCCESSFULLY VARIED ON
COMMAND(?): _
```

4. Do you want to use \$TERMUT1 to vary another terminal online or do you want to end it?

- To vary another terminal online, continue with step 2.
- To end \$TERMUT1, enter **EN**.

```
COMMAND(?): EN
$TERMUT1 ENDED AT 01:01:08
```

Changing Display Terminal Keyboard Values

You can set the keyboard of a 3101 or 4978 display terminal to all uppercase characters, or upper and lowercase characters. On a 4978 display terminal you can also assign special character strings, such as an operator command, to the program function (PF) keys and you can change the meanings of most of the alphanumeric keys.

Terminology

This section uses the following special terminology. The definitions are presented here for easy reference.

Alternative code mode

A term used to describe the mode of the keyboard when the ALT key is pressed.

Character/local function code

A hexadecimal code identifying either the hexadecimal character value of an alphanumeric key, or a key's local function. Keys such as the cursor movement keys are local function keys.

Control store

A storage resident table that defines the function of the keys on a 4978 or 4980 keyboard. A control store is stored as a disk or diskette data set. A copy is loaded into storage for each 4978 or 4980 terminal attached to the Series/1 during IPL. You can have multiple control stores; a default control store loaded during IPL and others loaded on request using the LC command of \$TERMUT2.

Control store data table

A table showing the values that define each key on the keyboard and its function. This applies to the 4978 and 4980 display terminals. Figure 8 on page OP-65 shows a portion of the chart for the keyboard RPQ D02056. A complete chart is included in the general information manual for whichever keyboard is attached to your 4978 or 4980 display terminal.

Define key

For the 4978, the define key is any keyboard key that is used to put the terminal into define mode. The 4980 has a key labeled DEFINE. The 4980 DEFINE key can only be used to put the terminal into define mode.

Define mode

A special operating mode of the 4978 and 4980 display stations that allows you to change the function of, or assign a unique character string to, any key on the keyboard.

Downshift-unshifted

A term used to describe the mode of the keyboard when the SHIFT key is not pressed. In this mode, if lowercase characters are defined in the control store, the lowercase values are entered when those keys are pressed.

Operating Your Display Terminal

Changing Display Terminal Keyboard Values (*continued*)

Function ID code

A hexadecimal code that identifies the function of a key.

Image store

A storage resident table that defines the characters that can be printed by a 4974 matrix printer or displayed by a 4978 or 4980 display terminal. A copy is loaded into storage during IPL for each 4974, 4978 and 4980 that is attached to the Series/1. It is possible to have multiple image stores with one, the default, being loaded during IPL and the others being loaded to individual printers or display terminals using the LI command of the \$TERMUT2 utility.

Interrupt code

A hexadecimal code assigned to a key allowing it to interrupt, or get the attention of, EDX.

Key position

A decimal number representing a particular key on the keyboard. Every key has a number. Figure 7 on page OP-64 shows the key position numbers of the PF keys on the keyboard RPQ D02056. A similar chart in the general information manual for the keyboard attached to your 4978 or 4980 display terminal shows the key position numbers for all the keys.

The key position number is used as a cross-reference to the control store data chart in the general information manual. An example of the chart is shown in Figure 8 on page OP-65.

Scan code

A hexadecimal code assigned to a particular key. It is used by EDX to relate a key to its values in the control store table.

Upshift-shifted

A term used to describe the mode of the keyboard when the SHIFT key is pressed. In this mode, the uppercase value, such as for a capital letter, is entered when a key is pressed.

Upper and Lowercase Characters

When using a 3101, 4978 or 4980 display terminal, you have the option of entering either upper and lowercase characters or all uppercase characters from the keyboard.

If you specify upper and lowercase characters for one of these keyboards and wish to print the screens on the 4974 matrix printer, you must also set the 4974 to print upper and lowercase characters by loading a table called an *image store*. The procedure under "Loading a Control Store or Image Store" on page OP-58 shows you how to do this.

Changing Display Terminal Keyboard Values (*continued*)

3101 Display Terminal

The 3101 display terminal has a switch that you can set for either uppercase (MONO) mode or upper and lowercase (DUAL) mode. This switch is located under the setup switch cover on the top of the keyboard. It is switch number 8 in the left-most group of switches. Place this switch in the ON position (up) for DUAL mode, and in the OFF position (down) for MONO mode. (See Figure 6 for an example of the switch location.)

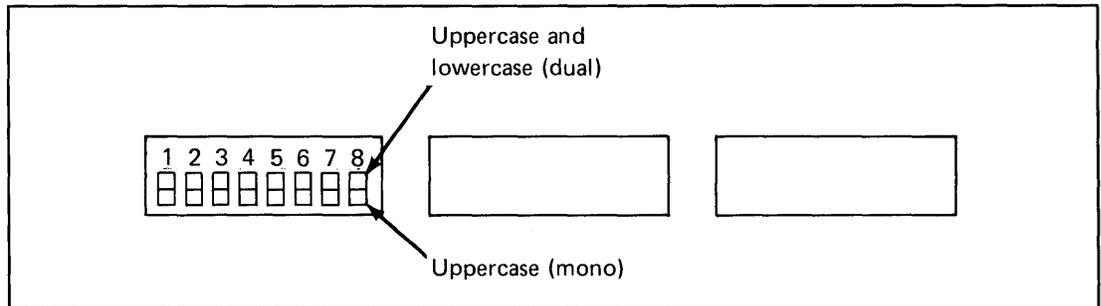


Figure 6. 3101 Setup Switches

4978 Display Stations, 4980 Display Stations, and 4974 Printers

On 4978 and 4980 display stations and 4974 matrix printers the character selection is controlled by tables called *control stores* and *image stores*. If you have 4978 keyboard RPO D02056³, you can use the control stores supplied by IBM to select between all uppercase or upper and lowercase characters. If you have 4980 keyboard, you can also use the control stores supplied by IBM to select between all uppercase or upper and lowercase characters. You can also use an image store supplied by IBM to change the character selection for a 4974 matrix printer. The procedure "Loading a Control Store or Image Store" on page OP-58 shows you how to load a control store or image store.

Note: When using a 4980 terminal, you can load an image store or control store in a program that uses the \$RAMSEC utility. For information on how to do this, refer to the *Language Reference*

If you do not have 4978 keyboard RPO D02056 you can create your own control store as described under "Adding Lowercase Characters to a Control Store" on page OP-62 and "Changing 4978 and 4980 Keyboard Values" on page OP-62.

³ An RPO number is similar to a model number. There should be a label on the underside of the keyboard showing the RPO number of the keyboard.

Operating Your Display Terminal

Changing Display Terminal Keyboard Values (*continued*)

Loading a Control Store or Image Store

Use this procedure to load a control store or image store to a 4978 or 4980 display station or to load an image store to a 4974 matrix printer. Once you have used the following procedure to load a control or image store, it remains in effect until you load a new one or IPL the Series/1.

1. Load the utility \$TERMUT2:

- Using the \$L operator command, press the attention key and enter **\$L \$TERMUT2**.
- Using the session manager, select option 4.1.

```
> $L $TERMUT2
LOADING $TERMUT2 34P,01:01:20, LP= 9100,...
COMMAND(?): _
```

2. Are you loading a control store or an image store?

- If you are loading a control store, enter **LC**.
- If you are loading an image store, enter **LI**.

```
COMMAND(?): LC
FROM DATA SET (NAME,VOLUME): _
```

Note: You do not have to enter the volume name if the control store is on the IPL volume. This volume is not always EDX002.

Changing Display Terminal Keyboard Values *(continued)*

3. Enter the name of the control store or image store that you are loading. (You do not have to enter the volume name if the control store is on the IPL volume.)

- For all uppercase characters on a 4978 display station, enter **\$4978CS0**.
- For upper and lowercase characters on a 4980 display station, enter **\$4980CS0**.
- For upper and lowercase characters on a 4978 display station, enter **\$4978CS1**.
- For the 4978 image store, enter **\$4978IS0**.
- For the 4980 image store, enter **\$4980IS0**.
- For all uppercase characters on a 4974 matrix printer, enter **\$4974IS0**.
- For upper and lowercase characters on a 4974 matrix printer, enter **\$4974IS1**.
- For a customized control or image store, enter the data set and volume names.

```
FROM DATA SET (NAME,VOLUME): $4978CS1
ENTER TERMINAL NAME (CR OR * = THIS ONE): _
```

4. Enter the name of the display terminal or printer that you are changing. If you are changing the control store for the 4978 or 4980 display station you are using, you can enter an * instead of the terminal name.

```
ENTER TERMINAL NAME (CR OR * = THIS ONE): *
COMMAND (?): _
```

5. Do you want to load a control or image store for another display terminal or printer?

- If you want to load another control or image store, continue with step 2 on page OP-58.
- If you do not, enter **EN** to end \$TERMUT2.

```
COMMAND(?): EN
$TERMUT2 ENDED AT 01:02:00
```

Operating Your Display Terminal

Changing Display Terminal Keyboard Values *(continued)*

Loading a 4980 Terminal

Use this procedure to load a 4980 terminal.

Before You Start:

Observe the following:

This procedure is required when you do not include the automatic power-on support for the 4980 during system generation time and the 4980 is powered off. It allows you to load the control and image stores with one command (LT). You can load the control or image store that was active in the terminal before it was powered off, or you can load the system default. The LT command allows you to change a control or image store without IPLing the system.

1. Load the utility \$TERMUT2:

- Using the \$L operator command, press the attention key and enter \$L \$TERMUT2.
- Using the session manager, select option 4.1.

```
> $L $TERMUT2
LOADING $TERMUT2 34P,01:01:20, LP= 9100,...
COMMAND(?): _
```

2. Enter the LT command to load a 4980 terminal.

```
COMMAND(?): LT
ENTER TERMINAL NAME (CR OR * = THIS ONE): _
```

3. Enter the name of the terminal that you are loading. To load the terminal you are presently using, enter an * or press the enter key.

```
ENTER TERMINAL NAME (CR OR * = THIS ONE): $VIG02
CURRENT IMAGE STORE DATA SET NAME IS $4980ISX
```

Changing Display Terminal Keyboard Values *(continued)*

4. If you wish to use the current image store, reply 0 to this prompt. If you wish to use the default, reply X to this prompt.

```
CURRENT IMAGE STORE DATA SET NAME IS $4980ISX
SYSTEM DEFAULT DATA SET NAME IS $4980ISO
ENTER LAST CHARACTER OF DATA SET NAME TO BE USED:0
```

5. If you wish to use the current control store, reply 0 to this prompt. If you wish to use the default, reply X to this prompt.

```
ENTER LAST CHARACTER OF DATA SET NAME TO BE USED: 0
CURRENT CONTROL STORE DATA SET NAME IS $4980CSX
SYSTEM DEFAULT DATA SET NAME IS $4980CSO
ENTER LAST CHARACTER OF DATA SET NAME TO BE USED:0
LOADING IMAGE STORE AND CONTROL STORE TO $VIG02
SUCCESSFUL COMPLETION OF 4980 LOAD
```

6. Do you want to load another terminal?

- If you want to load another terminal, continue with step 2 on page OP-60
- If you do not, enter an EN to end \$TERMUT2

```
COMMAND(?): EN
$TERMUT2 ENDED AT 01:02:00
```

Operating Your Display Terminal

Changing Display Terminal Keyboard Values (*continued*)

Adding Lowercase Characters to a Control Store

If you do not have 4978 keyboard RPQ D02056, you can use step 11 on page OP-68 through step 22 on page OP-70 of the procedure "Changing 4978 and 4980 Keyboard Values" to add lower-case characters to the control store for your keyboard. You will need the control store data chart and EBCDIC character code chart from the general information manual for your keyboard.

To add lower-case characters, first determine the hexadecimal value for each lower-case letter. These values are in an EBCDIC character code chart in the general information manual for your keyboard and on most decimal-to-EBCDIC conversion charts.

Then, starting with step 11 on page OP-68, update the downshifted-unshifted values for each alphabetic key. When you enter the character/local function code in step 16 on page OP-69, substitute the lower-case EBCDIC value for the uppercase character/local function code in the control store data table. For example, on the keyboard RPQ D02056 the letter A is key number 112 (labeled **C** in Figure 8 on page OP-65) and the Downshift-Unshifted control store table values are:

- Scan code = 4C
- Function ID code = 00
- Character/local function code = C1
- Interrupt code = 00.

The *Character/local function code* C1 is the EBCDIC value for an uppercase A. The EBCDIC value for a lowercase a is 81. Therefore, enter **81** instead of **C1** in step 16 on page OP-69 to change to a lowercase a.

Changing 4978 and 4980 Keyboard Values

Use the following procedure to enter special character strings for PF keys or special values for the alphanumeric keys, such as for lowercase characters. This procedure changes the 4978 and 4980 control store. In this procedure you will:

- Load \$TERMUT2 (step 1 on page OP-66).
- Specify a define key (step 2 on page OP-66 - step 4 on page OP-66). You do not have to specify a define key when using a 4980 terminal.
- Assign a character string to a PF key (step 6 on page OP-67 - step 10 on page OP-68).
- Clear the define key (step 11 on page OP-68 - step 18 on page OP-70). These steps can also be used to add lower-case characters to the control store.
- Save the changed control store and load it to a specified terminal (step 19 on page OP-70 - step 21 on page OP-70).
- End \$TERMUT2 (step 22 on page OP-70).

Changing Display Terminal Keyboard Values (*continued*)

Before You Start:

Observe the following requirements:

- If your 4978 display station has keyboard RPQ D02056, Figure 8 on page OP-65 has the portion of the control store chart needed for this procedure.

If it does not have keyboard RPQ D02056 shown in Figure 7 on page OP-64, you need a copy of the control store data chart in the general information manual for the keyboard you do have.

If you do not know which keyboard you have, there should be a label showing the RPQ number on the underside of the keyboard. (An RPQ number is similar to a model number.)

- For a 4978, you must allocate a 16-record data set to hold the new control store. For a 4980, allocate an 8-record data set. Although you can replace the control store supplied by IBM (FOR A 4978, \$4978CS0 or \$4978CS1. FOR A 4980, \$4980CS0) you should keep an unchanged copy for a backup. If you want the system to load the new control store automatically during IPL, you can rename the data sets, so that the new data set is named \$4978CS0 OR \$4980CS0. (See “Allocate a Data Set” on page OP-124 for the data set allocation procedure and “Renaming Volumes and Data Sets” on page OP-131 to rename the data sets.)
- Get the name of the terminal, unless you are redefining keys for the terminal you are using. (See “Listing System Terminal Assignments” on page OP-44 for instructions on determining the names of the terminals attached to your system.)

Operating Your Display Terminal

Changing Display Terminal Keyboard Values (*continued*)

- For the define key:
 - Decide which key you want to use for the define key. You can use any of the keys that are not shaded in Figure 7, or the corresponding figure for your keyboard. Note the key position number of the selected key. (The examples in this procedure use key number 66, labeled **A** in Figure 7.)

Note: The 4980 has a hardware-supported DEFINE key.

- Using the control store data chart in Figure 8 on page OP-65, or the corresponding chart for your keyboard, note the following numbers:
 - Scan code
 - Function ID
 - Character/local function code
 - Interrupt code

(For key number 66 used in the examples for this procedure, they are: 20, 20, 00, and 23.)

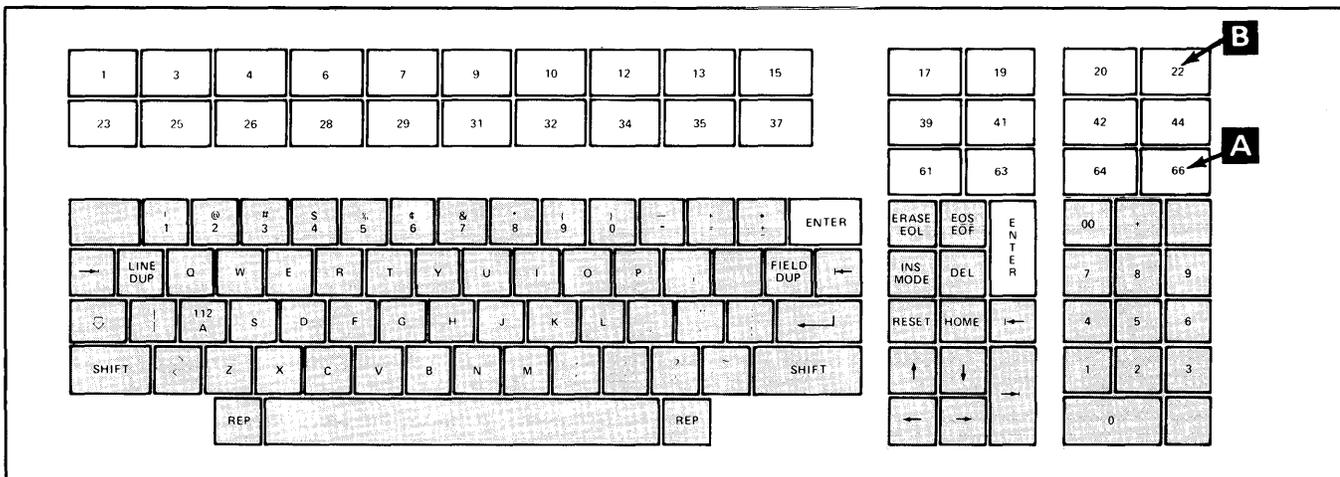


Figure 7. 4978 Display Station Keyboard, RPQ D02056

Changing Display Terminal Keyboard Values (continued)

Control Store Data																			
Downshift Unshifted										Upshift Shifted									
K e y p o s i t i o n	Scan code				K e y t o p s y m b o l		K e y t o p s y m b o l	Scan code											
	Function ID code							Function ID code											
	Character/local function code							Character/local function code											
	Interrupt code							Interrupt code											
	Character image table							Character image table											
	Row							Row											
	0	1	2	3	4	5	6	7		0	1	2	3	4	5	6	7		
1	01	20	00	01						1	81	20	00	01					
3	02	20	00	02						3	82	20	00	02					
4	03	20	00	03						4	83	20	00	03					
6	04	20	00	04						6	84	20	00	04					
7	05	20	00	05						7	85	20	00	05					
9	06	20	00	06						9	86	20	00	06					
10	07	20	00	07						10	87	20	00	07					
12	08	20	00	0B						12	88	20	00	0B					
13	09	20	00	0C						13	89	20	00	0C					
15	0A	20	00	0D						15	8A	20	00	0D					
17	0B	20	00	01						17	8B	20	00	01					
19	0C	20	00	01						19	8C	20	00	01					
20	0D	20	00	10						20	8D	20	00	10					
22	0E	20	00	11						22	8E	20	00	11					
23	0F	20	00	12						23	8F	20	00	12					
25	10	20	00	13						25	90	20	00	13					
26	11	20	00	14						26	91	20	00	14					
28	12	20	00	15						28	92	20	00	15					
29	13	20	00	16						29	93	20	00	16					
31	14	20	00	17						31	94	20	00	17					
32	15	20	00	18						32	95	20	00	18					
34	16	20	00	19						34	96	20	00	19					
35	17	20	00	1A						35	97	20	00	1A					
37	18	20	00	1B						37	98	20	00	1B					
39	19	20	00	1C						39	99	20	00	1C					
41	1A	20	00	1D						41	9A	20	00	1D					
42	1B	20	00	1E						42	9B	20	00	1E					
44	1C	20	00	11						44	9C	20	00	11					
61	1D	20	00	20						61	9D	20	00	20					
63	1E	20	00	21						63	9E	20	00	21					
64	11	20	00	22						64	9F	20	00	22					
66	20	20	00	23						66	A0	20	00	23					
67	21	20	00	24	00	00	00	00	00	00	00	00	00	00	00	00	00	00	(Blank)
112	4C	00	C1	0C	30	05	48	48	78	48	00	A							
113	4D	00	E2	00	07	48	04	30	01	48	07	00	S						
114	4E	00	C4	00	70	05	0C	0C	0C	05	70	00	D						
115	4F	00	C6	00	78	40	40	70	40	40	40	00	F						
116	50	00	C7	00	31	0C	40	40	58	0C	38	00	G						
117	51	00	C8	00	48	48	48	78	48	48	48	00	H						
118	52	00	D1	00	0B	01	01	01	41	41	34	00	J						
112	CC	00	C1	00	30	05	48	48	78	48	00	A							
113	CD	00	E2	00	07	48	04	30	01	48	07	00	S						
114	CE	00	C4	00	70	05	0C	0C	0C	05	70	00	D						
115	CF	00	C6	00	78	40	40	70	40	40	40	00	F						
116	D0	00	C7	00	31	0C	40	40	58	0C	38	00	G						
117	D1	00	C8	00	48	48	48	78	48	48	48	00	H						
118	D2	00	D1	00	0B	01	01	01	41	41	34	00	J						

Figure 8. Control Chart for 4978 Display Station, RPQ D02056

Operating Your Display Terminal

Changing Display Terminal Keyboard Values (*continued*)

1. Load the utility \$TERMUT2:

- Using the \$L operator command, press the attention key and enter \$L \$TERMUT2.
- Using the session manager, select option 4.1.

Note: If you are using a 4980 and the predefined define key, load \$TERMUT2 and skip to step 6.

```
> $L $TERMUT2
LOADING $TERMUT2 34P,01:01:20, LP= 9100,...

COMMAND(?): _
```

2. Enter the AD command to specify a *define* key.

If your display terminal already has an active define key, skip to step 6 on page OP-67.

```
COMMAND (?): AD
ENTER SCAN CODE OF THE KEY TO BE ASSIGNED
AS THE DEFINE KEY (HEX): _
```

3. Enter the scan code. (The code for our example is 20)

```
ENTER SCAN CODE OF THE KEY TO BE ASSIGNED
AS THE DEFINE KEY (HEX): 20
ENTER TERMINAL NAME (CR OR * = THIS ONE): _
```

4. Enter an * if you are assigning a define key to the terminal you are using.

Enter the name of the terminal to which you are assigning the define key if you are assigning one for another terminal.

```
ENTER TERMINAL NAME (CR OR * = THIS ONE): *
```

5. If you are using a 4980, enter the nonshift, shift, or alternative code mode. If not, skip to step 6 on page OP-67. For a definition of nonshift, shift and alternative code mode, refer to "Terminology" on page OP-55.

```
ENTER TERMINAL NAME (CR OR * = THIS ONE): *
ENTER NONSHIFT, SHIFT, OR ALTERNATIVE CODE
MODE (N/S/A): N
COMMAND(?): _
```

Changing Display Terminal Keyboard Values (*continued*)

6. Press the define key to enter define mode.



(66)

Notes:

- You do not have to end \$TERMUT2 to enter define mode.
- When you press the define key on a 4978 terminal, indicator light 2 on the right-hand side of the display screen comes on to show that you are in define mode. When you press the define key on a 4980 terminal, the letter D appears on the status line.

7. Press the key you are changing. The example uses key 22 - labeled **B** in Figure 7 on page OP-64 .



(22)

Notes:

- If you press the define key again without entering any values for the selected key, the current meaning of the key is lost.
- Any key that is pressed after the key you are changing will be assigned as part of the changed key's value.

8. Enter the character string you are assigning to the key. (The characters are not displayed on the screen.)

Notes:

- You cannot enter the value of the attention key as part of the character string.
- You can include the **ENTER** key as part of the character string on some keyboards, such as RPQ D02056.



\$L \$FSEDIT

Operating Your Display Terminal

Changing Display Terminal Keyboard Values (*continued*)

9. Press the define key to enter the new character(s) into the control store for your terminal and leave define mode, turning off indicator light 2.

(66)

10. Press the newly defined PF key without pressing the attention key first. This displays the character string entered so you can verify it.

(22)

If you entered the correct character string, you can now use the redefined key.

\$L \$FSEDIT

If you entered the wrong information, you can do one of the following:

- Repeat the portion of the procedure, starting with step 6 on page OP-67.
- Reload the original control store for your terminal. The default control store for RPQ D02056 is named \$4978CS0 and is on volume EDX002. (Refer to the \$TERMUT2 LC command in the *Operator Commands and Utilities Reference* for information on loading a control store.)
- IPL the system.

11. Enter C to clear the define key or to change any key definition (except a PF key).

If you do not want to redefine the define key, continue with step 19 on page OP-70 or step 22 on page OP-70.

COMMAND(?): C

ENTER TERMINAL NAME (CR or * = THIS ONE): _

Note: An active define key is not desirable for normal terminal usage because, if you press it the control store is altered.

Changing Display Terminal Keyboard Values *(continued)*

- 12.** Enter an * if you are clearing a key on the terminal you are using.

Enter the name of the terminal and press the enter key if you are clearing a key on another terminal.

```
ENTER TERMINAL NAME (CR or * = THIS ONE): *  
ENTER SCAN CODE OF THE KEY TO BE REDEFINED (HEX): _
```

- 13.** Enter the scan code. If you are using a 4980 terminal, continue with the next step. If not, skip to step 15.

```
ENTER SCAN CODE OF THE KEY TO BE REDEFINED (HEX): 20
```

- 14.** If you are using a 4980, enter the nonshift, shift, or alternative code mode.

```
ENTER NONSHIFT, SHIFT, OR ALTERNATIVE CODE MODE (N/S/A): N  
ENTER NEW FUNCTION ID (HEX):
```

- 15.** Enter the function ID.

```
ENTER NEW FUNCTION ID (HEX): 20  
ENTER NEW CHARACTER/FUNCTION CODE (HEX): _
```

- 16.** Enter the character local function code.

```
ENTER NEW CHARACTER/FUNCTION CODE (HEX): 00  
ENTER NEW INTERRUPT CODE (HEX): _
```

- 17.** Enter the interrupt code.

```
ENTER NEW INTERRUPT CODE (HEX): 23  
ANOTHER KEY? _
```

Operating Your Display Terminal

Changing Display Terminal Keyboard Values *(continued)*

18. Do you want to clear another key?

- If so, enter a **Y** and continue with step 11 on page OP-68.
- If not, enter an **N** and continue with step 19 or step 22.

```
ANOTHER KEY? N  
COMMAND(?): _
```

19. Enter **SC** to save the new key values. If you do not want to save the control store, continue with step 22.

```
COMMAND(?): SC  
SAVE DATA SET (NAME,VOLUME): _
```

20. Enter the data set and volume names of the data set where you want the new control store saved. (If the data set is on the IPL volume, you do not have to enter the volume name.)

```
SAVE DATA SET (NAME,VOLUME): MYDSET,MYVOL  
ENTER TERMINAL NAME (CR OR * = THIS ONE): _
```

21. Enter an ***** and press the enter key if the control store is for the terminal you are currently using. Enter the name of the terminal and press the enter key if the control store is for another terminal. (\$TERMUT2 loads a copy of the control to the specified terminal and saves it in the specified data set.)

```
ENTER TERMINAL NAME (CR OR * = THIS ONE): *  
COMMAND(?): _
```

22. Do you want to define a different set of keys for this or another display terminal?

- If so, continue with step 6 on page OP-67.
- If not, enter **EN** to end \$TERMUT2.

```
COMMAND(?): EN  
$TERMUT2 ENDED AT 01:02:00
```

Chapter 4. Using the Session Manager

This chapter introduces you to the session manager. After reading this chapter you will be able to load and log on to the session manager, select an option, manage the menus, and log off.

This chapter also has a table that cross references the system utilities supported by the session manager to the appropriate menu option.

- “Introduction to the Session Manager” on page OP-72
- “Loading the Session Manager” on page OP-77
- “Logging On” on page OP-79
- “Selecting an Option” on page OP-80
- “Suspending and Restarting the Session Manager” on page OP-82
- “Clearing the Screen” on page OP-83
- “Returning to the Previous Screen” on page OP-83
- “Returning to the Primary Option Menu” on page OP-83
- “Logging Off” on page OP-84

Using the Session Manager

Introduction to the Session Manager

The session manager provides access to system utilities and application programs from a display terminal. It uses a series of menu screens to direct you to the system utility you need. It also prompts you for parameters, such as data set names, that are required by the specified option.

The session manager must be active at your display terminal before you can use it. This can be accomplished by either loading it on the terminal you are using, or having it loaded automatically when you IPL the system. See "Loading the Session Manager" on page OP-77 for instructions on loading the session manager by either method.

Menus

The session manager menus, or display screens, list system facilities available through the session manager. They also display prompts for required parameters.

The session manager has the following menus:

Logon/logoff menu:

The logon menu prompts you for a user ID and an optional alternate session menu if you are logging on to the session manager, or for the word LOGOFF if you are ending your session (logging off).

Your user ID must be 1 to 4 unique characters, such as your initials. The session manager uses your ID as part of the data set names of six work data sets that it allocates for your session. It does not use your ID as a password to verify that you are authorized to use the system.

The alternate session menu is an alternate menu that you want displayed instead of the primary option menu. An alternate session menu is available only if your copy of the session manager has been customized. (Refer to the *Customization Guide* for instructions on adding menus to the session manager.)

An example of the logon menu is shown in the procedure "Logging On" on page OP-79.

Primary option menu:

The primary option menu lists the 10 primary options supplied with the session manager. If your session manager has been customized, you may have more than 10 options, or the options may be different from the 10 provided. The 10 basic options are:

1. TEXT EDITING: Accesses the \$FSEDIT text editor.
2. PROGRAM PREPARATION: Accesses the program preparation utilities.
3. DATA MANAGEMENT: Accesses the utilities for managing data on disk, diskette, or tape.
4. TERMINAL UTILITIES: Access the terminal support utilities.

Introduction to the Session Manager (*continued*)

5. **GRAPHICS UTILITIES:** Access the utilities that generate, maintain, and display two- and three-dimensional fixed graphic backgrounds, and stores them in data sets.
6. **EXEC PROGRAM/UTILITY:** Allows you to load any program. The program can be an EDX system program, an EDX utility, or an application program.
7. **EXEC \$JOBUTIL PROC:** Allows you to load a previously built \$JOBUTIL procedure.
8. **COMMUNICATION UTILITIES:** Access the utilities that support communications.
9. **DIAGNOSTIC AIDS:** Access the utilities that help with problem determination.

Secondary option menu:

A secondary option menu lists the utilities that are available under the related primary option. Primary options 2, 3, 4, 5, 8, and 9 have secondary option menus.

Parameter input menu:

The parameter input menus prompt you for parameters, such as a data set and volume name, that are required by the requested utility.

Primary options 1, 6, and 7 have a parameter input menu but no secondary option menu.

Custom menus:

You can add your own custom menus which give you access to your application programs with the session manager. Refer to the *Customization Guide* for instructions on customizing the session manager.

Using the Session Manager

Introduction to the Session Manager (*continued*)

Utilities Supported

Figure 9 lists the EDX system utilities that are supported by the session manager, and the primary and secondary option numbers for each. (See Figure 24 on page OP-346 for a table which includes a list of the jobs that can be done with the session manager, including the appropriate primary and secondary option numbers for each.)

Note: The session manager menus are independent of the EDX supervisor installed on your EDX system. Therefore, all the utilities listed on the menu screens may not be a part of your system.

Utility	Description	Options
\$ARJE	Advanced RJE program and utilities	8.10
\$BSTRCE	Trace I/O activity on a BSC line	8.1
\$BSCUT1	Format BSC trace files	8.2
\$BSCUT2	Communications I/O exerciser	8.3
\$CHANUT1	Channel attach utility	8.9
\$COBOL	COBOL language compiler	2.4
\$COMPRES	Compress disk, diskette, or volume	3.4
\$COPY	Copy data set or volume	3.5
\$COPYUT1	Copy data set with allocation	3.3
\$DASDI	Format disk or diskette	3.6
\$DICOMP	Display and change graphics profiles	5.2
\$DIINTR	Graphics interpreter	5.3
\$DISKUT1	Allocate, delete, list data set directory data	3.1
\$DISKUT2	Patch, dump, or list a data set or program a data set or program	3.2, 9.2
\$DIUTIL	Maintain partitioned data base	5.1
\$DUMP	Format and list saved environment	9.1
\$EDXASM	Event Driven Language Compiler	2.1

Figure 9 (Part 1 of 3). Session Manager Options by Utility

Introduction to the Session Manager *(continued)*

Utility	Description	Options
\$EDXASM/\$EDXLINK	Compile and link	2.2
\$EDXASM/\$XPSLINK	Compile and link a supervisor program	2.13
\$EDXLINK	Linkage editor	2.7
\$FONT	Process 4974, 4978, and 4980 character image tables	4.5
\$FORT	FORTRAN language compiler	2.5
\$FSEEDIT	Full-screen editor	1
\$GPIBUT1	General purpose interface bus utility	4.9
\$HCFUT1	Interact with Host Communications Facility	8.8
\$HXUT1	H exchange diskette utility	3.11
\$IAMUT1	Indexed Access Method Utility	3.9
\$IMAGE	Define 4978, 4979, 4980 or 3101 screen image	4.4
\$INITDSK	Initialize disk or diskette, and volume control	3.7
\$IOTEST	Test sensor I/O; list hardware configuration	9.3
\$JOBUTIL	Job stream processor	7
\$JOBQUT	Job queue processing controller	10.1
\$MOVEVOL	Disk volume dump and restore	3.8
\$MSGUT1	Message-source processing	2.14
\$PASCAL/\$EDXLINK	Pascal language compile and link	2.12
\$PFMAP	Display 4978 and 4980 PF key codes	4.6
\$PLI/\$EDXLINK	PLI language compile and link	2.6
\$PREFIND	Prefind data sets and EDL overlays	2.11
\$PRT2780	2780 spooled RJE file printer	8.6
\$PRT3780	3780 spooled RJE file printer	8.7

Figure 9 (Part 2 of 3). Session Manager Options by Utility

Using the Session Manager

Introduction to the Session Manager (*continued*)

Utility	Description	Options
\$RJE2780	2780 remote job entry to host	8.4
\$RJE3780	3780 remote job entry to host	8.5
\$SPLUT1	Spool utility	4.7
\$SUBMIT	Job queue job submission utility	10.2
\$S1ASM	Series/1 assembler	2.3
\$S1S1UT1	Series/1 to Series/1	4.8
\$TAPEUT1	Tape management	3.10
\$TERMUT1	Change terminal parameters	4.1
\$TERMUT2	Change 4974, 4978, and 4980 image or control store	4.2
\$TERMUT3	Send message to a terminal	4.3
\$UPDATE	Converting Series/1 programs	2.9
\$UPDATEH	Convert host system programs	2.10
\$VERIFY	Verify Indexed Access Method Files	9.4
\$XPSLINK	Link a supervisor program	2.8

Figure 9 (Part 3 of 3). Session Manager Options by Utility

Loading the Session Manager

You can load (start) the session manager manually using the \$L operator command or automatically during initial program load (IPL).

Loading with \$L

Use the following procedure to load the session manager to run at your display terminal.

1. Press the attention key and type \$L \$SMMAIN to load the session manager.

```
>$L $SMMAIN_
```

2. Press the enter key and the system displays the session manager logon screen.

```
$SMMLG: THIS TERMINAL IS LOGGED ON TO THE SESSION MANAGER  
08:55:31  
ENTER 1-4 CHAR USER ID ==> _ 04/19/82  
(ENTER LOGOFF TO EXIT)  
  
ALTERNATE SESSION MENU ==>  
(OPTIONAL)
```

Using the Session Manager

Loading the Session Manager (*continued*)

Loading Automatically

Use the following procedure to specify that the system load the session manager automatically during initial program load (IPL). When the system loads the session manager automatically during IPL, the session manager is active at all display terminals.

1. Press the attention key and enter **\$L \$DISKUT1**.

```
> $L $DISKUT1
LOADING $DISKUT1 49P, 08:04:28, LP= 0000,...

$DISKUT1 - DATA SET MGMT. UTILITY I
USING VOLUME EDX002

COMMAND (?): _
```

2. Enter **RE \$SMINIT \$INITIAL** to change the name of \$SMINIT to \$INITIAL.

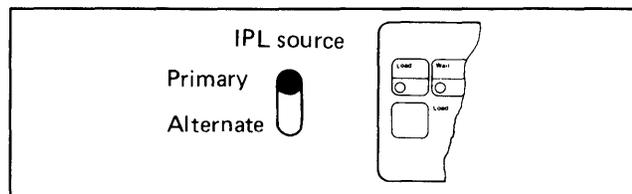
```
COMMAND(?): RE $SMINIT $INITIAL_
RENAME COMPLETED
COMMAND (?): _
```

3. Enter **EN** to end \$DISKUT1.

```
COMMAND(?): EN_

$DISKUT1 ENDED AT 08:06:30
```

4. IPL the system to load the session manager to all display terminals.



Logging On

Use the following procedure to log on to the session manager.

Before You Start:

Observe the following requirements:

- The session manager must be active at your display terminal before you can log on to it. If the logon menu is not displayed on your display screen, you must load the session manager as described under “Loading the Session Manager” on page OP-77.
- Determine your user ID. It must be 1-4 characters in length and should be unique, such as your initials.
- Get the data set and volume names for the alternate session menu if you have one on your system and want to use it instead of the session manager primary option menu.

1. Type your user ID on the ID prompt line. If you are specifying an alternate session menu, continue with step 2 before pressing the enter key.

```
$SMMLOG: THIS TERMINAL IS LOGGED ON ...  
ENTER 1-4 CHAR USER ID ==> WJB  
(ENTER LOGOFF TO EXIT)  
  
ALTERNATE SESSION MENU ==>  
(OPTIONAL)
```

2. Enter the data set and volume names of an alternate session menu if you want that menu instead of the session manager primary option menu. Enter the names in the format **dsname,volname**, where **dsname** is the name of the data set containing the alternate menu, and **volname** is the name of the volume where it resides.

```
ALLOCATING SESSION MANGER WORK DATA SETS
```

```
$SMMPRIM: SESSION MANAGER PRIMARY OPTION MENU  
ENTER/SELECT PARAMETERS: PRESS PF3 TO EXIT  
SELECT OPTION ==>  
1 - TEXT EDITING  
2 - PROGRAM PREPARATION  
3 - DATA MANAGEMENT  
4 - TERMINAL UTILITIES  
5 - GRAPHICS UTILITIES  
6 - EXEC PROGRAM/UTILITY  
7 - EXEC $JOBUTIL PROC  
8 - COMMUNICATION UTILITIES  
9 - DIAGNOSTIC AIDS  
10 - BACKGROUND JOB CONTROL UTILITIES
```

Using the Session Manager

Selecting an Option

Use the following procedure to select a session manager option.

Before You Start:

Display the primary option menu on your terminal.

1. Do you want to view the secondary option menu for the primary option menu you are selecting? You can bypass the secondary option menu by entering both the primary and secondary option numbers on the primary option menu. (Primary options 1, 6, and 7 do not have secondary option menus.)
 - If you want to view the secondary option menu, continue with step 1(a).
 - If you want to skip the secondary option menu, continue with step 1(b).

- 1(a). To view the secondary option menu, enter the primary option number as shown in the example.

If the session manager displays a secondary option menu, continue with step 2.

If it displays a parameter input menu, continue with step 3.

```
$SMMPRIM: SESSION MANAGER PRIMARY OPTION MENU
ENTER/SELECT PARAMETERS: PRESS PF3 TO EXIT
SELECT OPTION ==> 2
 1 - TEXT EDITING
 2 - PROGRAM PREPARATION
 3 - DATA MANAGEMENT
 4 - TERMINAL UTILITIES
 5 - GRAPHICS UTILITIES
 6 - EXEC PROGRAM/UTILITY
 7 - EXEC $JOBUTIL PROC
 8 - COMMUNICATION UTILITIES
 9 - DIAGNOSTIC AIDS
10 - BACKGROUND JOB CONTROL UTILITIES
```

Selecting an Option (*continued*)

- 1(b).** To bypass the secondary option menu, enter the primary and secondary option numbers in the format *x.y*, where *x* is the primary option number and *y* is the secondary option number. The two numbers must be separated by a period (.).

If the session manager displays a parameter input menu, continue with step 3.

If it loads the utility and this book contains a procedure for the job you are doing, continue with that procedure. Otherwise, refer to the *Operator Commands and Utilities Reference* for information about the utility.

```
$$MMPRIM: SESSION MANAGER PRIMARY OPTION MENU
ENTER/SELECT PARAMETERS:  PRESS PF3 TO EXIT
SELECT OPTION ==> 2.1
 1 - TEXT EDITING
 2 - PROGRAM PREPARATION
 3 - DATA MANAGEMENT
 4 - TERMINAL UTILITIES
 5 - GRAPHICS UTILITIES
 6 - EXEC PROGRAM/UTILITY
 7 - EXEC $JOBUTIL PROC
 8 - COMMUNICATION UTILITIES
 9 - DIAGNOSTIC AIDS
10 - BACKGROUND JOB CONTROL UTILITIES
```

- 2.** Enter the option number of the utility that you want to use. (The example shows the secondary option menu for primary option 2 - PROGRAM PREPARATION - and we are selecting the \$EDXASM compiler, option 1.)

If the session manager displays a parameter input menu, continue with step 3.

If it loads the utility and this book contains a procedure for the job you are doing, continue with that procedure. Otherwise, refer to the *Operator Commands and Utilities Reference* for information about the utility.

```
$$M02 SESSION MANAGER PROGRAM PREPARATION...
ENTER/SELECT PARAMETERS:  PRESS PF3 TO...
SELECT OPTION ==> 1
 1 - $EDXASM COMPILER
 2 - $EDXASM/$EDXLINK
 3 - $$IASM ASSEMBLER
 4 - $COBOL COMPILER
 5 - $FORT FORTRAN COMPILER
 6 - $PLI COMPILER/$EDXLINK
 7 - $EDXLINK LINKAGE EDITOR
 8 - $XPDLINK LINKAGE EDITOR FOR SUPERVISORS
 9 - $UPDATE
10 - $UPDATEH (HOST)
11 - $PREFIND
12 - $PASCAL COMPILER/$EDXLINK
13 - $EDXASM/$XPDLINK FOR SUPERVISORS
14 - $MSGUT1 MESSAGE SOURCE PROCESSING UTILITY
```

Using the Session Manager

Selecting an Option (*continued*)

3. Enter the requested information. The parameters must be entered in the format expected by that utility or program. (The example shows the parameter input menu for primary option 2, secondary option 1 - \$EDXASM COMPILER.)

```
$SMM0201:SESSION MANAGER $EDXASM PARAMETER INPUT MENU  
ENTER/SELECT PARAMETERS:          PRESS PF3 TO RETURN
```

```
SOURCE INPUT (NAME,VOLUME) ==> MYSRC,MYVOL
```

```
OBJECT OUTPUT (NAME,VOLUME) ==> MYOBJ,MYVOL
```

```
OPTIONAL PARAMETERS ==> LI TERM2  
(SELECT FROM THE LIST BELOW)
```

```
-----  
AVAILABLE PARAMETERS:  ABBREVIATION:  DESCR  
NOLIST                 NO             USED  
LIST TERMINAL-NAME     LI TERMINAL-NAME  USE  
ERRORS TERMINAL-NAME  ER TERMINAL-NAME  USE  
CONTROL DATA SET,VOLUME  CO DATA SET,VOLUME  $EDX  
OVERLAY #              OV #             # IS
```

```
DEFAULT PARAMETERS:  
LIST $SYSPRTR CONTROL $EDXL,ASMLIB OVERLAY 4
```

Suspending and Restarting the Session Manager

You can stop the session manager to enter operator commands, such as \$L, without having to reload the session manager when you are finished. Suspending the session manager is quicker than logging off then logging back on when you want to use it again. When restarted, the session manager returns to the last menu you were using.

Suspending

Press program function key 1 (PF1) to suspend the session manager. The example shows the system response after you press the PF1 key.

1

```
ENTERING SYSTEM COMMAND MODE -  
TO REENTER THE SESSION MANAGER,  
DEPRESS ATTN KEY AND ENTER '$SM':
```

Suspending and Restarting the Session Manager *(continued)*

Restarting

Use the following procedure to restart the session manager if you used the PF1 key to suspend it.

Press the attention key and enter **\$SM**.

The session manager displays the last session manager menu you were using.

```
> $SM
.
```

Clearing the Screen

2

Press program function key 2 (**PF2**) to erase the entries you have typed on a session manager menu screen. The session manager displays the menu as it appeared before you typed your entries.

Returning to the Previous Screen

3

Press program function key 3 (**PF3**) to return to the previous screen.

Returning to the Primary Option Menu

4

Press program function key 4 (**PF4**) to return to the Primary Option Menu from any menu.

Using the Session Manager

Logging Off

Use the following procedure to log off the session manager. You have the option of leaving the session manager active at the display terminal.

Before You Start:

Return to the primary option menu.

1. Press the **PF3** key to exit the primary option menu.

3

```
TERMINATING THE SESSION MANAGER  
DO YOU WANT TO SAVE WORK DATASETS (Y/N): _
```

2. Do you want to save the work data sets?. If you save them, the next time you log on with the same user ID, the contents of the work data sets will be the same as they are now. For example, if you used \$FSEDIT during this session and you save the work data sets, the current contents of the \$FSEDIT work data set will be available the next time you sign-on.

```
TERMINATING THE SESSION MANAGER  
DO YOU WANT TO SAVE WORK DATASETS (Y/N): Y  
  
WORK DATA SETS SAVED
```

- Enter a **Y** to save the work data sets.
- Enter an **N** to delete the work data sets.

3. Do you want to leave the session manager active at the display terminal?

- If so, you are finished with this procedure.
- If not, enter **LOGOFF**.

```
$SMMLOG: THIS TERMINAL IS LOGGED ON TO THE...  
  
ENTER 1-4 CHAR USER ID ==> LOGOFF  
(ENTER LOGOFF TO EXIT)  
  
ALTERNATE SESSION MENU ==>  
(OPTIONAL)
```

```
SESSION MANAGER TERMINATED  
ENTERING SYSTEM COMMAND MODE  
TO RELOAD THE SESSION MANGER: $L $SMMAIN
```

Chapter 5. Storing and Handling Information

Your Series/1 does many of the jobs people do in any business office. One of these jobs is the filing of information. The Series/1 files (stores) two types of information: programs and data. A computer program is a collection of computer language instructions that control what a computer does, and data is information that the programs use, such as customer addresses, payroll records, and production figures.

This chapter explains how information is stored on the Series/1 and contains procedures for storing and handling it as follows:

- “How Information is Stored” on page OP-86
- “Preparing Disks, Diskettes, and Tapes for Use” on page OP-91
- “Initializing Diskettes” on page OP-92
- “Initializing Tapes” on page OP-112
- “Allocating Volumes and Data Sets” on page OP-121
- “Renaming Volumes and Data Sets” on page OP-131
- “Deleting Volumes and Data Sets” on page OP-136
- “Compressing Volumes and Devices” on page OP-142
- “Copying Data to/from a Non-EDX System” on page OP-171
- “Copying Data on EDX Systems” on page OP-148
- “Listing Volume and Data Set Directory Information” on page OP-181
- “Determining Volume Size and Free Space” on page OP-205
- “Creating a Stand-Alone or \$TRAP Dump Diskette” on page OP-207
- “Entering/Editing Data with \$FSEEDIT” on page OP-210

Storing and Handling Information

How Information is Stored

The way the Series/1 stores information can be compared to an office filing system. In an office filing system, you place individual pieces of information, such as employee addresses, in file folders. You then put the folders into a specific drawer of a filing cabinet. In some filing systems you even update an index showing where the information is filed.

On a Series/1, pieces of information are called records; the file folders are called data sets; the drawers are called volumes; the filing cabinet is called a disk, diskette, or tape; and the indexes are called directories. Each volume has a directory which lists the data sets on the volume, and each disk, diskette, and tape has a directory which lists the volumes on that device.

EDX records the individual records on the disk, diskette, or tape in much the same way as a tape recorder records your voice on a cassette tape. EDX uses data sets and volumes as a means of organizing the information on the disk, diskette, or tape, and uses the directories to find where the data sets and volumes are located.

Figure 10 on page OP-89 illustrates the relationship between the two filing systems as explained in the following paragraphs.

Records

W. Borden
R. Kenney
J. McCauley

In an office filing system, a piece of information, or record, can be anything from a person's name, to his complete life history. However, on the Series/1, a record has a fixed size of 256 bytes. (A byte is one computer character.) As a means of comparison, one record can hold 256 letters from the alphabet.

Records are the divisions of a data set and are the smallest unit of information available to a program. Therefore, a data set is said to have X number of 256-byte records. EDX numbers records beginning with 1.

How Information is Stored (*continued*)

Data Sets

DEPT 301
DEPT 302
DEPT 303

Data sets⁴ are the “file folders” of an information storage organization. You must use an EDX utility program to define (allocate) a data set before you can store information in it. When you allocate a data set, you assign it a name, specify the number of records it is to contain, and the type of information it will store. The name of a data set can be from one to 8 alphanumeric characters in length and must be unique to the volume containing the data set.⁵ In the example in Figure 10 on page OP-89, the data sets are named DEPT301, DEPT302, and DEPT303.

A data set can hold one of two types of information: data and program. Data-type data sets are used to store information such as address files, and program source statements. Program-type data sets are used to hold executable programs. A program is a collection of computer language instructions that control what a computer does. A data set is part of a volume and can consist of from one to as many records as the volume can hold.

Volumes

DIVI01
DIVI02
DIVI03

Volumes are the drawers of an information storage system. You must use an EDX utility to define (allocate) a volume before you can allocate data sets on it. When you allocate a volume, you assign it a name and specify its size; the number of records it contains. The name of a volume can be from 1 to 6 characters long and must be unique.⁶ In the example in Figure 10 on page OP-89 the volumes are named DIVI01, DIVI02, and DIVI03.

A volume consists of 3 to as many records as the disk, diskette, or tape can hold. Disks and diskettes can have multiple volumes, but tapes can only have one volume.

⁴ On some of the utility program display screens, data sets are called members.

⁵ If you enter a name of more than 8 characters EDX shortens it to the first 8 characters.

⁶ If you enter a volume name longer than 6 characters, EDX shortens it to the first 6 characters.

Storing and Handling Information

How Information is Stored (*continued*)

Directories

Volume
directory

Data set
directory

Directories are the indexes to an information filing system. EDX has two types of directories; volume and data set. A volume directory contains information about the volumes on an individual disk, diskette, or tape device (the file cabinets of our filing system). A data set directory contains information about the data sets on an individual volume. Volume directories are set up when the disk, diskette, or tape is prepared for use or *initialized*, and data set directories are set up when the volume is allocated.

Volume Label and TAPEID

PAYROLL



Disk, diskette and tape devices are the filing cabinets of an information storage system. EDX does not require that a disk have a unique identification. However, it does require a unique identification for diskette and tape drives.

The identification for a diskette is called a volume label (sometimes called as VOL1 on system utility screens). A volume label can be from 1 to 6 characters in length and is specified when the diskette is prepared for use (initialized). The volume label for the diskette in the example in Figure 10 on page OP-89 is PAYROLL.

The identification for a tape drive is called a TAPEID. A TAPEID can be from 1 to 6 characters in length and is specified on the TAPE statement during system generation.

How Information is Stored (continued)

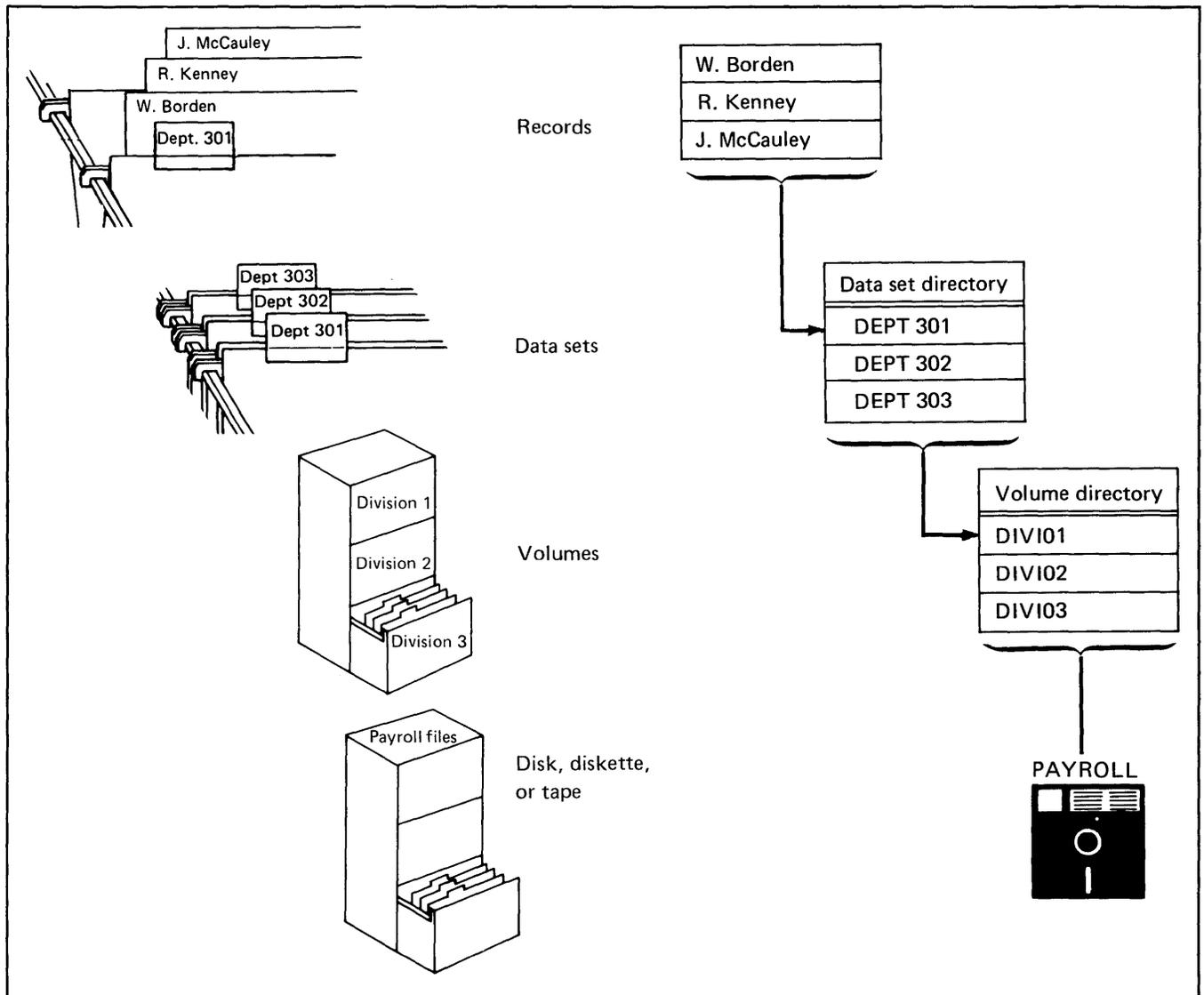


Figure 10. Series/1 Data Organization 1

Storing and Handling Information

Diskette Formats

The Event Driven Executive uses three types of diskettes: the Diskette 1, the Diskette 2, and the Diskette 2D.⁷ A diskette must be initialized to a specified format and number of bytes-per-sector before it can be used.

Bytes-per-sector, or sector size, refers to the number of bytes of information that can be recorded on one sector of a disk or diskette. A sector is a fixed-length physical location on a disk or diskette surface. The sector size to which a diskette is initialized is usually determined by the application program that will use the data and the limitations of the diskette drives attached to the Series/1. The best choice is usually the largest sector size that can be read on your system, because the larger sector size means that the diskette can hold more data.

On an EDX system, you can initialize diskettes to one of the following three formats:

Event Driven Executive format

This format is for diskettes that you will use with the Event Driven Executive. Diskettes initialized to this format can have 128, 256, or 1024 bytes per sector.

The procedure to initialize a diskette to this format is under "Initialize for EDX Format" on page OP-93.

IBM Standard for Information Interchange format

This format is for diskettes that will be used to exchange data to or from a non-EDX system. It is not for diskettes that will be used with EDX programs. Diskettes initialized to this format can have 128, 256, 512, or 1024 bytes-per-sector.

For basic exchange data sets, you must initialize a Diskette 1 to 128 bytes-per-sector. For H-exchange data sets, you must initialize a Diskette 2D to 256 bytes-per-sector.

Whether you use a basic exchange or H-exchange diskette is determined by the hardware on your Series/1 and the usage of the diskette on the non-EDX system. The H-exchange is usually preferable because it holds more data. However, you must have a diskette drive that can read a Diskette **0D**.

The procedure for initializing diskettes to this format is under "Initialize to IBM Standard for Information Interchange" on page OP-97.

Bytes-per-sector format

This format is for diskettes that will be used on a system other than a Series/1 Event Drive Executive. Diskettes initialized to this format can have 128, 256, 512, or 1024 bytes-per-sector. The procedure for this format is under "Initialize to Bytes Per Sector Format" on page OP-102.

⁷ The Event Driven Executive recognizes double density only at 256 and 1024 bytes-per-sector.

Preparing Disks, Diskettes, and Tapes for Use

You must prepare disks, diskettes and tapes before you can store information on them. They are prepared for use as follows:

1. Initialize the recording surface. This includes specifying the volume label on a diskette or tape and allocating a volume directory.

You do not normally allocate disk directories as part of your daily operations because you have to set up all the information on the disk again. If you have to initialize a disk, refer to the \$INITDSK utility in the *Operator Commands and Utilities Reference*.

- To assign a volume label to a diskette, see “Initializing Diskettes” on page OP-92.
- To assign a volume label to a tape see, “Initializing Tapes” on page OP-112.
- To allocate a directory on a diskette see, “Initialize a Diskette Volume Directory” on page OP-106.
- To allocate a directory on a tape see, “Initializing Tapes” on page OP-112.

2. Allocate the volume(s), including the data set directory.

- To allocate a volume on a disk and all diskettes except H-exchange diskettes, see “Allocate a Volume” on page OP-121.
- To allocate a volume on an H-exchange diskette, see “Initialize an H Exchange Volume” on page OP-111.
- To allocate a volume on a tape, see “Initializing Tapes” on page OP-112.
- To allocate a data set directory on a volume, see “Allocate a Volume” on page OP-121 or theIV command of \$INITDSK in the *Operator Commands and Utilities Reference*.

3. Allocate the data sets.

To allocate a data set, see “Allocating Volumes and Data Sets” on page OP-121.

Storing and Handling Information

Initializing Diskettes

Use one of the following procedures to initialize a diskette:

- “Initialize for EDX Format” on page OP-93
- “Initialize to IBM Standard for Information Interchange” on page OP-97
- “Initialize to Bytes Per Sector Format” on page OP-102.

You can reinitialize a diskette to another format. However, any data currently on the diskette will be lost. The chart in Figure 11 shows the valid format combinations.

When selecting sector size, remember that the sector size is determined by the application program that will use the data and the limitations of the diskette drives attached to the Series/1. The best choice is usually the largest sector size that can be read on your system because the larger sector size means that the diskette can hold more data.

Formats	128 bytes/sector		256 bytes/sector		512 bytes/sector		1024 bytes/sector	
	S	D	S	D	S	D	S	D
Event Driven Executive	Y	N/A	Y	Y	N/A	N/A	N/A	Y
Standard for Information Interchange	Y	N/A	Y	Y	Y	Y	N/A	Y
Sector size	Y	N/A	Y	Y	Y	Y	N/A	Y

Legend:

S = Single-density (Diskette 1 and Diskette 2)

D = Double-density (Diskette 2D on 4965/4966 only)

Figure 11. Diskette Format Combinations

Initializing Diskettes (*continued*)

Initialize for EDX Format

Use this procedure to initialize a diskette to be used with the Event Driven Executive operating system and utilities.

Before You Start

Observe the following requirements:

- Insert the diskette to be initialized into the diskette device. If you are using a 4966 Magazine unit, the diskette must be in slot number 1.
- Obtain the hexadecimal address of the diskette unit. See "Listing the Devices on Your System" on page OP-318 for the procedure for obtaining this number.
- Determine the sector size - bytes-per-sector. See Figure 11 on page OP-92 for valid sector sizes for the type of diskette you are initializing.

1. Load the utility \$DASDI:

- Using the \$L operator command, press the attention key and enter \$L \$DASDI.
- Using the session manager, select option 3.6.

```
> $L $DASDI
LOADING $DASDI      7P,08:30:06, LP= 0000, PART=2
DIRECT ACCESS DEVICE INITIALIZATION
DISK INITIALIZATION OPTIONS:
  0=CREATE STAND-ALONE DUMP/$STRAP DISKETTE
  1=DISKETTE INITIALIZATION
  2=4962 DISK INITIALIZATION
  3=4963 DISK INITIALIZATION
  4=4967 DISK INITIALIZATION
  5=DDSK DISK INITIALIZATION
  9=EXIT DISK INITIALIZATION
ENTER DISK INITIALIZATION OPTION: _
```

2. Enter a 1 to initialize a diskette

```
ENTER DISK INITIALIZATION OPTION: 1
DISKETTE INITIALIZATION
*****
*          DISKETTE FORMATTING PROGRAM          *
* IF FORMATTING IS IN PROGRESS, DO NOT         *
* CANCEL ($C) THIS PROGRAM INSTEAD, PRESS     *
* ATTN AND ENTER $DASDI TO TERMINATE.        *
*****
ENTER DISKETTE ADDRESS IN HEX: _
```

Storing and Handling Information

Initializing Diskettes (*continued*)

3. Enter the hexadecimal address of the diskette device. Our example uses device address 02.

```
ENTER DISKETTE ADDRESS IN HEX: 02
INITIALIZE FOR USAGE WITH THE IBM EVENT DRIVEN EXECUTIVE? _
```

Note: If you enter an N, \$DASDI displays the option menu shown in step 2 on page OP-93.

4. Enter a Y to initialize the diskette. If you are using a 4964 Diskette Drive, skip to step 6 on page OP-95.

```
INITIALIZE FOR USAGE WITH THE EVENT DRIVEN EXECUTIVE? Y _
SELECT DENSITY (1-SINGLE, 2-DOUBLE): _
```

If you are not initializing the diskette to be used with the Event Driven Executive, enter an N and continue with step 5 on page OP-99 under "Initialize to IBM Standard for Information Interchange" on page OP-97, or step 4 on page OP-103 under "Initialize to Bytes Per Sector Format" on page OP-102.

5. Enter a 1 for single density, a 2 for double density. To format for double density, the diskette must be a 2D diskette.

```
SELECT DENSITY (1-SINGLE, 2-DOUBLE): 1
SELECT SECTOR SIZE (1-128, 2-256): _
```

Initializing Diskettes (*continued*)

6. Enter the number corresponding to the sector size desired. \$DASDI only presents the options that are compatible with the density selected in step 5 on page OP-94.

```
SELECT SECTOR SIZE (1-128, 2-256): 2
DEVICE VARIED OFFLINE

DEVICE ADDRESS 02, EDX FORMAT, SINGLE DENSITY, 256 BYTES/SECTOR
WARNING:

FORMATTING WILL DESTROY ALL DATA ON THE DISKETTE. CONTINUE? _
```

7. Do you have the correct diskette in the diskette unit? (\$DASDI asks you to verify that you have the correct diskette in the diskette unit because any data on the diskette will be deleted.)

- If so, enter a Y.
- If not, insert the correct diskette and enter a Y. (If you do not have a diskette available, enter an N and skip to step 10 on page OP-96 to end the utility.)

```
DEVICE ADDRESS 02, EDX FORMAT, SINGLE DENSITY, 256 BYTES/SECTOR
WARNING:

FORMATTING WILL DESTROY ALL DATA ON THE DISKETTE. CONTINUE? Y
IBMEDX VARIED ONLINE

FORMATTING COMPLETE

LOAD $INITDSK? _
```

After completing this step, the diskette volume label is IBMEDX.

8. Do you want to create a volume directory on the diskette? (You can initialize the directory now or at a later time.)

- If so, enter a Y and continue with the procedure "Initialize a Diskette Volume Directory" on page OP-106.
- If not, enter an N and continue with step 9 on page OP-96 .

```
LOAD $INITDSK? N
ANOTHER DISKETTE? _
```

Storing and Handling Information

Initializing Diskettes (*continued*)

9. Do you want to initialize another diskette?
- If so, enter a Y and continue with step 10.
 - If not, enter an N and continue with step 11 on page OP-97 .

```
ANOTHER DISKETTE?  Y
SAME DEVICE ADDRESS AND FORMAT?  _
ANOTHER DISKETTE?  N
DIRECT ACCESS DEVICE INITIALIZATION
.
.
ENTER DISK INITIALIZATION OPTION:  _
```

10. Are you using the same device and do you want the diskette initialized to EDX format?

- If so, insert another diskette into the device, enter a Y, and continue with step 7 on page OP-95.
- If not, insert the diskette into the device you are using, enter an N and continue with step 3 on page OP-94.

```
SAME DEVICE ADDRESS AND FORMAT?  Y
FORMATTING WILL DESTROY ALL DATA ON THE DISKETTE. CONTINUE?
```

```
ANOTHER DISKETTE?  Y
SAME DEVICE ADDRESS AND FORMAT?  N
ENTER DISKETTE ADDRESS IN HEX:  _
```

Initializing Diskettes (*continued*)

11. Enter a 9 to end \$DASDI.

```
DIRECT ACCESS DEVICE INITIALIZATION
DISK INITIALIZATION OPTIONS:
0=CREATE STAND-ALONE DUMP/$TRAP DISKETTE
1=DISKETTE INITIALIZATION
2=4962 DISK INITIALIZATION
3=4963 DISK INITIALIZATION
4=4967 DISK INITIALIZATION
5=DDSK DISK INITIALIZATION
9=EXIT DISK INITIALIZATION
ENTER DISK INITIALIZATION OPTION: 9

$DASDI ENDED 08:05:30
```

Initialize to IBM Standard for Information Interchange

Use this procedure to initialize a diskette to the IBM Standard for Information Interchange. This format is used for basic exchange and H-exchange diskettes when transferring information between an EDX system and another operating system.

Before You Start:

Observe the following requirements:

- Use a Diskette 1 if you will be using the diskette for basic exchange data sets.
- Use a Diskette 2D if you will be using the diskette for H-exchange data sets.
- Insert the diskette into the diskette device. If you are using a 4966 Magazine unit, the diskette must be in slot number 1.
- Get the hexadecimal address of the diskette unit. (See "Listing the Devices on Your System" on page OP-318 for the procedure for obtaining this number.)
- Select the sector size. The sector size can be 128, 256, 512, or 1024 bytes. For basic exchange diskettes it must be 128; for H-exchange diskettes it must be 256. (See Figure 11 on page OP-92 for the valid sector sizes and diskette combinations.)

Storing and Handling Information

Initializing Diskettes (*continued*)

1. Load the utility \$DASDI:

- Using the \$L operator command, press the attention key and enter \$L \$DASDI.
- Using the session manager, select option 3.6.

```
> $L $DASDI
LOADING $DASDI      7P,08:30:06, LP= 0000, PART=2
DIRECT ACCESS DEVICE INITIALIZATION
DISK INITIALIZATION OPTIONS:
  0=CREATE STAND-ALONE DUMP/$TRAP DISKETTE
  1=DISKETTE INITIALIZATION
  2=4962 DISK INITIALIZATION
  3=4963 DISK INITIALIZATION
  4=4967 DISK INITIALIZATION
  5=DDSK DISK INITIALIZATION
  9=EXIT DISK INITIALIZATION
ENTER DISK INITIALIZATION OPTION: _
```

2. Enter a 1 to initialize a diskette.

```
ENTER DISK INITIALIZATION OPTION: 1
DISKETTE INITIALIZATION
*****
*      DISKETTE FORMATTING PROGRAM      *
* IF FORMATTING IS IN PROGRESS, DO NOT *
* CANCEL ($C) THIS PROGRAM INSTEAD PRESS *
* ATTN AND ENTER $DASDI TO TERMINATE.  *
*****
ENTER DISKETTE ADDRESS IN HEX: _
```

3. Enter the hexadecimal address of the diskette device. (If you enter an N, \$DASDI displays the option menu shown in step 2.)

```
ENTER DISKETTE ADDRESS IN HEX: 02
INITIALIZE FOR USAGE WITH THE IBM EVENT DRIVEN EXECUTIVE? _
```

4. Enter an N.

```
INITIALIZE FOR USAGE WITH THE EVENT DRIVEN EXECUTIVE? N
INITIALIZE TO IBM STANDARDS FOR INFORMATION INTERCHANGE? _
```

Initializing Diskettes (*continued*)

5. Enter a **Y**. If you enter an **N**, \$DASDI initializes the diskette to bytes-per-sector format.

Skip to step 7 if you are using a 4964 Diskette Drive.

```
INITIALIZE TO IBM STANDARDS FOR INFORMATION INTERCHANGE? Y
```

```
NOTE: EDX SUPPORT FOR IBM STANDARDS FOR INFORMATION INTERCHANGE  
DISKETTES IS LIMITED TO THE DOCUMENTED EDX EXCHANGE UTILITIES  
SELECT DENSITY (1=SINGLE, 2=DOUBLE): _
```

6. Enter the number of the required density:

- Enter a **1** for a diskette that will be used for basic exchange data sets.
- Enter a **2** for a diskette that will be used for H-exchange data sets.

```
SELECT DENSITY (1=SINGLE, 2=DOUBLE): 1  
SELECT SECTOR SIZE (1-128, 2-256, 3-512): _
```

7. Enter the number corresponding to the required sector size. \$DASDI presents the sector sizes that are compatible with the diskette type and density you are using.

- For diskettes that will be used for basic exchange data sets, select a sector size of 128 bytes-per-sector.
- For diskettes that will be used for H-exchange data sets, select a sector size of 256 bytes-per-sector.

```
SELECT SECTOR SIZE (1-128, 2-256, 3-512): 1  
DEVICE VARIED OFFLINE
```

```
DEVICE ADDRESS 02, IBM STANDARD FORMAT, SINGLE DENSITY, 128  
BYTES/SECTOR
```

```
WARNING:  
FORMATTING WILL DESTROY ALL DATA ON THE DISKETTE, CONTINUE? _
```

Storing and Handling Information

Initializing Diskettes (*continued*)

8. Do you have the correct diskette in the diskette unit? (\$DASDI asks you to verify that you have the correct diskette in the diskette unit because any data on the diskette will be deleted.)

- If so, enter a **Y**.
- If not, insert the correct diskette and enter a **Y**. (If you do not have a diskette available, enter an **N** and skip to step 9 to end the utility.)

```
FORMATTING WILL DESTROY ALL DATA ON THE DISKETTE. CONTINUE? Y
IBMEX VARIED ONLINE
FORMATTING COMPLETE
ANOTHER DISKETTE? _
```

9. Do you want to initialize another diskette?

- If so, enter a **Y** and continue with step 10 on page OP-101.
- If not, enter an **N** and continue with step 11 on page OP-101 .

```
ANOTHER DISKETTE? Y
SAME DEVICE ADDRESS AND FORMAT? _
```

```
ANOTHER DISKETTE? N
DIRECT ACCESS DEVICE INITIALIZATION
.
.
ENTER DISK INITIALIZATION OPTION: _
```

Initializing Diskettes (*continued*)

10. Are you using the same device and do you want the diskette initialized to IBM Standard for Information Interchange?

- If so, insert another diskette into the device, enter a **Y**, and continue with step 8 on page OP-100.

```
SAME DEVICE ADDRESS AND FORMAT? Y
FORMATTING WILL DESTROY ALL DATA ON THE DISKETTE. CONTINUE?
```

- If not, insert a diskette into the device you are using, enter an **N** and continue with step 3 on page OP-98.

```
SAME DEVICE ADDRESS AND FORMAT? N
ENTER DISKETTE ADDRESS IN HEX: _
```

11. Enter a **9** to end \$DASDI.

```
DIRECT ACCESS DEVICE INITIALIZATION
DISK INITIALIZATION OPTIONS:
0=CREATE STAND-ALONE DUMP/$TRAP DISKETTE
1=DISKETTE INITIALIZATION
2=4962 DISK INITIALIZATION
3=4963 DISK INITIALIZATION
4=4967 DISK INITIALIZATION
5=DDSK DISK INITIALIZATION
9=EXIT DISK INITIALIZATION
ENTER DISK INITIALIZATION OPTION: 9
```

```
$DASDI ENDED 08:05:30
```

Storing and Handling Information

Initializing Diskettes (*continued*)

Initialize to Bytes Per Sector Format

Use this procedure to initialize an entire diskette to 128, 256, 512, or 1024 bytes-per-sector.

Before You Start:

Observe the following requirements:

- Insert the diskette to be initialized into the diskette device. If you are using a 4966 Magazine unit, the diskette must be in slot number 1.
- Get the hexadecimal address of the diskette unit. (See "Listing the Devices on Your System" on page OP-318 for the procedure for obtaining this number.)
- Determine the sector size. (See Figure 11 on page OP-92 for valid sector sizes for the type of diskette you are initializing.)

1. Load the utility \$DASDI:

- Using the \$L operator command, press the attention key and enter \$L \$DASDI.
- Using the session manager, select option 3.6.

```
> $L $DASDI
LOADING $DASDI      7P,08:30:06, LP= 0000, PART=2
DIRECT ACCESS DEVICE INITIALIZATION
DISK INITIALIZATION OPTIONS:
0=CREATE STAND-ALONE DUMP/$STRAP DISKETTE
1=DISKETTE INITIALIZATION
2=4962 DISK INITIALIZATION
3=4963 DISK INITIALIZATION
4=4967 DISK INITIALIZATION
5=DDSK DISK INITIALIZATION
9=EXIT DISK INITIALIZATION
ENTER DISK INITIALIZATION OPTION: _
```

2. Enter a 1 to initialize a diskette.

```
ENTER DISK INITIALIZATION OPTION: 1
DISKETTE INITIALIZATION
*****
*          DISKETTE FORMATTING PROGRAM          *
* IF FORMATTING IS IN PROGRESS, DO NOT          *
* CANCEL ($C) THIS PROGRAM INSTEAD, PRESS      *
* ATTN AND ENTER $DASDI TO TERMINATE.          *
*****
ENTER DISKETTE ADDRESS IN HEX: _
```

Initializing Diskettes (*continued*)

3. Enter the hexadecimal address of the diskette device. (If you enter an N, \$DASDI displays the option menu shown in step 2 on page OP-102.)

```
ENTER DISKETTE ADDRESS IN HEX: 02
INITIALIZE FOR USAGE WITH THE EVENT DRIVEN EXECUTIVE? _
```

4. Enter an N.

```
INITIALIZE FOR USAGE WITH THE EVENT DRIVEN EXECUTIVE? N
INITIALIZE TO STANDARDS FOR INFORMATION INTERCHANGE? _
```

5. Enter an N.

Skip to step 7 if you are using a 4964 Diskette Drive.

```
INITIALIZE TO STANDARDS FOR INFORMATION INTERCHANGE? N
SELECT DENSITY (1-SINGLE, 2-DOUBLE): _
```

6. Enter the number corresponding to the density you want.

```
SELECT DENSITY (1-SINGLE, 2-DOUBLE): 1
SELECT SECTOR SIZE (1-128, 2-256, 3-512, 4-1024): _
```

7. Enter the number corresponding to the desired sector size.

```
SELECT SECTOR SIZE (1-128, 2-256, 3-512, 4-1024): 2
DEVICE VARIED OFFLINE

DEVICE ADDRESS IS 02, NON-STANDARD, SINGLE DENSITY, 256 BYTE...

WARNING:
FORMATting WILL DESTROY ALL DATA ON THE DISKETTE. CONTINUE? _
```

Storing and Handling Information

Initializing Diskettes (*continued*)

8. Do you have the correct diskette in the diskette unit? (\$DASDI asks you to verify that you have the correct diskette in the diskette unit because any data on the diskette will be deleted.)

- If so, enter a Y.
- If not, insert the correct diskette and enter a Y. (If you do not have a diskette available, enter an N and skip to step 9 to end the utility.)

```
FORMATTING WILL DESTROY ALL DATA ON THE DISKETTE. CONTINUE? Y
IBMEDX VARIED ONLINE

FORMATTING COMPLETE

ANOTHER DISKETTE? _
```

9. Do you want to format another diskette?

- If so, enter a Y and continue with step 10 on page OP-105.
- If not, enter an N and continue with step 11 on page OP-105.

```
ANOTHER DISKETTE? Y
SAME DEVICE ADDRESS AND FORMAT? _
```

```
ANOTHER DISKETTE? N
DIRECT ACCESS DEVICE INITIALIZATION
.
.
ENTER DISK INITIALIZATION OPTION: _
```

Initializing Diskettes (*continued*)

10. Are you using the same device and do you want the diskette initialized to bytes-per-sector format?

- If so, insert another diskette into the device, enter a **Y**, and continue with step 8 on page OP-104.

```
SAME DEVICE ADDRESS AND FORMAT? Y
FORMATTING WILL DESTROY ALL DATA ON THE DISKETTE. CONTINUE?
```

- If not, insert a diskette into the device you are using, enter an **N** and continue with step 3 on page OP-103.

```
SAME DEVICE ADDRESS AND FORMAT? N
ENTER DISKETTE ADDRESS IN HEX: _
```

11. Enter a **9** to end \$DASDI.

```
DIRECT ACCESS DEVICE INITIALIZATION
DISK INITIALIZATION OPTIONS:
0=CREATE STAND-ALONE DUMP/$TRAP DISKETTE
1=DISKETTE INITIALIZATION
2=4962 DISK INITIALIZATION
3=4963 DISK INITIALIZATION
4=4967 DISK INITIALIZATION
5=DDSK DISK INITIALIZATION
9=EXIT DISK INITIALIZATION
ENTER DISK INITIALIZATION OPTION: 9
```

```
$DASDI ENDED 08:05:30
```

Storing and Handling Information

Initializing Diskettes (*continued*)

Initialize a Diskette Volume Directory

Use this procedure to initialize a volume directory on a diskette. You can also allocate logical volumes (step 9), their data set directories (step 12), \$EDXNUC (step 15), and IPL text (step 16) at the same time.

Before You Start:

Observe the following requirements:

- The diskette must be initialized for EDX data sets before the volume directory can be initialized. If the diskette has not been initialized, use the procedure "Initialize for EDX Format" on page OP-93.
- The hexadecimal address of the diskette unit. See "Listing the Devices on Your System" on page OP-318 for the procedure for obtaining this number.
- The number of the slot the diskette is in if the diskette is in a 4966 Magazine Unit. (The diskette cannot be in a magazine.)
- The 1 to 6 character name you are assigning to the diskette volume label.
- The owner identification. This can be either a name, initials, department number, or some other identification of who owns the diskette.
- The 1 to 6 character name of each logical volume and its size.
- If you are entering this procedure from the procedure "Initialize for EDX Format" on page OP-93 skip to step3.

1. Load the utility \$INITDSK:

- Using the \$L operator command, press the attention key and enter \$L \$INITDSK.
- Using the session manager, select option 3.7.

```
> $L $INITDSK
LOADING $INITDSK      83P,08:01:20, LP= 0000,...
$INITDSK - DISK INITIALIZATION UTILITY
COMMAND(?): _
```

2. Enter ID to initialize the diskette.

```
COMMAND(?): ? ID
DEVICE ADDRESS: _
```

3. Enter the hexadecimal address of the diskette device containing the diskette being initialized.

```
DEVICE ADDRESS: 02
SLOT NUMBER: _
```

Initializing Diskettes (*continued*)

4. If the diskette is in a 4966 diskette magazine unit, enter the number of the slot containing the diskette.

If the diskette is not in a 4966, skip to step 5.

```
SLOT NUMBER: 02
INITIALIZE DEVICE MAY DESTROY ALL DATA
CONTINUE (Y/N)? _
```

5. \$INITDSK verifies that you want to initialize this particular diskette. If the diskette is already initialized, \$INITDSK tells you so as part of the verification message. You can reinitialize a diskette, but all data currently recorded on the diskette will be lost.

- If you want to continue, enter a Y.
- If you do not want to initialize the current diskette, enter an N and continue with step 18 on page OP-110 for another diskette, or step 3 on page OP-106 to end \$INITDSK.

```
INITIALIZE DEVICE MAY DESTROY ALL DATA
CONTINUE (Y/N)? Y
INITIALIZE DISKETTE AS MULTI-VOLUME TYPE? _
```

6. Do you want more than one volume on this diskette?

- If so, enter a Y, and continue with step 7.
- If not, enter an N, and continue with step 7.

```
INITIALIZE DISKETTE AS MULTI-VOLUME TYPE? Y
NEW VOLUME LABEL: _
```

7. Enter the 1 to 6 character volume label for the diskette.

```
NEW VOLUME LABEL: MYDKT1
ENTER OWNER IDENTIFICATION: _
```

Storing and Handling Information

Initializing Diskettes (*continued*)

8. Enter an identification about who owns the diskette contents. If this is a single-volume diskette (you entered an N in step 6 on page OP-107), skip to step 14 on page OP-109.

```
ENTER OWNER IDENTIFICATION: DEPT 103
MULTI-VOLUME TYPE DISKETTE INITIALIZED
ALLOCATE A VOLUME (Y/N)? _
```

9. Do you want to allocate one or more logical volumes at this time? (You can allocate volumes at a later time using the procedure "Allocate a Volume" on page OP-121.)

```
ALLOCATE A VOLUME (Y/N)? Y
VOLUME: _
```

- If so, enter a Y.
- If not, enter an N. Continue with step 3 on page OP-106 to initialize another diskette, or with step 18 on page OP-110 to end \$INITDSK.

10. Enter the name you are assigning to the logical volume. The name can be up to 8 alphanumeric characters in length.

```
VOLUME: VOLUME1
ENTER SIZE IN RECORDS: _
```

11. Enter the number of records to be allocated for the volume.

```
ENTER SIZE IN RECORDS: 500
VOLUME ALLOCATED
INITIALIZE THE VOLUME JUST ALLOCATED (Y/N)? _
```

Initializing Diskettes (*continued*)

- 12.** Do you want to initialize the data set directory on the volume? (If you do not do so now, you will have to use the \$INITDSK IV command before you can allocate data sets on the volume. See *Operator Commands and Utilities Reference* for a description of the \$INITDSK IV command.)

- If so, enter a Y.
- If not, enter an N. Continue with step 17 on page OP-110.

```
INITIALIZE THE VOLUME JUST ALLOCATED (Y/N)? Y
MAXIMUM NUMBER OF DATASETS? _
```

- 13.** Enter the maximum number of data sets that you want to be able to allocate on this logical volume.

```
MAXIMUM NUMBER OF DATASETS? 10
DO YOU WANT WRITE VERIFY FOR THIS VOLUME? _
```

- 14.** Do you want EDX to verify that it can read all data it writes in the data sets allocated on the volume?

- If so, enter a Y.
- If not, enter an N.

```
DO YOU WANT WRITE VERIFY FOR THIS VOLUME? N
ALLOCATE $EDXNUC (Y/N)? _
```

- 15.** If the volume is large enough to contain the system nucleus, \$INITDSK asks you if you want to allocate space for it. If you do not receive this message, continue with step 17 on page OP-110.

- If you want \$EDXNUC allocated, enter a Y.
- If you do not want \$EDXNUC allocated, enter an N and continue with step 17 on page OP-110.

```
ALLOCATE $EDXNUC (Y/N)? Y
VOLUME INITIALIZED
INITIALIZE IPL TEXT (Y/N)? _
```

Storing and Handling Information

Initializing Diskettes (*continued*)

16. Do you want to be able to IPL the system from this diskette?

- If so, enter a **Y**.
- If not, enter an **N**.

If this is a single-volume diskette, continue with step 18.

```
INITIALIZE IPL TEXT (Y/N)? Y
IPL TEXT WRITTEN
ALLOCATE ANOTHER VOLUME (Y/N)? _
```

17. Do you want to allocate another logical volume?

- If so, enter a **Y** and continue with step 10 on page OP-108.
- If not, enter an **N**.

```
ALLOCATE ANOTHER VOLUME (Y/N)? N
COMMAND (?): _
```

18. Do you want to initialize another diskette volume directory?

- If so, insert the diskette into a diskette drive and continue with step 2 on page OP-106.
- If not, enter **EN** to end \$INITDSK.

If you came to this procedure from the procedure "Initialize for EDX Format" on page OP-93, continue with step 9 on page OP-96.

```
COMMAND (?): EN
$INITDSK ENDED AT 08:30:01
```

Initializing Diskettes (*continued*)

Initialize an H Exchange Volume

Use this procedure to initialize a volume on an H-exchange diskette.

Before You Start:

Use a 2D diskette that is initialized to IBM Standards for Information Interchange at 256 bytes-per-sector, as shown under "Initialize to IBM Standard for Information Interchange" on page OP-97.

Note: H-exchange diskettes can only be read on 4965 and 4966 diskette units.

1. Load the utility \$HXUT1:

- Using the \$L operator command, press the attention key and enter \$L \$HXUT1.
- Using the session manager, select option 3.11.

```
> $L $HXUT1
LOADING $HXUT1      71P,08:04:28, LP= 2300

$HXUT1 - H EXCHANGE UTILITY

VOLUME NAME =
```

2. Enter the volume label of the diskette you are using. This label was assigned when the diskette was initialized.

```
VOLUME NAME = IBMEDX

COMMAND (?): _
```

3. Enter IV to initialize the H-exchange volume.

```
COMMAND (?): IV

*** WARNING ***
INITIALIZATION WILL DELETE ALL EXISTING DATASETS --
CONTINUE (Y/N) _
```

Storing and Handling Information

Initializing Diskettes (*continued*)

4. If this is not a new diskette, all existing data sets will be deleted: do you want to do this?

- If so, enter a Y.
- If not, enter an N and continue with step 3 on page OP-111 to start over.

```
CONTINUE (Y/N) Y
INITIALIZATION COMPLETE
COMMAND (?): _
```

5. Do you want to initialize another diskette volume?

- If so, insert another diskette and continue with step 3 on page OP-111.
- If not, enter EN to end \$HXUT1.

```
COMMAND (?): EN
$HXUT1 ENDED 08:30:24
```

Initializing Tapes

Use this procedure to initialize a new tape or to change the label information on a used tape. This procedure can be used to initialize tapes for non-labeled, standard label, or bypass label processing.

Before You Start:

Obtain the following information:

- The ID (TAPEID) of the tape drive you are using.
- The label type - standard, non-label, or bypass label - for the tape drive you are using.
- The density for the tape drive you are using. (It is set during system generation.) A 4969 may be set for 800 or 1600 BPI or DUAL density, and a 4968 may be set for 1600 or 3200 or DUAL density. Note that 3200 BPI is not compatible with the ANSI standard, but is compatible with other 4968 tape units.
- The 1-6 character name of the volume you are initializing.
- The 1-8 character name of the first data set.
- The 1-10 character owner identification.
- The Julian expiration date for the data set. A Julian date has the format of YYDDD, where YY is the year and DDD is the day in the form 1-365.

Initializing Tapes (*continued*)

1. Load the utility \$TAPEUT1:

- Using the \$L operator command, press the attention key and enter \$L \$TAPEUT1.
- Using the session manager, select option 3.10.

After it is loaded, \$TAPEUT1 displays the following information about the tapes defined on your system:

- The tape drive ID (TAPEID).
- The word DUAL if the tape is set for dual density.
- The label type (SL=standard label, NL=non-labeled, BLP= bypass label processing).
- Current density setting:
 - 4968 = 1600 or 3200.
 - 4969 = 800 or 1600.
- The volume name, if the tape is an online standard label tape.
- The tape speed, if the tape device is a 4968.
- Whether the tape is online or offline.
- The number of the file where the tape is currently positioned.
- The tape device address.

```
> $L $TAPEUT1
LOADING $TAPEUT1    21P,08:01:20, LP= 0000,...
TAPE01 DUAL NL 1600 OFFLINE
      DEVICE ADDRESS = 004D
TAPE02 DUAL NL 1600
TAPESPEED 25 IPS
OFFLINE DEVICE ADDRESS = 0006

COMMAND(?): _
```

2. Enter IT to initialize the tape.

```
COMMAND(?): IT
ENTER TAPEID (1-6 CHARS): _
```

Storing and Handling Information

Initializing Tapes (*continued*)

3. Enter the 1-6 character ID of the tape drive.

If the tape drive is not offline, \$TAPEUT1 displays a message asking if you want to continue. If you receive this question, continue with step 3(a).

If you are reinitializing a used tape that has an unexpired data set, \$TAPEUT1 warns you that the data set has not expired. If you receive the warning message, continue with step 3(b).

If you do not receive either warning message, continue with step 4.

```
ENTER TAPEID (1-6 CHARS): TAPE01
STANDARD LABEL 1600 BPI (Y/N)? _
```

3(a). Do you want \$TAPEUT1 to vary the tape drive offline and continue?

- If so, enter a Y.
- If not, enter an N and continue with step 2 to try another tape, or step 8 to end \$TAPEUT1.

```
WARNING! DEVICE ISN'T OFFLINE.
CONTINUE? Y
```

```
STANDARD LABEL 1600 BPI (Y/N)? _
```

3(b). Do you want \$TAPEUT1 to override the unexpired date?

- If so, enter a Y.
- If not, enter an N and continue with step 2 to try another tape, or step 8 to end \$TAPEUT1.

```
UNEXPIRED DATA SET!
CONTINUE? Y
```

```
STANDARD LABEL 1600 BPI (Y/N)? _
```

Initializing Tapes (*continued*)

4. \$TAPEUT1 asks if you want to initialize the tape to the same label type and density specified for the tape drive.

- If you do, enter a **Y** and continue with step 6.
- If you do not, enter an **N** and continue with step 5.

```
NO LABEL 1600 BPI (Y/N)? N
CHANGE (Y/N)? _
```

5. \$TAPEUT1 verifies whether you want to change the label type or density. You can change the density only if the tape drive is set for DUAL density. These changes are in effect only until you IPL the Series/1.

- If you want to change the label type and/or density, enter a **Y** and continue with step 5(a).
- If you do not want to change the label type and/or density, enter an **N** and continue with step 6.

```
CHANGE (Y/N)? Y
LABEL (SL,NL): _
```

5(a). Enter the abbreviation for the desired label type:

- Enter **NL** for no label processing.
- Enter **SL** for standard label processing. A tape must have a standard label to have a volume.

```
LABEL (SL,NL): SL
DENSITY (800,1600): _
```

5(b). Enter the desired density:

- Enter **800** for 800 BPI on a 4969.
- Enter **1600** for 1600 BPI on a 4968 or 4969.
- Enter **3200** for 3200 BPI on a 4968.

```
DENSITY (800,1600): 800
STANDARD LABEL 800 BPI (Y/N)? _
```

Storing and Handling Information

Initializing Tapes (*continued*)

5(c). Have you specified the desired parameters?

- If you have, enter a **Y** and continue with step 6.
- If you have not, enter an **N** and continue with step 5.

```
STANDARD LABEL 800 BPI (Y/N)? Y
TAPEDS (NAME,VOLUME): _
```

6. Enter the data set and volume name for the first data set and volume to be allocated.

```
TAPEDS (NAME,VOLUME): MYDATA1,MYVOL
OWNERID (1-10 CHARS): _
```

7. Enter a 1-10 character owner ID.

```
OWNERID (1-10 CHARS): MYNAME
EXPIRATION DATE (YYDDD): _
```

8. Enter the Julian expiration date for the data set. The format of this date is YYDDD, where YY is the year and DDD is the day.

If the data set currently exists on the tape and the date has not expired, \$TAPEUT1 displays a message asking if you want to continue. If this message is displayed, continue with step 8(a), otherwise continue with step 9.

```
EXPIRATION DATE (YYDDD): 82103
TAPE INITIALIZED
MYVOL ONLINE

COMMAND (?): _
```

8(a). Do you want to change the label information before the expiration date?

- If so, enter a **Y**.
- If not, enter an **N**.

```
UNEXPIRED DATA SET!
CONTINUE? Y
TAPE INITIALIZED
MYVOL ONLINE

COMMAND (?): _
```

Initializing Tapes *(continued)*

9. Do you want to initialize another tape?

- If so, continue with step 2.
- If not, enter EN to end \$TAPEUT1.

Note: Record the information about the tape (density, label type, volume and data set name, and file number of the data set) on an external label attached to the reel of tape and/or on a tape log. You cannot get a directory listing of this information like you can for a disk or diskette.

```
COMMAND(?): EN
```

```
$TAPEUT1 ENDED AT 01:02:00
```

Storing and Handling Information

Changing Tape Drive Attributes

Use this procedure to change the label type, density, or speed attributes on a tape drive.

Before You Start

Obtain the following information:

- The ID (TAPEID) of the tape drive you are using.
- The hexadecimal device address of the tape drive if it is not offline.

1. Enter `$VARYOFF ioda`, where *ioda* is the hexadecimal address of the tape drive, to vary the tape drive offline. If the tape drive is already offline, skip to step 2.

```
> $VARYOFF 6  
TAPE02 OFFLINE
```

2. Load the utility `$TAPEUT1`:

- Using the `$L` operator command, press the attention key and enter `$L $TAPEUT1`.
- Using the session manager, select option 3.10.

```
> $L $TAPEUT1  
LOADING $TAPEUT1      21P,08:01:20, LP= 0000,...  
TAPE01 DUAL NL 1600 OFFLINE  
      DEVICE ADDRESS = 004D  
TAPE02 DUAL NL 1600  
TAPESPEED 25 IPS  
OFFLINE DEVICE ADDRESS = 0006  
  
COMMAND(?): _
```

After it is loaded, `$TAPEUT1` displays the following information about the tapes defined on your system:

- The tape drive ID (TAPEID).
- The word DUAL if the tape is set for dual density.
- The label type (SL=standard label, NL=non-labeled, BLP= bypass label processing).
- Current density setting:
 - 4968 = 1600 or 3200.
 - 4969 = 800 or 1600.
- The volume name, if the tape is an online standard label tape.
- The tape speed, if the tape device is a 4968.
- Whether the tape is online or offline.
- The number of the file where the tape is currently positioned.
- The tape device address.

Changing Tape Drive Attributes (*continued*)

3. Enter **CT** to change the tape drive attributes.

```
COMMAND(?): CT
```

```
ENTER TAPEID (1-6 CHARS): _
```

4. Enter the 1-6 character ID of the tape drive.

```
ENTER TAPEID (1-6 CHARS): TAPE02
TAPE TAPE02 AT ADDRESS 0006 IS NO 1600BPI
TAPE SPEED 25 IPS
DO YOU WISH TO MODIFY (Y/N)? _
```

- 5 Do you want to continue and modify the tape drive attributes?

- If so, enter a **Y**.
- If not, enter an **N** and continue with step 8 to end \$TAPEUT1.

```
DO YOU WISH TO MODIFY (Y/N)? Y
LABEL TYPE (NULL, SL, NL, BLP)? _
```

6. Enter the desired label type:

- Leave blank (**NULL**) to keep the same label type.
- **SL** for standard label processing.
- **NL** for no label processing.
- **BLP** for bypass label processing.

```
LABEL TYPE (NULL, SL, NL, BLP)? SL
DENSITY (NULL,1600,3200)? _
```

Storing and Handling Information

Changing Tape Drive Attributes (*continued*)

7. Enter the desired density (bytes per inch):

- Leave blank (NULL) to keep the current density.
- **800** for 800 BPI on a 4969.
- **1600** for 1600 BPI on a 4968 or 4969.
- **3200** for 3200 BPI on a 4969.

```
DENSITY (NULL,1600,3200)? 1600  
IPS (25,100)? _
```

8. Do you want to change the attributes on another tape drive?

- If so, continue with step 1.
- If not, enter **EN** to end \$TAPEUT1.

```
COMMAND(?): EN  
$TAPEUT1 ENDED AT 01:02:00
```

Allocating Volumes and Data Sets

Use the procedures in this section to allocate volumes and data sets on your disks and diskettes.

Allocate a Volume

Use this procedure to allocate a volume on a disk or multivolume diskette. A volume can contain from three to the maximum number of records allowable on the device.

This procedure also initializes a data set directory on the volume (step 6 on page OP-123), allocates \$EDXNUC (step 9 on page OP-123), and initializes IPL text (step 10 on page OP-123).

This procedure will not allocate a volume under the fixed-head area of a disk or a volume on an H-exchange diskette. See the AF command of \$INITDSK in the *Operator Commands and Utilities Reference* to allocate a volume under the fixed-head area of a disk and "Initialize an H Exchange Volume" on page OP-111 to allocate a volume on an H-exchange diskette.

Before You Start:

Observe the following requirements:

- The diskette must be initialized to be used with EDX format data sets. See "Initialize for EDX Format" on page OP-93 for the procedure.
- The diskette must have a volume directory. You can determine if the diskette has a volume directory by trying to list it as shown under "List All Volumes on One Device" on page OP-183. See "Initialize a Diskette Volume Directory" on page OP-106 for the procedure to initialize the volume directory.
- Get the hexadecimal address of the diskette device. If you do not know what it is, see "Listing the Devices on Your System" on page OP-318.
- Determine the 1-6 character name of the new volume.
- Determine the decimal number of records you want in the volume. A volume can be from three to the maximum number of records that can be contained on the disk or diskette. The following chart shows the maximum number of records that can be on a diskette:

Diskette type	Number of records at 128 BPS	Number of records at 256 BPS	Number of records at 1024 BPS
Diskette 1	946	1107	-
Diskette 2	1921	2217	-
Diskette 2-D	-	3845	4733

Storing and Handling Information

Allocating Volumes and Data Sets (*continued*)

- This procedure also gives you the option of initializing a data set directory for the volume you are allocating. A data set directory is required before you can allocate individual data sets on the volume. If you are initializing the data set directory at this time, you need the following information:
 - The maximum number of data sets you expect to allocate on the volume.
 - Whether you want write verify for the volume. If you specify that you want write verify, EDX re-reads all data that it writes on the data sets allocated on the volume, to make sure the data was written correctly.
 - If the volume is large enough to contain the system nucleus (data set name \$EDXNUC), \$INITDSK asks if you want \$EDXNUC allocated.
 - If you have \$EDXNUC allocated, \$INITDSK also asks if you want the IPL text initialized. Initializing the IPL text allows you to IPL the system from the diskette.

1. Load the utility \$INITDSK:

- Using the \$L operator command, press the attention key and enter **\$L \$INITDSK**.
- Using the session manager, select option 3.7.

```
> $L $INITDSK
LOADING $INITDSK 49P,08:04:28, LP= 0000,...
$INITDSK - DISK INITIALIZATION UTILITY
COMMAND (?): _
```

2. Enter AV to allocate a volume.

```
COMMAND (?): AV
DEVICE ADDRESS: _
```

3. Enter the hexadecimal address of the device you are using. (If you receive the message "INVALID DEVICE TYPE" or a return code of 13, you may have entered an incorrect device address, or you may need to initialize the volume directory.)

```
DEVICE ADDRESS: 48
VOLUME: _
```

4. Enter the 1-6 character name of the volume.

```
VOLUME: MYVOL
ENTER SIZE IN RECORDS: _
```

5. Enter the number of records. The number must be entered in decimal.

```
ENTER SIZE IN RECORDS: 1000
MYVOL ALLOCATED
INITIALIZE THE VOLUME JUST ALLOCATED (Y/N)? _
```

Allocating Volumes and Data Sets *(continued)*

6. Do you want to initialize a data set directory on this volume? A volume must have a data set directory initialized before you can allocate data sets on it. If you do not initialize it now, you will have to use the \$INITDSK IV command described in the *Operator Commands and Utilities Reference*.

```
INITIALIZE THE VOLUME JUST ALLOCATED? Y
MAXIMUM NUMBER OF DATA SETS: _
```

7. Enter the maximum number of data sets you expect to allocate on this volume.

```
MAXIMUM NUMBER OF DATA SETS: 5
DO YOU WANT WRITE VERIFY FOR THIS VOLUME? _
```

8. Do you want EDX to verify that it can read all data it writes on data sets allocated on the volume?

```
DO YOU WANT WRITE VERIFY FOR THIS VOLUME? N
ALLOCATE $EDXNUC (Y/N)?
```

- If so, enter a Y.
- If not, enter an N.

9. If the volume is large enough to contain the system nucleus, \$INITDSK asks you if you want to allocate it. If you do not receive this message, skip to step 11 on page OP-124

```
ALLOCATE $EDXNUC (Y/N)? Y
VOLUME INITIALIZED
INITIALIZE IPL TEXT (Y/N)? _
```

- If you want \$EDXNUC allocated, enter a Y.
- If you do not want \$EDXNUC allocated, enter an N and skip to step 11 on page OP-124.

10. Do you want to be able to IPL the system from this diskette?

```
INITIALIZE IPL TEXT? Y
IPL TEXT WRITTEN
COMMAND (?):
```

- If so, enter a Y.
- If not, enter an N.

Storing and Handling Information

Allocating Volumes and Data Sets *(continued)*

11. Do you want to allocate another volume?

- If so, continue with step 3 on page OP-122.
- If not, enter **EN** to end \$INITDSK.

```
COMMAND(?): EN
$INITDSK ENDED 08:30:24
```

Allocate a Data Set

Use this procedure to allocate a data set on a disk or diskette volume. After the data set is allocated, it contains whatever data is on its area of the disk or diskette until a program writes data into it.

Note: This procedure will not allocate a data set on an H-exchange data set. See “Allocate an H Exchange Data Set” on page OP-126 for that procedure.

Before You Start:

Observe the following requirements:

- If the volume is on a diskette, insert the diskette into a diskette drive.
- The volume must have a data set directory. (See “Allocate a Volume” on page OP-121 for instructions for allocating a data set directory on a volume.)
- Obtain the 1-6 character name of the volume on which you want the data set.
- Determine the 1-8 character name of the data set you are allocating.
- Determine the number of records you want allocated.
- Determine whether the data set is to contain data or a program.

1. Load the utility \$DISKUT1:

- Using the \$L operator command, press the attention key and enter **\$L \$DISKUT1**.
- Using the session manager, select option 3.1.

```
> $L $DISKUT1
LOADING $DISKUT1 49P,08:04:28, LP= 0000,...

$DISKUT1 - DATA SET MANAGEMENT UTILITY I

USING VOLUME EDX002

COMMAND (?): _
```

Allocating Volumes and Data Sets *(continued)*

2. Is \$DISKUT1 using the correct volume name?

- If so, continue with step 3.
- If not, change the name of the volume by typing **CV xxxxxx**, where xxxxxx is the name of the volume you want to use.

```
USING VOLUME EDX002
COMMAND (?): CV MYVOL
USING VOLUME MYVOL
COMMAND (?): _
```

3. Enter **AL** to allocate a data set.

```
COMMAND (?): AL
MEMBER NAME:
```

4. Enter the 1-8 character name of the data set you are allocating.

```
MEMBER NAME: NEWPROG
HOW MANY RECORDS? _
```

5. Enter the number of 256-byte records you are allocating.

```
HOW MANY RECORDS? 100
DEFAULT TYPE = DATA - OK? _
```

6. Is this data set going to contain data or a program?

- If data, enter a **Y** and continue with step 8 on page OP-126
- If a program, enter an **N** and continue with step 7.

```
DEFAULT TYPE = DATA - OK? N
TYPE = PROGRAM (Y/N)? _
```

7. Are you going to use the data set for a program?

- If so, enter a **Y**.
- If not, enter an **N** and no data set will be allocated.

```
TYPE = PROGRAM (Y/N)? Y
NEWPROG CREATED
COMMAND (?): _
```

Storing and Handling Information

Allocating Volumes and Data Sets (*continued*)

8. Do you want to allocate another data set?

- If so, continue at step 3 on page OP-125.
- If not, enter **EN** to end \$DISKUT1.

```
COMMAND (?): EN
$DISKUT1 ENDED 08:30:24
```

Allocate an H Exchange Data Set

Use this procedure to allocate a data set on an H-exchange diskette.

Note: H-exchange diskettes can only be read on 4965 and 4966 diskette units.

Before You Start:

Observe the following requirements:

- The diskette must be a Diskette 2D that is initialized for IBM Standards for Information Interchange and has a volume. (See “Initialize to IBM Standard for Information Interchange” on page OP-97 and “Initialize an H Exchange Volume” on page OP-111 for these procedures.)
- Obtain the volume label name of the diskette. (\$DASDI sets this to **IBMEDX** when you initialize the diskette. If you don’t know what the volume label name is, see “List All Volumes on One Device” on page OP-183 to determine what it is.)
- Obtain the name of the data set you are allocating.
- Determine the number of 256-byte records you are allocating.

1. Load the utility \$HXUT1:

- Using the \$L operator command, press the attention key and enter **\$L \$HXUT1**.
- Using the session manager, select option 3.11.

```
> $L $HXUT1
LOADING $HXUT1      71P,08:04:28, LP= 2300
$HXUT1 - H EXCHANGE UTILITY
VOLUME NAME =
```

2. Enter the volume label of the diskette you are using.

```
VOLUME NAME = IBMEDX
COMMAND (?): _
```

Allocating Volumes and Data Sets *(continued)*

3. Enter AL to allocate a data set.

```
COMMAND (?): AL
DATASET NAME: _
```

4. Enter the 1-8 character name of the data set.

```
DATASET NAME: MYHXDS
NUMBER OF RECORDS: _
```

5. Enter the number of 256-byte records you are allocating.

```
NUMBER OF RECORDS: 100
MYHXDS ALLOCATED, SIZE IS 100 RECORDS
COMMAND (?): _
```

6. Do you want to allocate another data set?

- If so, continue with step 4.
- If not, enter EN to end \$HXUT1.

```
COMMAND (?): EN
$HXUT1 ENDED 08:30:24
```

Allocate Tape Data Sets

Use this procedure to allocate a tape data set. This procedure uses \$TAPEUT1 to delete and reallocate an existing data set or to add a new data set after the last data set on a tape volume.

Before You Start:

Observe the following requirements:

- The tape must be initialized for stand label (SL) processing.
- Obtain the hexadecimal device address of the tape drive. (The device address is listed when you load \$TAPEUT1 or you can use the procedure "Listing the Devices on Your System" on page OP-318.)
- Obtain the name of the volume on the tape. This name is listed when you load \$TAPEUT1 or you can get it from the tape log or the external label on the tape reel.

Storing and Handling Information

Allocating Volumes and Data Sets (*continued*)

- If you are reallocating an existing data set, you need to know if the data set has an expiration date. If the date has not expired, you may not want to reallocate the data set.
- Obtain the file number of the data set being allocated. This number can be obtained from the physical label on the tape or from your tape log. (The first data set on the tape is file number 1, the second is file number 2, the *n*th data set is file number *n*.) When deciding where to put the data set, remember that all data following the new data set is deleted.
- Determine the 1-8 character name of the data set being allocated.
- Determine the new expiration date for the data set. The expiration date must be in Julian format YYDDD, where YY is the year and DDD is the day.

1. Enter `$VARYON ioda file EX` to vary the tape online to the file number of the data set being allocated or reallocated, where:

```
> $VARYON 4C 2 EX
IGNORE EXPIRATION DATE? (Y,N): _
```

- *ioda* is the hexadecimal device address of the tape drive.
- *file* is the decimal file number being accessed. In the example, we are accessing file number 2.
- *EX* specifies expiration date override. *EX* has to be entered only if you are replacing or reallocating a data set that has an unexpired expiration date.

You will receive the message "IGNORE EXPIRATION DATE?" only if you enter the EX parameter. If you do not enter EX, skip to step 3 on page OP-129.

2. If you entered the EX parameter, EDX verifies that you want to be able to override an unexpired date.

```
IGNORE EXPIRATION DATE? (Y,N): Y
MYVOL ONLINE
```

- If you want to be able to override an expiration date, enter a Y.
- If you do not want to be able to override an expiration date, enter an N.

Allocating Volumes and Data Sets (*continued*)

3. Load the utility \$TAPEUT1:

- Using the \$L operator command, press the attention key and enter **\$L \$TAPEUT1**.
- Using the session manager, select option 3.10.

After it is loaded, \$TAPEUT1 displays the following information about the tapes defined on your system:

- The tape drive ID (TAPEID).
- The word DUAL if the tape is set for dual density.
- The label type (SL=standard label, NL=non-labeled, BLP= bypass label processing).
- Current density setting:
 - 4968 = 1600 or 3200.
 - 4969 = 800 or 1600.
- The volume name, if the tape is an online standard label tape.
- The tape speed, if the tape device is a 4968.
- Whether the tape is online or offline.
- The number of the file where the tape is currently positioned.
- The tape device address.

```
> $L $TAPEUT1
$TAPEUT1 21P,08:01:20, LP= 0000, PART= 1
TAPE01 DUAL SL 1600 OFFLINE
        DEVICE ADDRESS = 004C
TAPE02 DUAL NL 1600
TAPESPEED 25 IPS
OFFLINE DEVICE ADDRESS = 0006

COMMAND(?): _
```

4. Enter TA to allocate a data set.

```
COMMAND(?): TA
TAPEDS (NAME,VOLUME): _
```

5. Enter the 1-8 character name of the data set and the name of the volume on the tape.

```
TAPEDS (NAME,VOLUME): MYDATA2,MYVOL
EXPIRATION DATE (YYDDD): _
```

Storing and Handling Information

Allocating Volumes and Data Sets *(continued)*

6. Enter the Julian expiration date for the data set.

```
EXPIRATION DATE (YYDDD): 83103
UNEXPIRED DATA SET!
CONTINUE? _
```

7. If you are reallocating a data set with an unexpired expiration date and you entered the EX parameter on the \$VARYON command, \$TAPEUT1 asks if you want to override the expiration date.

- Enter a Y to override the date and reallocate the data set.
- Enter an N to cancel the reallocation.

```
UNEXPIRED DATA SET!
CONTINUE? Y
DATA SET ALLOCATED

COMMAND (?) _
```

8. Do you want to allocate another data set?

- If so, continue with step 4 on page OP-129.
- If not, enter EN to end \$TAPEUT1.

```
COMMAND (?) EN
$TAPEUT1 ENDED AT 08:30:50
```

Renaming Volumes and Data Sets

Use the procedures in this section to rename disk or diskette volumes and data sets. To rename a tape data set, see "Allocate Tape Data Sets" on page OP-127.

Rename a Volume

Use this procedure to rename a disk or diskette logical volume.

Before You Start:

Obtain the following information:

- The hexadecimal address of the disk or diskette device the volume is on. (See "Listing the Devices on Your System" on page OP-318 to determine the device address.)
- The old name of the volume.
- The new volume name (1-6 characters).

1. Load the utility \$INITDSK:

- Using the \$L operator command, press the attention key and enter **\$L \$INITDSK**.
- Using the session manager, select option 3.1.

```
> $L $INITDSK
LOADING $INITDSK 83P,08:04:28, LP= 9100,...
$INITDSK - DISK INITIALIZATION UTILITY
COMMAND (?): _
```

2. Enter **RV** to rename a volume.

```
COMMAND (?): RV
ADDRESS: _
```

3. Enter the hexadecimal address of the device containing the volume.

```
DEVICE ADDRESS: 03
VOLUME: _
```

4. Enter the current volume name.

```
VOLUME: OLDVOL
NEWNAME: _
```

Storing and Handling Information

Renaming Volumes and Data Sets *(continued)*

5. Enter the new volume name (1-6 characters).

```
NEWNAME: NEWVOL  
RENAME COMPLETED  
  
COMMAND (?): _
```

6. Do you want to rename another volume?

- If so, continue with step 2 on page OP-131
- If not, enter EN to end \$INITDSK.

```
COMMAND (?): EN  
$INITDSK ENDED 08:06:30
```

Rename a Data Set

Use this procedure to rename a disk or diskette data set.

Before You Start:

Observe the following requirements:

- If the data set is on a diskette, make sure the diskette is in a diskette unit.
- Obtain the name of the volume on which the data set is found.
- Obtain the current name of the data set.
- Determine the new name of the data set (1-8 characters).

1. Load the utility \$DISKUT1:

- Using the \$L operator command, press the attention key and enter **\$L \$DISKUT1**.
- Using the session manager, select option 3.1.

```
> $L $DISKUT1  
LOADING $DISKUT1 49P,08:04:28, LP= 0000,...  
  
$DISKUT1 - DATA SET MANAGEMENT UTILITY I  
USING VOLUME EDX002  
  
COMMAND (?): _
```

Renaming Volumes and Data Sets *(continued)*

2. Is \$DISKUT1 using the correct volume name?

- If so, continue with step 3.
- If not, change the name of the volume by typing **CV xxxxxx**, where xxxxxx is the name of the volume you want to use.

```
USING VOLUME EDX002
```

```
COMMAND (?): CV MYVOL
USING VOLUME MYVOL.
COMMAND (?): _
```

3. Enter **RE** to rename a data set.

```
COMMAND (?): RE
MEMBER NAME: _
```

4. Enter the current name of the data set.

```
MEMBER NAME: MYPROG2
NEW NAME: _
```

5. Enter the new 1-8 character name being assigned to the data set.

```
NEW NAME: YOURPROG
RENAME COMPLETED
COMMAND (?): _
```

6. Do you want to rename another data set?

- Yes on the same volume, continue with step 3.
- Yes but on a different volume, continue with step 2.
- If not, enter **EN** to end \$DISKUT1.

```
COMMAND (?): EN
$DISKUT1 ENDED 08:30:24
```

Storing and Handling Information

Renaming a Diskette Volume Label and Owner Id

Use this procedure to rename a volume label and owner id on a diskette.

Before You Start:

Do the following:

- Insert the diskette, containing the volume label and owner id you wish to change, into the diskette drive.
- Determine the hexadecimal address of the disk or diskette device the volume is on. (See "Listing the Devices on Your System" on page OP-318 to determine the device address.)
- Determine the number of the slot the diskette is in if the diskette is in a 4966 Magazine Unit. (The diskette cannot be in a magazine.)
- Determine what the owner identification will be before starting the procedure. It can be a name, initials, department number, or any name you specify. This identification should consist of 1 to 14 characters.
- Determine the 1 to 6 character name you are assigning to the diskette volume label.

1. Load the utility \$INITDSK:

- Using the \$L operator command, press the attention key and enter **\$L \$INITDSK**.
- Using the session manager, select option 3.1.

```
> $L $INITDSK
LOADING $INITDSK 83P,08:04:28, LP=9100,...
$INITDSK - DISK INITIALIZATION UTILITY
COMMAND (?): -
```

2. Enter **RD** to rename a volume label and owner id.

```
COMMAND (?): RD
ADDRESS: -
```

3. Enter the hexadecimal address of the device you are using.

```
DEVICE ADDRESS: 02
CURRENT VOLUME: MYVOL
```

Note: If you receive the message "INVALID DEVICE TYPE" or a return code of 13, you may have entered an incorrect device address, or you may need to initialize the volume directory.

Renaming a Diskette Volume Label and Owner Id (*continued*)

4. The system supplies the current volume label.
Enter the new volume label (1-6 characters).

```
CURRENT VOLUME LABEL: MYVOL  
ENTER VOLUME LABEL: NEWVOL
```

5. The system supplies you with the current owner id.
Enter the new owner id (1-14 characters).

```
ENTER VOLUME LABEL: NEWVOL  
CURRENT OWNER ID: MYDSKT  
ENTER OWNER ID:MYDSKL  
COMMAND (?):
```

6. Do you want to rename another volume?
- If so, continue with step 2 on page OP-134
 - If not, enter **EN** to end \$INITDSK.

```
COMMAND (?): EN  
$INITDSK ENDED 08:06:30
```

Storing and Handling Information

Deleting Volumes and Data Sets

Use the procedures in this section to delete volumes and data sets from your disks and diskettes. (To delete data sets from tape see "Allocate Tape Data Sets" on page OP-127.)

Delete a Volume

Use this procedure to delete a logical volume from a disk or diskette device.

Before You Start:

Obtain the following information:

- The hexadecimal address of the device the volume is on. (See "Listing the Devices on Your System" on page OP-318 to get this number.)
- The name of the volume being deleted.

1. Load the utility \$INITDSK:

- Using the \$L operator command, press the attention key and enter **\$L \$INITDSK**.
- Using the session manager, select option 3.1.

```
> $L $INITDSK
LOADING $INITDSK 83P,08:04:28, LP=9100,...
$INITDSK - DISK INITIALIZATION UTILITY
COMMAND (?): _
```

2. Enter DV to delete a volume.

```
COMMAND (?): DV
ADDRESS: _
```

3. Enter the hexadecimal address of the device containing the volume.

```
ADDRESS: 03
VOLUME: _
```

4. Enter the volume name.

```
VOLUME: MYVOL
MYVOL DELETE? _
```

Deleting Volumes and Data Sets *(continued)*

5. Did you enter the correct volume name?

- If so, enter a **Y** and continue with step 6
- If not, or you do not want to delete the volume at this time, enter an **N** and continue with step 2 on page OP-136 to try again, or step 6 to end \$INITDSK.

```
MYVOL DELETE? Y
MYVOL DELETED
COMMAND (?): _
```

6. Do you want to delete another volume?

- If so, continue with step 3 on page OP-138.
- If not, enter **EN** to end \$INITDSK.

```
COMMAND (?): EN
$INITDSK ENDED 08:06:30
```

Delete One Data Set

Use this procedure to delete one data set from a disk or diskette volume.

Before You Start:

Obtain the following information:

- The name of the volume containing the data set being deleted.
- The name of the data set being deleted.

1. Load the utility \$DISKUT1:

- Using the \$L operator command, press the attention key and enter **\$L \$DISKUT1**.
- Using the session manager, select option 3.1.

```
> $L $DISKUT1
LOADING $DISKUT1 49P,08:04:28, LP=0000,...
$DISKUT1 - DATA SET MANAGEMENT UTILITY I
USING VOLUME EDX002
COMMAND (?): _
```

Storing and Handling Information

Deleting Volumes and Data Sets *(continued)*

2. Is \$DISKUT1 using the correct volume name?

- If so, continue with step 4.
- If not, change the name of the volume by typing **CV xxxxxx**, where xxxxxx is the name of the volume you want to use.

```
USING VOLUME EDX002
COMMAND (?): CV MYVOL
USING VOLUME MYVOL.
COMMAND (?): _
```

3. Enter **DE** to delete one data set.

```
COMMAND (?): DE
MEMBER NAME: _
```

4. Enter name of the data set you are deleting.

```
MEMBER NAME: PROGRAM1
PROGRAM1 DELETE? _
```

5. Did you enter the correct data set name?

- If so, enter a **Y** and continue with step 6.
- If not, or want to change you mind, enter an **N** and continue with step 3 to try again, or step 6 to end \$INITDSK.

```
PROGRAM1 DELETE? Y
PROGRAM1 DELETED
COMMAND (?): _
```

6. Do you want to delete another data set?

- If so, continue with step 3.
- If not, enter **EN** to end \$DISKUT1.

```
COMMAND (?): EN
$DISKUT1 ENDED 08:06:30
```

Deleting Volumes and Data Sets *(continued)*

Delete Multiple Data Sets

Use this procedure to delete multiple data sets from a disk or diskette volume. You can delete:

- All data sets with a specified prefix
- All data sets, except those with a specified prefix
- All data-type data sets with a specified prefix
- All data-type data sets, except those with a specified prefix
- All program-type data sets with a specified prefix
- All program-type data sets, except those with a specified prefix

Before You Start:

Obtain the following information:

- The name of the volume containing the data sets being deleted.
- The prefix you want to use.

1. Load the utility \$DISKUT1:

- Using the \$L operator command, press the attention key and enter \$L \$DISKUT1.
- Using the session manager, select option 3.1.

```
> $L $DISKUT1
LOADING $DISKUT1 49P,08:04:28,LP= 0000,...
$DISKUT1 - DATA SET MGMT. UTILITY I
USING VOLUME EDX002
COMMAND (?): _
```

2. Is \$DISKUT1 using the correct volume name?

- If so, continue with step 4 on page OP-140.
- If not, change the name of the volume by typing CV xxxxxx, where xxxxxx is the name of the volume you want to use.

```
USING VOLUME EDX002
COMMAND (?): CV MYVOL
USING VOLUME MYVOL.
COMMAND (?): _
```

Storing and Handling Information

Deleting Volumes and Data Sets *(continued)*

3. Do you want to delete all data sets with or without the prefix?

- If so, enter **SNQ** and \$DISKUT1 deletes them without prompting you for verification.
- If not, enter **SQ** and \$DISKUT1 will ask you to verify each data set before deleting it.

```
COMMAND (?): SQ
```

```
COMMAND (?): _
```

4. Enter the appropriate command for the type of delete you are doing:

- All data sets with a prefix, enter **DG**.
- All data sets except those with a prefix, enter **DNG**.
- All data-type data sets with a prefix, enter **DGD**.
- All data-type data sets except those with a prefix, enter **DNGD**.
- All program-type data sets with a prefix, enter **DGP**.
- All program-type data sets except those with a prefix, enter **DNGP**.

```
COMMAND (?): DG
```

```
ENTER GENERIC TEXT: _
```

5. Enter the prefix.

```
ENTER GENERIC TEXT: MY
```

```
DELETE ALL MEMBERS STARTING WITH MY (Y/N)? _
```

Deleting Volumes and Data Sets (continued)

6. \$DISKUT1 verifies your request:

- If you entered the correct prefix, enter a **Y** and continue with step 7.
- If you entered the wrong prefix, enter an **N** and continue with step 4 on page OP-140.

```
DELETE ALL MEMBERS STARTING WITH MY (Y/N)? Y
CONFIRM DELETE GENERIC REQUEST ON VOLUME: MYVOL
CONTINUE (Y/N)? _
```

7. Is \$DISKUT1 using the correct volume?

- If so, enter a **Y** and continue with step 8.
- If not, enter an **N** and continue with step 2 on page OP-139.

```
CONTINUE (Y/N)? Y
MYDATA2 DATA      532   200   NA
DELETE (Y/N)?
```

If you entered **SNQ** in step 3 on page OP-140, continue with step 9.

8. Do you want to delete the listed data set?

- If so, enter a **Y**.
- If not, enter an **N**.

Repeat this step until all data sets have been deleted.

```
MYDATA2 DATA      532   200   NA
DELETE? Y
MYDATA2 DELETED

      1531 FREE RECORDS IN LIBRARY

COMMAND (?) _
```

9. Do you want to delete another set of data sets?

- Yes and the volume name is the same, continue with step 3 on page OP-140.
- Yes and the volume name is different, continue with step 2 on page OP-139.
- If not, enter **EN** to end \$DISKUT1.

```
COMMAND (?): EN
$DISKUT1 ENDED 08:06:30
```

Storing and Handling Information

Compressing Volumes and Devices

Use this procedure to compress individual volumes or all volumes on a device.

Before You Start:

Consider the following information:

- If you are compressing the device or volume containing the supervisor (\$EDXNUC), you need a diskette that has been initialized for use with EDX data sets. This diskette is used to make a backup copy of the supervisor in case it is moved during the compress. If the supervisor is moved during the compress, you will not be able to IPL the system.

Step 1 of this procedure explains how to initialize the volume directory on this diskette and which supervisor modules to copy. (See "Initialize for EDX Format" on page OP-93 for the procedure if you need to initialize a diskette for EDX format.)

- There cannot be any programs running while the compress is being done. If you are using the session manager, the program \$SMUxxxx will be in the system, but it is not active until after the \$COMPRES utility ends.
- If you are compressing the entire device, you need the hexadecimal address of the device. (See "Listing the Devices on Your System" on page OP-318 to determine this address.)
- If you are compressing a single volume, you need the name of the volume.

1. If you are compressing the device or volume containing the supervisor (\$EDXNUC), make a backup copy of the supervisor in case it is moved by \$COMPRES.

If you are not compressing the device or volume containing the supervisor, skip to step 2.

Compressing Volumes and Devices *(continued)*

1(a). If you do not have a single volume diskette with a copy of the supervisor (\$EDXNUC) that can be used to IPL the system, use the \$INITDSK ID command to initialize an EDX format diskette as follows. (See “Initialize a Diskette Volume Directory” on page OP-106 for the step-by-step procedure.)

- Either a single- or multi-volume type. (The example uses a single-volume diskette.)
- Any volume label.
- A volume directory for 5 data sets.
- Allocate \$EDXNUC.
- Initialize IPL text.

```
> $L $INITDSK
COMMAND (?): ID
DEVICE ADDRESS: 02
INITIALIZE DEVICE MAY DESTROY ALL DATA
CONTINUE? Y
INITIALIZE DISKETTE AS MULTI-VOLUME TYPE? N
NEW VOLUME LABEL: COMPRS
ENTER OWNER IDENTIFICATION: BACKUP
COMPRS ALLOCATED
INITIALIZE THE VOLUME JUST ALLOCATED? Y
MAXIMUM NUMBER OF DATASETS? 5
DO YOU WANT WRITE VERIFY FOR THIS VOLUME? N
ALLOCATE $EDXNUC? Y
VOLUME INITIALIZED
INITIALIZE IPL TEXT? Y
IPL TEXT WRITTEN
ALLOCATE ANOTHER VOLUME? _
```

1(b). Use the \$COPYUT1 CM command to copy data sets \$LOADER and \$INITDSK from volume EDX002. If you have a 4978 display terminal as your SYSLOG terminal, copy \$4978ISO and \$4978CSO also. (The copy data set procedure is under “Copy a Data Set with Allocation” on page OP-152.)

```
> $L $COPYUT1
THE DEFINED SOURCE VOLUME IS EDX002, OK? Y
THE DEFINED TARGET VOLUME IS EDX002, OK? N
ENTER NEW TARGET VOLUME: COMPRS
MEMBER WILL BE COPIED FROM EDX002 TO COMPRS, OK? Y
COMMAND(?): CM
ENTER FROM (SOURCE) MEMBER: $LOADER
ENTER TO (TARGET) MEMBER OR * FOR
SAME NAME AS SOURCE: *
$LOADER COPY COMPLETE      nn RECORDS COPIED
COMMAND(?): _
```

Storing and Handling Information

Compressing Volumes and Devices *(continued)*

1(c). Verify that this diskette can be used to IPL the system before proceeding:

- If it works, re-IPL the system from disk and continue with step 2.
- If it does not work, repeat step 1.

2. Verify that there are no programs running by pressing the attention key and typing **\$A ALL**. If you are using the session manager, the **\$SMUxxxx** program will be in the system, but it is not active until after **\$COMPRES** ends.

```
> $A ALL  
PROGRAMS AT 08:14:19  
IN PARTITION #1 NONE  
PART.ADDR: 0000 HEX; SIZE: 12288 DEC.BYTES  
IN PARTITION #2 NONE  
PART.ADDR: 0000 HEX; SIZE: 65536 DEC.BYTES
```

3. Load the utility **\$COMPRES**:

- Using the **\$L** operator command, press the attention key and enter **\$L \$COMPRES**.
- Using the session manager, select option 3.4.

```
> $L $COMPRES  
LOADING $COMPRES 26P 08:15:20, LP= 4C00, PART= 1  
  
VOLUME/DEVICE COMPRESS  
WARNING! SHOULD BE RUN ONLY WHEN  
NO OTHER PROGRAMS ARE ACTIVE  
SET TERMINAL IN ROLL MODE (Y/N)? _
```

4. Do you want the display terminal to operate in roll mode? (In roll mode, you do not have to press the enter key when the display screen is full.)

- If so, enter a **Y**.
- If not, enter an **N**.

```
SET TERMINAL IN ROLL MODE (Y/N)? Y  
DO YOU WISH TO COMPRESS A DEVICE OR VOLUME?  
ENTER ('D' FOR DEVICE | 'V' FOR VOLUME): _
```

Compressing Volumes and Devices *(continued)*

5. Are you compressing the entire device or a single volume?

- If an entire device, enter a **D** and continue with step 6(a).
- If a single volume, enter a **V** and continue with step 6(b).

6(a). If you are compressing the entire device, enter the hexadecimal device address. Continue with step 7(a) or 7(b) as appropriate.

```
DO YOU WISH TO COMPRESS A DEVICE OR VOLUME?  
ENTER ('D' FOR DEVICE | 'V' FOR VOLUME): D  
DEVICE ADDRESS: 48
```

```
COMPRESS DEVICE AT ADDRESS 0048? _
```

```
CAUTION: YOU ARE ABOUT TO COMPRESS THE IPL VOLUME>  
IF THE NUCLEUS IS MOVED THEN THIS VOLUME WILL NOT IPL  
AGAIN, UNTIL THE "I" COMMAND OF "$INITDSK" IS REISSUED.  
ALSO IF "$LOADER" IS MOVED THEN NO PROGRAMS CAN BE LOADED.  
DO YOU WISH TO PROCEED (Y/N): _
```

6(b). If you are compressing a single volume, enter the name of the volume. Continue with step 7(a) or 7(b) as appropriate.

```
DO YOU WISH TO COMPRESS A DEVICE OR VOLUME?  
ENTER ('D' FOR DEVICE | 'V' FOR VOLUME): V
```

```
VOLUME LABEL = EDX002
```

```
COMPRESS VOLUME ON EDX002? _
```

```
CAUTION: YOU ARE ABOUT TO COMPRESS THE IPL VOLUME>  
IF THE NUCLEUS IS MOVED THEN THIS VOLUME WILL NOT IPL  
AGAIN, UNTIL THE "I" COMMAND OF "$INITDSK" IS REISSUED.  
ALSO IF "$LOADER" IS MOVED THEN NO PROGRAMS CAN BE LOADED.  
DO YOU WISH TO PROCEED (Y/N): _
```

Storing and Handling Information

Compressing Volumes and Devices (*continued*)

7(a). If you are compressing the IPL volume, \$COMPRES issues a warning message and gives you a chance to change your mind if you do not have a backup diskette set up as described in step 1 or a current backup copy of your system.

If you do not receive the caution message, skip to step 7(b).

- Enter a **Y** to continue the compress.
- Enter an **N** to cancel the compress.

```
DO YOU WISH TO PROCEED (Y/N): Y
COMPRESS VOLUME ON MYVOL? _
```

7(b). If you are not compressing the IPL volume, \$COMPRES asks you to verify that you entered the correct device address, or volume name.

- If you did, enter a **Y** and continue with step 8.
- If you entered the wrong device address or volume name, enter an **N** and return to step 6(a) or 6(b) as appropriate.

```
COMPRESS VOLUME ON EDX002? Y
DIRECTORY HAS BEEN SORTED BY MEMBER IN ASCENDING ORDER
$EDXNUC COPIED
$INITIAL COPIED
.
.
THE VOLUME IS COMPRESSED.
ANOTHER COMPRESS? _
```

8. Do you want to do another compress?

- If so, enter a **Y** and continue with step 3.
- If not, enter an **N**.

```
ANOTHER COMPRESS? N
WANT TO SORT THE VOLUME DIRECTORY (Y/N)? _
```

Compressing Volumes and Devices (*continued*)

9. Do you want to sort the volume directory?

- If so, enter a **Y** and refer to the **\$DIRECT** utility in the *Operator Commands and Utilities Reference*. (The directory should be sorted only when no other programs are running.)
- If not, enter an **N**.

```
WANT TO SORT THE VOLUME DIRECTORY (Y/N)? N
```

```
$COMPRES ENDED AT 08:16:03
```

10. If you compressed the device or volume containing the supervisor and you cannot IPL the system from that volume, **\$EDXNUC** was moved by **\$COMPRES** and you must write IPL text on the volume as describe in steps 10(a) through 10(d).

10(a). IPL the system from the backup diskette created in step 1.

10(b). After IPL completes, load the utility **\$INITDSK** by typing **\$L \$INITDSK**.

```
> $L $INITDSK
LOADING $INITDSK      83P,08:30:20, LP= 0000,...
$INITDSK - DISK INITIALIZATION UTILITY
COMMAND (?): _
```

10(c). Enter **II nucleus volume** where **nucleus** is the name of the nucleus being initialized, and **volume** is the name of the volume containing the nucleus.

```
COMMAND (?): II $EDXNUC EDX002
IPL TEXT WRITTEN
COMMAND (?): _
```

10(d). Enter **EN** to end **\$INITDSK**. You should now be able to IPL from the disk.

```
COMMAND (?): EN
$INITDSK ENDED AT 08:45:00
```

Storing and Handling Information

Copying Data on EDX Systems

Use the procedures in this section to copy the contents of volumes and data sets used on your system. You can also use these procedures to copy the contents of volumes and data sets to a diskette for transfer to another EDX system.

- “Transmitting Data from System to System” shows you how to transmit the contents of data sets and volumes from one Series/1 to another.
- “Copy a Data Set with Allocation” on page OP-152 shows you how to copy one or more data sets with the target data sets being allocated automatically.
- “Copy a Volume or Data Set without Allocation” on page OP-162 shows you how to copy a volume or all or part of a data set to a previously defined data set. The target (copy to) data set must already exist.
- “Copy to or from a Tape Data Set” on page OP-167 shows you how to copy data sets on tape.

Notes:

1. On any copy operation, the current contents of the target, (copy to) volume or data set will be lost.
2. Two people should not specify the dynamic storage option (for example, \$L \$COPYUT1,,48000) when loading \$COPYUT1 to make copies of the same volume. To do so causes volume table of content problems.

Transmitting Data from System to System

Use this procedure to transmit the contents of data sets and volumes from one Series/1 to another.

Before You Start:

Determine the following:

- The name of the volume containing the data set you wish to transfer.
- The name of the volume into which the data set will be transferred.
- The name of the source data set.
- The system you wish to send data to. The system receiving data becomes the remote system. The system sending data becomes the local system. You must always start this procedure on the receiving Series/1.
- The address of the bisynchronous communications line on both the remote system and the local system.
- If you are using a diskette, determine the address of the diskette drive. You must vary the diskette drive online.

Copying Data on EDX Systems (*continued*)

1. Load the \$TRANS utility on the remote system. The remote system is the system receiving the data.

To do this, press the attention key and enter \$L \$TRANS.

```
> $L $TRANS
LOADING $TRANS   XXP,08:30:20, LP= 0000, PART= 2
```

```
$TRANS - BISYNC TRANSPORT UTILITY
```

```
THIS UTILITY SHOULD ONLY BE USED WHEN TRANSPORTING
DATA SETS OR PROGRAMS OVER A BISYNC LINE.
DO YOU WISH TO CONTINUE (Y/N)? _
```

2. If you plan to transmit data from one system to another over a BISYNC line, reply Y to continue.

```
THIS UTILITY SHOULD ONLY BE USED WHEN TRANSPORTING
DATA SETS OR PROGRAMS OVER A BISYNC LINE.
DO YOU WISH TO CONTINUE (Y/N)? Y
```

```
IS THIS THE TRANSMITTING SYSTEM (Y/N)? _
```

3. Reply N to this prompt. (When you reply N to this prompt, you designate this system as the remote and the system to receive data.)

```
IS THIS THE TRANSMITTING SYSTEM (Y/N)? N
```

```
BSC RECEIVE PROGRAM ACTIVE
```

```
ENTER BSC LINE ADDRESS IN HEX: _
```

4. Enter the BSC line address. If you do not know the address, refer to "Listing the Devices on Your System" on page OP-318.

```
ENTER BSC LINE ADDRESS IN HEX: 09
```

```
BSC LINE OPENED - READY FOR TRANSMISSION
```

Note: As soon as you receive the ready for transmission message you are finished on the remote system. You are ready to load the \$TRANS utility on the local system.

5. Load the \$TRANS utility on the local system. The local system is the system transmitting the data.

To do this, press the attention key and enter \$L \$TRANS.

```
> $L $TRANS
LOADING $TRANS   XXP,08:30:20, LP= 0000, PART= 2
```

```
$TRANS - BISYNC TRANSPORT UTILITY
```

```
THIS UTILITY SHOULD ONLY BE USED WHEN TRANSPORTING
DATA SETS OR PROGRAMS OVER A BISYNC LINE.
DO YOU WISH TO CONTINUE (Y/N)? _
```

Storing and Handling Information

Copying Data on EDX Systems (*continued*)

6. If you plan to transmit data from one system to another over a BISYNC line, reply Y to continue.

```
THIS UTILITY SHOULD ONLY BE USED WHEN TRANSPORTING  
DATA SETS OR PROGRAMS OVER A BISYNC LINE.  
DO YOU WISH TO CONTINUE (Y/N)? Y
```

```
IS THIS THE TRANSMITTING SYSTEM (Y/N)? _
```

7. Reply Y to this prompt. (When you reply Y to this prompt, you designate this system as the local and the system to transmit data.)

```
IS THIS THE TRANSMITTING SYSTEM (Y/N)? Y
```

```
BSC RECEIVE PROGRAM ACTIVE
```

```
ENTER BSC LINE ADDRESS IN HEX: _
```

8. Enter the BSC line address. If you do not know the address, refer to "Listing the Devices on Your System" on page OP-318.

```
ENTER BSC LINE ADDRESS IN HEX: 09
```

```
BSC LINE OPENED
```

```
ENTER THE SOURCE VOLUME NAME: _
```

9. Enter the names of the source and target volumes.

```
ENTER THE SOURCE VOLUME NAME: FRVOL
```

```
ENTER THE TARGET VOLUME NAME: TOVOL
```

```
COMMAND (?): _
```

10. Enter CM to copy a single member from one system to another.

```
COMMAND (?): CM
```

```
ENTER DATA SET NAME: _
```

Copying Data on EDX Systems (*continued*)

11. Specify the name of the data set you wish to copy.

```
ENTER DATA SET NAME: ALI
ALI COPY COMPLETE
```

12. Do you want to copy a data set on a different volume?

- If yes, enter the CV command to change the source and target volumes. Continue with step 13.

Note: The system defaults to the volumes you specified previously, unless you use the CV command to change them

- If not, continue with step {chuc}.

```
COMMAND (?): CV
ENTER THE SOURCE VOLUME NAME: ___
ENTER THE TARGET VOLUME NAME: ___
```

13. Enter the names of the source and target volumes you wish to copy.

```
COMMAND (?): CV
ENTER THE SOURCE VOLUME NAME: YOURVOL
ENTER THE TARGET VOLUME NAME: MYVOL
MEMBERS ON VOLUME YOURVOL WILL BE COPIED TO VOLUME MYVOL
COMMAND (?): ___
```

14. chuc. Do you want to copy another data set?

- Yes, continue with 11
- No, enter an ES or EN to end \$TRANS.

Note: The ES command closes the \$TRANS utility down on both the receiving and transmitting ends.

```
COMMAND (?): ES
BSC LINE CLOSED
REMOTE BSC LINE CLOSED
$TRANS ENDED AT 12:14:00
```

Storing and Handling Information

Copying Data on EDX Systems *(continued)*

Copy a Data Set with Allocation

Use the following procedures to copy one or more data sets from one volume to another on a disk or diskette device.

These procedures use the utility \$COPYUT1. If the target data set doesn't exist, \$COPYUT1 allocates it. If the target data set already exists, \$COPYUT1 deletes it, and reallocates it to the required size. Any data in the target data set is lost.

Copy One Data Set

Use this procedure to copy one data set to another volume. The target set is automatically allocated by \$COPYUT1. (See "Copy All or Selected Data Sets" on page OP-155 to copy multiple data sets at one time.)

Before You Start:

Obtain the following information:

- The name of the volume containing the source (copy from) data set.
- The name of the volume into which it will be copied (target).
- The name of the source data set.
- The 1-8 character name you want assigned to the target data set, if it is to be renamed.

1. Load the utility \$COPYUT1.

- Using the \$L operator command, press the attention key and enter \$L \$COPYUT1.
- Using the session manager, select option 3.3.

```
> $L $COPYUT1
LOADING $COPYUT1   XXP,08:30:20, LP= 0000, PART= 2

$COPYUT1 - DATA SET COPY UTILITY

      ***WARNING***
MEMBERS ON TARGET VOLUME WILL BE DELETED
REALLOCATION AND COPYING OR MEMBERS IS
DEPENDENT ON SUFFICIENT CONTIGUOUS SPACE

THE DEFINED SOURCE VOLUME IS EDX002, OK(Y/N)?
```

2. Is the source volume correct?

- If so, enter a Y and continue with step 4 on page OP-153.
- If not, enter an N.

```
THE DEFINED SOURCE VOLUME IS EDX002, OK(Y/N)?N
ENTER NEW SOURCE VOLUME: _
```

Copying Data on EDX Systems (*continued*)

3. Enter the name of the source volume.

```
ENTER NEW SOURCE VOLUME: MYTRGL
THE DEFINED TARGET VOLUME IS EDX002, OK(Y/N)?_
```

4. Is the target volume correct?

- If so, enter a Y and continue with step 6.
- If not, enter a N.

```
THE DEFINED TARGET VOLUME IS EDX002, OK(Y/N)?_
ENTER NEW TARGET VOLUME: _
```

5. Enter the target volume name.

```
ENTER NEW TARGET VOLUME: MYTRGT
MEMBER WILL BE COPIED FROM MYSRCE TO MYTRGT OK(Y/N)?_
```

6. Did you enter the correct volume names?

- If so, enter a Y and continue with step 7.
- If not, enter an N and continue with step 3.

```
MEMBER WILL BE COPIED FROM MYSRCE TO MYTRGT OK(Y/N)?Y
COMMAND (?): _
```

7. Enter CM to copy your data set.

```
COMMAND (?): CM
ENTER FROM (SOURCE) MEMBER: _
```

Storing and Handling Information

Copying Data on EDX Systems *(continued)*

8. Enter the name of the data set from which you are copying.

```
ENTER FROM (SOURCE) MEMBER: FROMBR
ENTER TO (TARGET) MEMBER OR * FOR SAME NAME AS SOURCE: _
```

9. Enter the name of the data set to which you are copying. If you want the same data set name, you can enter an * instead of the name. If the data set does not exist, \$COPYUT1 allocates it for you.

```
ENTER TO (TARGET) MEMBER OR * FOR SAME NAME AS SOURCE:TOMBR
TOMBR COPY COMPLETE          30 RECORDS COPIED

COMMAND (?): _
```

10. Do you want to copy another data set?

```
COMMAND (?):EN
$COPYUT1 ENDED 08:49:30
```

- If so, and the from/to volumes are the same, continue with step 7 on page OP-153.
- If so, but the volume names are different, enter **CV** and continue with step 2 on page OP-152.
- If not, you are finished. Enter **EN** to end \$COPYUT1.

Copying Data on EDX Systems *(continued)*

Copy All or Selected Data Sets

Use this procedure to copy all data sets from one volume to another. With this procedure, you can do the following types of copies:

- All or selected data sets regardless of type.
- All or selected data-type data sets.
- All or selected program-type data sets.

If the target data set names already exist, \$COPYUT1 will delete them and reallocate them. (See “Copy All or Selected Data Sets by Prefix” on page OP-159 to copy all or selected data sets with or without a prefix.)

Before You Start:

Determine the following:

- The name of the source (copy from) volume containing the data sets to be copied.
- The name of the target (copy to) volume into which you are copying.

1. Load the utility \$COPYUT1:

- Using the \$L operator command, press the attention key and enter \$L \$COPYUT1.
- Using the session manager, select option 3.3.

```
> $L $COPYUT1
LOADING $COPYUT1  52P,08:30:20, LP= 0000,...
$COPYUT1 - DATA SET COPY UTILITY
*** WARNING ***
MEMBERS ON TARGET VOLUME WILL BE DELETED
REALLOCATION AND COPYING OF MEMBERS IS
DEPENDENT ON SUFFICIENT CONTIGUOUS SPACE
THE DEFINED SOURCE VOLUME IS EDX002, OK?
```

2. Is the source volume correct?

- If so, enter a Y and continue with step 4 on page OP-156.
- If not, enter an N.

```
THE DEFINED SOURCE VOLUME IS EDX002, OK? N
ENTER NEW SOURCE VOLUME: -
```

Storing and Handling Information

Copying Data on EDX Systems (*continued*)

3. Enter the name of the source volume.

```
ENTER NEW SOURCE VOLUME: MYSRCE
THE DEFINED TARGET VOLUME IS EDX002, OK? _
```

4. Is the target volume correct?

- If so, enter a **Y** and continue with step 6.
- If not, enter an **N**.

```
THE DEFINED TARGET VOLUME IS EDX002, OK? N
ENTER NEW TARGET VOLUME: _
```

5. Enter the target volume name.

```
ENTER NEW TARGET VOLUME: MYTRGT
MEMBER WILL BE COPIED FROM MYSRCE TO MYTRGT OK (Y/N)? _
```

6. Are the from/to volume names correct?

- If so, enter a **Y** and continue with step 7.
- If not, enter an **N** and continue with step 2 on page OP-155.

```
MEMBER WILL BE COPIED FROM MYSRCE TO MYTRGT OK (Y/N)? Y
COMMAND (?): _
```

7. Do you want to copy all or selected data sets in the group you are copying?

- All data sets, enter **NQ**.
- Selected data sets, enter **SQ** and \$COPYUT1 will ask you to verify each data set before copying it.

```
COMMAND (?): SQ
COMMAND (?): _
```

Copying Data on EDX Systems *(continued)*

8. Enter **ROLLON** to tell \$COPYUT1 to operate in roll-screen mode. In roll screen mode, \$COPYUT1 will not pause to wait for you to press the enter key when the display screen is full.

```
COMMAND (?): ROLLON
```

```
COMMAND (?): _
```

9. Enter the appropriate command, for the type of copy you are doing:

- To copy all or selected data sets in a volume, enter **CALL**.
- To copy all or selected data-type data sets in a volume, enter **CAD**.
- To copy all or selected program-type data sets in a volume, enter **CAP**.

```
COMMAND (?): CALL  
COPY FROM BEGINNING (Y/N)? _
```

10. Do you want to start the copy at the beginning of the volume, or with a specified data set?

- At the beginning of the volume, enter a **Y** and continue with step 12 on page OP-158 .
- With a specified data set, enter an **N** and continue with step 11.

```
COPY FROM BEGINNING (Y/N)? N  
ENTER STARTING MEMBER: _
```

11. Enter the name of the first data set you are copying.

```
ENTER STARTING MEMBER: FIRSTDS  
COPY ALL THE MEMBERS AFTER FIRSTDS (Y/N)? _
```

Storing and Handling Information

Copying Data on EDX Systems *(continued)*

12. Do you want to copy the rest of the data sets in the volume?

- If you want \$COPYUT1 to copy the rest of the data sets in the volume, enter a **Y** and continue with step 14.
- If you want \$COPYUT1 to stop copying with a specified data set, enter an **N** and continue with step 13 (step 15 if you entered **NQ** in step 7 on page OP-156).

```
COPY ALL THE MEMBERS AFTER FIRSTDS (Y/N)? N
ENTER ENDING MEMBER : _
```

13. Enter the name of the last data set you want \$COPYUT1 to copy. (If you entered **NQ** in step 7 on page OP-156, skip to step 15.)

```
ENTER ENDING MEMBER : LASTDS
COPY FIRSTDS?
```

14. Do you want to copy the listed data set?

- If so, enter a **Y**; \$COPYUT1 copies the data set and asks you if you want to copy the next data set.
- If not, enter an **N**; \$COPYUT1 asks if you want to copy the next data set.

```
COPY FIRSTDS ? Y
FIRSTDS      COPY COMPLETE   30 RECORDS COPIED
COMMAND (?): _
```

Repeat this step until all data sets have been copied.

15. Do you want to copy more data sets?

- Yes and the from/to volumes are the same, continue with step 9 on page OP-157.
- Yes but the volume names are different, enter **CV** and continue with step 2 on page OP-155.
- If not, enter **EN** to end \$COPYUT1.

```
COMMAND (?): EN
$COPYUT1 ENDED 08:49:30
```

Copying Data on EDX Systems *(continued)*

Copy All or Selected Data Sets by Prefix

Use this procedure to copy all data sets from one volume to another based on a specified prefix. If the target data sets already exist, \$COPYUT1 will delete them and reallocate them. (See "Copy All or Selected Data Sets" on page OP-155 to copy all or selected data sets without using a prefix.)

Before You Start:

Determine the following:

- The name of the source (copy from) volume containing the data sets to be copied.
- The name of the target (copy to) volume into which you are copying.
- The 1-8 character prefix.

1. Load the utility \$COPYUT1:

- Using the \$L operator command, press the attention key and enter \$L \$COPYUT1.
- Using the session manager, select option 3.3.

```
> $L $COPYUT1
LOADING $COPYUT1 52P,08:30:20, LP= 0000,...

$COPYUT1 - DATA SET COPY UTILITY

*** WARNING ***
MEMBERS ON TARGET VOLUME WILL BE DELETED
REALLOCATION AND COPYING OF MEMBERS IS
DEPENDENT ON SUFFICIENT CONTIGUOUS SPACE

THE DEFINED SOURCE VOLUME IS EDX002, OK?
```

2. Is the source volume correct?

- If so, enter a Y and continue with step 4 on page OP-160.
- If not, enter an N.

```
THE DEFINED SOURCE VOLUME IS EDX002, OK? N
ENTER NEW SOURCE VOLUME: _
```

3. Enter the name of the source volume.

```
ENTER NEW SOURCE VOLUME: MYSRCE
THE DEFINED TARGET VOLUME IS EDX002, OK? _
```

Storing and Handling Information

Copying Data on EDX Systems *(continued)*

4. Is the target volume correct?
- If so, enter a **Y** and continue with step 6.
 - If not, enter an **N**.

```
THE DEFINED TARGET VOLUME IS EDX002, OK? N
ENTER NEW TARGET VOLUME: _
```

5. Enter the target volume name.

```
ENTER NEW TARGET VOLUME: MYTRGT
MEMBER WILL BE COPIED FROM MYSRCE TO MYTRGT OK (Y/N)? _
```

6. Are the from/to volume names correct?
- If so, enter a **Y** and continue with step 7.
 - If not, enter an **N** and continue with step 2 on page OP-159.

```
MEMBER WILL BE COPIED FROM MYSRCE TO MYTRGT OK (Y/N)? Y
COMMAND (?): _
```

7. Do you want to copy all or selected data sets in the group you are copying?
- All data sets, enter **NQ**.
 - Selected data sets, enter **SQ** and \$COPYUT1 will ask you to verify each data set before copying it.

```
COMMAND (?): SQ
COMMAND (?): _
```

8. If you are copying all data sets, you may want to tell \$COPYUT1 to operate in roll-screen mode. In roll screen mode, \$COPYUT1 will not pause to wait for you to press the enter key when the display screen is full.

```
COMMAND (?): ROLLON
COMMAND (?): _
```

Copying Data on EDX Systems (*continued*)

9. Enter the appropriate command, for the type of copy you are doing:

- To copy all or selected data sets beginning with a prefix, enter **CG**.
- To copy all or selected data sets that do not begin with a prefix, enter **CNG**.

```
COMMAND (?): CG
ENTER GENERIC TEXT: _
```

10. Enter the 1 to 8 character prefix.

(If you entered **NQ** in step 7 on page OP-160, continue with step 12.)

```
ENTER GENERIC TEXT: MY
COPY MYDATA1 ? _
```

11. Do you want to copy the listed data set?

- If so, enter a **Y**; \$COPYUT1 copies the data set and asks you if you want to copy the next data set.
- If not, enter an **N**; \$COPYUT1 asks if you want to copy the next data set.

Repeat this step until all data sets that you want copied have been copied.

```
COPY MYDATA1 ? Y
MYDATA1 COPY COMPLETE 30 RECORDS COPIED
COMMAND (?): _
```

12. Do you want to copy more data sets?

- Yes and the from/to volumes are the same, continue with step 9.
- Yes but the volume names are different, enter **CV** and continue with step 2 on page OP-159.
- If not, enter **EN** to end \$COPYUT1.

```
COMMAND (?): EN
$COPYUT1 ENDED 08:49:30
```

Storing and Handling Information

Copying Data on EDX Systems *(continued)*

Copy a Volume or Data Set without Allocation

Use the three procedures in this section to copy a volume, or all or part of a data set, to a new volume or data set that has already been allocated. If you need to allocate a volume or data set see "Allocate a Volume" on page OP-121 or "Allocate a Data Set" on page OP-124; or use the procedures under "Copy a Data Set with Allocation" on page OP-152 to do the copy.

When copying a data set, the target (copy to) data set must have the same type (data or program) as the source (copy from) data set.

Copy a Volume

Use this procedure to copy an entire volume. A volume can be copied to the same or a different device.

If you are copying a diskette volume to a disk, you must copy it to a disk volume. The following chart shows the number of records required in the target disk volume, for the type of diskette. You also want to make sure that the data set directory is large enough to contain the directory information.

Diskette type	Number of records at 128 BPS	Number of records at 256 BPS	Number of records at 1024 BPS
Diskette 1	946	1107	-
Diskette 2	1921	2217	-
Diskette 2-D	-	3845	4733

Before You Start:

Observe the following requirements:

- If the target (copy to) volume is not already allocated, you must allocate it. See "List All Volumes" on page OP-181 if you need to verify whether the target volume exists. See "Allocate a Volume" on page OP-121 if you need to allocate it.
- Determine the name of the source (copy from) volume.
- Determine the name of the target (copy to) volume.

Copying Data on EDX Systems (*continued*)

1. Load the utility \$COPY:

- Using the \$L operator command, press the attention key and enter \$L \$COPY.
- Using the session manager, select option 3.5.

```
> $L $COPY_
LOADING $COPY      40P,08:39:20, LP= 0000,...
$COPY - COPY UTILITY
COMMAND(?): ? _
```

2. Enter CV to copy the contents of an entire volume.

```
COMMAND (?): CV
COPY VOLUME
ENTER SOURCE VOLUME: _
```

3. Enter the name of the source (copy from) volume.

```
ENTER SOURCE VOLUME: VOLFR
ENTER TARGET VOLUME: _
```

4. Enter the name of the target (copy to) volume.

```
ENTER TARGET VOLUME: VOLTO
ARE ALL PARAMETERS CORRECT (Y/N)? _
```

5. Have you entered the correct volume names?

- If so, enter a Y.
- If not, enter an N and \$COPY will not do the copy.

```
ARE ALL PARAMETERS CORRECT (Y/N)? Y
COPY COMPLETE
      2000 RECORDS COPIED
COMMAND (?): _
```

6. Do you want to copy another volume?

- If so, continue with step 2.
- If not, enter EN to end \$COPY.

```
COMMAND (?): EN
$COPY      ENDED AT 08:49:05
```

Storing and Handling Information

Copying Data on EDX Systems *(continued)*

Copy a Data Set

Use this procedure to copy a data set from one volume to another. You can copy all of a data set with a type of *program*, and you can copy all, or part, of a data set with a type of *data*. You can also merge the contents of two data type data sets by specifying that the second data set start one record after the end of the first, if the target data set is large enough to hold both data sets.

Before You Start:

Observe the following requirements:

- Ensure that the target (copy to) data set is allocated. See “List One or More Data Sets on a Volume” on page OP-187 to determine if it has been allocated. See “Allocate a Data Set” on page OP-124 if you have to allocate it.
- If you are copying a data set with a type of *program*, the target (copy to) data set must have the same number of records as the source (copy from) data set. If you are copying a data set with a type of *data*, the target data set has to be large enough to hold the number of records being copied.
- Get the data set and volume names of the source data set.
- If you are copying only part of the data set, get the numbers of the first and last records in the portion you are copying.
- If you are merging data sets, get the number of the record in the target data set where you want the copy to start.

1. Load the utility \$COPY:

- Using the \$L operator command, press the attention key and enter \$L \$COPY.
- Using the session manager, select option 3.5.

```
> $L $COPY
LOADING $COPY      40P,08:39:20, LP= 0000,...

$COPY      40P,08:39:20, LP= 0000, PART= 2

COMMAND(?): ? _
```

2. Enter CD to copy the contents of a data set.

```
COMMAND (?): CD

SOURCE (NAME,VOLUME): _
```

Copying Data on EDX Systems *(continued)*

- 3.** Enter the name of the source data set and volume.

If you are copying a program-type data set, continue with step 7.

```
SOURCE (NAME,VOLUME): FROMDS,MYVOL
COPY ENTIRE DATA SET (Y/N)? _
```

- 4.** If you are copying a data-type data set, you are asked whether you want to copy all or part of the data set.

- If you are copying the entire data-type data set, enter a **Y** and continue with step 7.
- If you are copying part of a data-type data set, enter an **N** and continue with step 5.

```
COPY ENTIRE DATA SET (Y/N)? N
FIRST RECORD: _
```

- 5.** Enter the number of the first record you want copied.

```
FIRST RECORD: 1
LAST RECORD: _
```

- 6.** Enter the number of the last record you want copied.

```
LAST RECORD: 3
TARGET (NAME,VOLUME): _
```

- 7.** Enter the target data set and volume names.

If you are copying a program-type data set, continue with step 10 on page OP-166.

```
TARGET (NAME,VOLUME): TODS,MYVOL
FIRST RECORD? _
```

Storing and Handling Information

Copying Data on EDX Systems *(continued)*

8. If you are copying a data-type data set, \$COPY asks you for the number of the first record where you want it to start copying into the target data set.

```
FIRST RECORD: 1
ARE ALL PARAMETERS CORRECT (Y/N)? _
```

9. Have you entered the correct names and record numbers?

- If you have, enter a Y.
- If you have not, enter an N and \$COPY will not do the copy.

```
ARE ALL PARAMETERS CORRECT (Y/N)? Y
COPY COMPLETE

      3 RECORDS COPIED

COMMAND (?): _
```

10. Do you want to copy another data set?

- If so, continue with step 3 on page OP-165.
- If not, enter EN to end \$COPY.

```
COMMAND (?): EN

$COPY ENDED AT 08:49:05
```

Copying Data on EDX Systems (*continued*)

Copy to or from a Tape Data Set

Use this procedure to:

- Copy a disk or diskette data set onto a tape.
- Copy a tape data set onto a disk or diskette data set.
- Copy a tape data set onto another tape.
- Merge two or more data sets onto one tape data set.

Consider the following when copying data sets:

- When \$TAPEUT1 encounters a tapemark (end of input data), it asks if it should continue. If there is enough room on the target tape for the remaining records, enter a Y in response to the question.
- Use one of the following to create an exact duplicate of one tape onto another:
 - To copy only data records, initialize the target tape with the same label as the source tape. When copying the tape, \$TAPEUT1 creates a new header label on the target tape and only duplicates the data records.
 - To create an exact duplicate of the source tape, including the header label, mount both tapes on drives specified for bypass label processing. When copying the tape, \$TAPEUT1 creates an exact copy of the tape, including all labels.
- The copy can be continued on another tape or disk data set if the source is larger than the target tape or data set.

Before You Start:

Observe the following requirements:

- Verify that the target (copy to) data set has been allocated.
- Obtain the hexadecimal address of the tape drive. (The address is displayed when you load \$TAPEUT1 or you can use the procedure “Listing the Devices on Your System” on page OP-318 to determine this address.)
- Obtain the file number(s) of the data set(s) on the tape(s). Get this number from the external label on the tape reel or the tape log.
- Obtain the data set and volume names of the source (copy from) data set.
- Obtain the data set and volume names of the target (copy to) data set.
- Obtain the blocksize of the source data set. The blocksize is not needed if the source data set is on a disk or diskette.

Note: The examples with this procedure illustrate copying from a disk data set to a tape data set.

Storing and Handling Information

Copying Data on EDX Systems (*continued*)

1. Enter **\$VARYON ioda file EX** to vary the tape online to the file number of the data set being copied where:

ioda is the hexadecimal device address of the tape drive.

file is the decimal file number being accessed. (This parameter is required with standard label tapes only.)

EX specifies expiration date override. EX only has to be entered if the target data set was initialized with an expiration date that has not expired and you want to be able to override it. (This parameter is required with standard label tapes only.)

If you enter the parameter EX, you will receive the message "IGNORE EXPIRATION DATE?". If you do not enter EX, skip to step 3.

```
> $VARYON 4C 2 EX
IGNORE EXPIRATION DATE? (Y,N): _
```

2. If you entered the EX parameter, \$TAPEUT1 asks you to verify that you want to be able to override an unexpired expiration date.

- If you do, enter a Y.
- If you do not, enter an N.

```
IGNORE EXPIRATION DATE? (Y,N): Y
MYVOL ONLINE
```

3. Are you copying from tape to tape? If you are, repeat steps 1 and 2 for the second tape drive before continuing.

Copying Data on EDX Systems *(continued)*

4. Load the utility \$TAPEUT1:

- Using the \$L operator command, press the attention key and enter **\$L \$TAPEUT1**.
- Using the session manager, select option 3.10.

After it is loaded, \$TAPEUT1 displays the following information about the tapes defined on your system:

- The tape drive ID (TAPEID).
- The word DUAL if the tape is set for dual density.
- The label type (SL=standard label, NL=non-labeled, BLP= bypass label processing).
- Current density setting:
 - 4968 = 1600 or 3200.
 - 4969 = 800 or 1600.
- The volume name, if the tape is an online standard label tape.
- The tape speed, if the tape device is a 4968.
- Whether the tape is online or offline.
- The number of the file where the tape is currently positioned.
- The tape device address.

```
> $L $TAPEUT1
$TAPEUT1      21P,08:01:20, LP= 0000, PART= 1
TAPE01 DUAL SL 1600 OFFLINE
      DEVICE ADDRESS = 004C
TAPE02 DUAL NL 1600
TAPESPEED 25 IPS
OFFLINE DEVICE ADDRESS = 0006

COMMAND(?): _
```

5. Enter **CD** to copy a data set.

```
COMMAND(?): CD
SOURCE (NAME,VOLUME): _
```

6. Enter the data set and volume name of the source data set.

```
SOURCE (NAME,VOLUME): FROM1,FRMVOL
TARGET (NAME,VOLUME): _
```

Storing and Handling Information

Copying Data on EDX Systems *(continued)*

7. Enter the data set and volume name of the target data set.

```
TARGET (NAME,VOLUME): TODATA,TOVOL  
ENTER SOURCE BLOCKSIZE (NULL=DISK(ETTE)): _
```

8. Enter the blocksize of the source data set if the data set is on tape. (Leave blank if the source data set is on a disk or diskette.)

```
ENTER SOURCE BLOCKSIZE (NULL=DISK(ETTE)): 256  
USE ATTN/CA TO CANCEL COPY
```

```
ARE ALL PARMS CORRECT? (Y,N): _
```

9. Did you enter all the parameters correctly?

- If so, enter a **Y**. (If you decide you want to cancel the copy operation after entering the **Y**, press the attention key, enter **CA**, and press the enter key.)
- If not, enter an **N** and continue with step 3 on page OP-168.

```
ARE ALL PARMS CORRECT? (Y,N): Y  
EOD ON SOURCE DATA SET  
DO YOU WANT TO CONTINUE: _
```

10. Do you want to continue and copy the next data set, merging it with the one just copied?

- If so, enter a **Y**.
- If not, enter an **N**.

```
EOD ON SOURCE DATA SET  
DO YOU WANT TO CONTINUE: N  
25 RECORDS COPIED
```

```
COMMAND (?): _
```

11. Enter **EN** to end \$TAPEUT1.

```
COMMAND (?): EN  
$TAPEUT1 ENDED AT 08:30:05
```

Copying Data to/from a Non-EDX System

Use the procedures in this section to copy data to/from a non-EDX system. Use the procedure that is compatible with the data format needed.

Copy From a Basic Exchange Data Set

Use this procedure to copy a basic exchange data set from a diskette to a disk data set.

Before You Start:

Observe the following requirements:

- Verify that the target (copy to) data set has been allocated. See “List One or More Data Sets on a Volume” on page OP-187 if you need to verify whether it exists, and “Allocate a Data Set” on page OP-124 if you need to allocate it.
- Obtain the name of the source (copy from) data set and diskette volume label. (\$DASDI names these \$\$EDXVOL and IBMEDX when it initializes the diskette, but they can be changed. See “List All Volumes on One Device” on page OP-183 to list the volume name.)
- Obtain data set and volume names of the target (copy to) data set.
- If you do not want \$COPY to start copying into the target data set at record number 1, determine the number of the record where you want it to start.

1. Load the utility \$COPY:

- Using the \$L operator command, press the attention key and enter \$L \$COPY.
- Using the session manager, select option 3.5.

```
> $L $COPY_  
LOADING $COPY      40P,08:39:20, LP= 0000,...  
$COPY - COPY UTILITY  
COMMAND(?): ? _
```

2. Enter RE to copy from the basic exchange diskette to disk.

```
COMMAND (?): RE  
SOURCE ($$EDXVOL,VOLUME): _
```

3. Enter the name of the source data set and diskette volume label.

```
SOURCE ($$EDXVOL,VOLUME): $$EDXVOL, IBMEDX  
TARGET (NAME,VOLUME): _
```

Storing and Handling Information

Copying Data to/from a Non-EDX System (*continued*)

4. Enter the name of the target data set and volume.

```
TARGET (NAME,VOLUME): MYDATA1,MYVOL
SPECIFY START/END (Y/N)? _
```

5. Do you want to specify a starting record in the target data set?

```
SPECIFY START/END (Y/N)? Y
FIRST RECORD: _
```

- If so, enter a Y and continue with step 6.
- If not, enter an N and continue with step 7.

6. Enter the number of the record in the target data set where you want \$COPY to start copying the contents of the basic exchange diskette.

```
FIRST RECORD: 9
ENTER BASIC EXCHANGE DATA SET NAME: _
```

7. Enter the name of the basic exchange data set on the diskette. This name was assigned when the data was copied to the diskette.

```
ENTER BASIC EXCHANGE DATA SET NAME: DATA
NUMBER OF RECORDS COPIED: 52
COPY COMPLETE

COMMAND (?): _
```

8. Do you want to copy another data set?

- If so, continue with step 2 on page OP-171.
- If not, enter EN to end \$COPY.

```
COMMAND (?): EN
$COPY ENDED AT 08:49:05
```

Copying Data to/from a Non-EDX System *(continued)*

Copy to a Basic Exchange Data Set

Use this procedure to copy a disk data set to a basic exchange diskette data set.

Before You Start:

Observe the following requirements:

- Obtain a one-sided diskette, formatted for IBM Standard for Information Interchange at 128-bytes per sector. If you do not have a diskette formatted to this standard, see “Initialize to IBM Standard for Information Interchange” on page OP-97 for the procedure to format one.
- Obtain the data set and volume names of the source (copy from) data set.
- If you are copying only part of the source data set, determine the numbers of the first and last records you are copying.
- Determine the name of the target diskette data set and diskette volume label. (\$DASDI names these \$\$EDXVOL and IBMEDX respectively, but they can be changed.)
- Determine the name you want to assign to the target data set.

1. Load the utility \$COPY:

- Using the \$L operator command, press the attention key and enter \$L \$COPY.
- Using the session manager, select option 3.5.

```
> $L $COPY_
LOADING $COPY      40P,08:39:20, LP=0000,...
$COPY - COPY UTILITY
COMMAND(?): ? _
```

2. Enter WE to copy a disk data set to a basic exchange diskette.

```
COMMAND(?): WE
SOURCE (NAME,VOLUME): _
```

3. Enter the name of the source data set and volume.

```
SOURCE (NAME,VOLUME): MYDATA1,MYVOL
SPECIFY START/END? _
```

Storing and Handling Information

Copying Data to/from a Non-EDX System (continued)

4. Do you want to copy only part of the data set?

- If so, enter a Y and continue with step 5.
- If not, enter an N and continue with step 7.

```
SPECIFY START/END? Y  
FIRST RECORD: _
```

5. Enter the number of the first record you are copying.

```
FIRST RECORD: 3  
LAST RECORD: _
```

6. Enter the number of the last record you are copying.

```
LAST RECORD: 10  
TARGET ($$EDXVOL,VOLUME): _
```

7. Enter the name of the target diskette volume and volume label. (\$DASDI names these \$\$EDXVOL and IBMEDX when it initializes the diskette.)

```
TARGET ($$EDXVOL,VOLUME): $$EDXVOL,IBMEDX  
ENTER BASIC EXCHANGE DATA SET NAME: _
```

8. Enter the name that you want assigned to the basic exchange data set on the diskette.

```
ENTER BASIC EXCHANGE DATA SET NAME: DATA  
COPY COMPLETE  
COMMAND (?): _
```

9. Do you want to copy another data set?

- If so, continue with step 2 on page OP-173.
- If not, enter EN to end \$COPY.

```
COMMAND (?): EN  
$COPY ENDED AT 08:49:05
```

Copying Data to/from a Non-EDX System *(continued)*

Copy from an H Exchange Data Set

Use this procedure to copy an H exchange data set into an EDX data set. If the EDX target data set does not exist, the utility \$HXUT1 allocates it for you.

Before You Start:

Observe the following:

- Obtain the volume label of the H exchange diskette. (This is set to IBMEDX when the diskette is initialized, but it can be changed. See “List All Volumes on One Device” on page OP-183 to determine the volume label name.)
- Determine the data set and volume names of the source data set.
- If you are copying only part of the data set, get the starting and ending record numbers.
- Determine the target data set and volume for the target data set. If they do not exist, \$HXUT1 will allocate them.
- Determine the record number in the target data set where you want \$HXUT1 to start copying.

1. Load the utility \$HXUT1:

- Using the \$L operator command, press the attention key and enter \$L \$HXUT1.
- Using the session manager, select option 3.11.

```
> $L $HXUT1_
$HXUT1      20P,01:01:20, LP= 9100, PART= 1
$HXUT1 - H EXCHANGE UTILITY
VOLUME NAME = _
```

2. Enter the volume name of the 2D diskette containing the H exchange data set. (\$DASDI names it IBMEDX when it initializes the diskette.)

```
VOLUME NAME = IBMEDX
COMMAND (?): _
```

3. Enter RX.

```
COMMAND (?): RX
SOURCE EXCHANGE DATA SET (NAME,VOLUME): _
```

Storing and Handling Information

Copying Data to/from a Non-EDX System *(continued)*

4. Enter the name of the H exchange diskette data set and volume.

```
SOURCE EXCHANGE DATA SET (NAME,VOLUME): MYHDS,HXVOL  
COPY ENTIRE DATA SET (Y/N): _
```

5. Do you want to copy the entire data set?

- If so, enter a Y and continue with step 9.
- If not, enter an N and continue with step 6.

```
COPY ENTIRE DATA SET (Y/N): N  
STARTING RECORD NUMBER: _
```

6. Enter the number of the first record you want copied.

```
STARTING RECORD NUMBER: 5  
COPY TO END OF DATASET? (Y/N):
```

7. Do you want to copy all the remainder of the data set?

- If so, enter a Y and continue with step 9.
- If not, enter an N and continue with step 8.

```
COPY TO END OF DATASET? (Y/N): N  
ENDING RECORD NUMBER: _
```

8. Enter the record number of the last record you want copied.

```
ENDING RECORD NUMBER: 100  
TARGET EDX DATASET (NAME,VOLUME): _
```

9. Enter the name and volume of the target EDX data set.

```
TARGET EDX DATASET (NAME,VOLUME): TOHXDS,TOHXVL  
STARTING RECORD NUMBER: _
```

Copying Data to/from a Non-EDX System *(continued)*

- 10.** If you are only copying part of the data set, enter the number of the record in the target EDX data set where you want the source data set to start.

```
STARTING RECORD NUMBER: 1
```

Are you copying a multiple-volume data set (the data set spans more than one H exchange diskette)?

- If so, continue with step 11.
- If not, continue with step 13.

- 11.** If you are copying a multiple-volume data set, enter the number of diskettes.

```
HOW MANY DISKETTES (1-99): 2
TOHXDS ALLOCATED, SIZE IS 7696 RECORDS
THIS IS A MULTIVOLUME COPY
SOURCE MAY NOT FIT INTO TARGET
PLEASE MOUNT DISKETTE WITH/FOR
NEXT SEGMENT (02) OF DATASET MYHDS
```

- 12.** Insert the next diskette. Continue inserting diskettes until the copy is complete.

```
COPY OF MYHXDS,HXVOL TO TOHXDS,TOHXVL COMPLETE
TOTAL NUMBER OF RECORDS COPIED WAS 7696
COMMAND (?): _
```

- 13.** Do you want to copy another H exchange data set?
- If so, continue with step 3 on page OP-175.
 - If not, enter **EN** to end \$HXUT1.

```
COMMAND (?): EN
$HXUT1 ENDED 08:30:05
```

Storing and Handling Information

Copying Data to/from a Non-EDX System *(continued)*

Copy to an H Exchange Diskette

Use this procedure to copy an EDX data set to an H exchange diskette data set. You can copy all or part of a data set, and you can merge data sets.

Note: Data on an H exchange data set must be copied back to an EDX format data set before it can be read by EDX programs.

Before You Start:

Observe the following requirements:

- Obtain a Diskette 2D that has been formatted for IBM Standards for Information Interchange and initialized as an H exchange diskette. See “Initialize to IBM Standard for Information Interchange” on page OP-97 if you need to format the diskette.
- Obtain the volume label name of the H exchange diskette. (When the diskette is initialized, \$DASDI assigns a volume label of IBMEDX, but it can be changed. See “List All Volumes on One Device” on page OP-183 to determine the volume label.)
- Obtain the data set and volume names of the _____ set. source EDX data
- If you are only copying part of the data set, determine the starting and ending record numbers of the portion you are copying.
- Obtain the data set and volume name of the target data set. \$HXUT1 allocates the H exchange data set if it does not already exist. If the source data set does not fit in the target volume, \$HXUT1 asks you for a new diskette volume.
- If you are merging data sets, get the record number in the target data set where you want \$HXUT1 to start copying.

1. Load the utility \$HXUT1:

- Using the \$L operator command, press the attention key and enter **\$L \$HXUT1**.
- Using the session manager, select option 3.11.

```
> $L $HXUT1_
LOADING $HXUT1 20P,01:01:20, LP= 9100,...
$HXUT1 - H EXCHANGE UTILITY
VOLUME NAME = _
```

2. Enter the volume name of the 2D diskette containing the H exchange data set.

```
VOLUME NAME = HVOL
COMMAND (?): _
```

Copying Data to/from a Non-EDX System *(continued)*

3. Enter WX.

COMMAND (?): WX

SOURCE EDX DATA SET (NAME,VOLUME): _

4. Enter the name of the EDX data set being copied.

SOURCE EDX DATA SET (NAME,VOLUME): MYHDS,MYVOL
COPY ENTIRE DATA SET (Y/N): _

5. Do you want to copy all the data set?

- If so, enter a Y and continue with step 9 on page OP-180.
- If not, enter an N and continue with step 6.

COPY ENTIRE DATA SET (Y/N): N
STARTING RECORD NUMBER: _

6. Enter the number of the first record you want copied.

STARTING RECORD NUMBER: 5
COPY TO END OF DATASET? (Y/N):

7. Do you want to copy the remainder of the data set?

- If so, enter a Y and continue with step 9 on page OP-180.
- If not, enter an N and continue with step 8.

COPY TO END OF DATASET? (Y/N): N
ENDING RECORD NUMBER: _

8. Enter the record number of the last record you want copied.

ENDING RECORD NUMBER: 100

TARGET EXCHANGE DATASET (NAME,VOLUME): _

Storing and Handling Information

Copying Data to/from a Non-EDX System (*continued*)

9. Enter the name and volume of the target EDX data set.

```
TARGET EXCHANGE DATASET (NAME,VOLUME): TOHXDS,TOHXVL
STARTING RECORD NUMBER: _
```

10. If you are copying only part of the data set, enter the number of the record in the EDX data set where you want \$HXUT1 to start copying to.

If you are copying a data set that fits on one diskette, continue with step 12.

```
STARTING RECORD NUMBER: 1
TOHXDS ALLOCATED, SIZE IS 100 RECORDS
THIS IS A MULTIVOLUME COPY
PLEASE MOUNT DISKETTE WITH/FOR
NEXT SEGMENT (02) OF DATA SET TOHXDS
VOLUME NAME: _
```

11. If you are copying a multiple-volume data set, insert the next diskette and enter the diskette volume name.

Repeat this step until you have copied the entire data set.

```
TOHXDS ALLOCATED, SIZE IS 100 RECORDS
COPY OF MYHDS,MYVOL TO TOHXDS,TOHXVL COMPLETE
TOTAL NUMBER OF RECORDS COPIED WAS 150
COMMAND (?): _
```

12. Do you want to copy another H exchange data set?

- If so, continue with step 3 on page OP-179.
- If not, enter EN to end \$HXUT1.

```
COMMAND (?): EN
$HXUT1 ENDED 08:30:05
```

Listing Volume and Data Set Directory Information

Use the procedures in this section to list the volume and data set directories on disk or diskette. The listings tell you the name of the volume or data set and information about it, such as its size and the location of the first record.

This section contains the following procedures:

- “List All Volumes”
- “List All Volumes on One Device” on page OP-183
- “List All Data Sets on All Volumes” on page OP-185
- “List One or More Data Sets on a Volume” on page OP-187
- “List Data- or Program-Type Data Sets on a Volume” on page OP-191

List All Volumes

Use this procedure to list the volume directories of all volumes on your system. The listing contains the following information about the volumes:

- Name
- Address of the device the volume is on
- Location of the first record

Before You Start:

Verify that the correct diskette is in the diskette unit if you are interested in specific volume names.

1. Load the utility \$DISKUT1:

- Using the \$L operator command, press the attention key and enter \$L \$DISKUT1.
- Using the session manager, select option 3.1.

```
> $L $DISKUT1
LOADING $DISKUT1 49P,08:04:28, LP= 0000,...
$DISKUT1 - DATA SET MANAGEMENT UTILITY I
USING VOLUME EDX002
COMMAND (?): _
```

Storing and Handling Information

Listing Volume and Data Set Directory Information (*continued*)

2. Do you want the \$DISKUT1 output displayed on the display screen, or printed a printer? \$DISKUT1 automatically displays the output on the display screen unless you tell it to print the output on the system printer (the printer named \$SYSPRTR), or some other specified printer.

- Skip to step 3 if you want \$DISKUT1 to display the output on the display terminal.
- Enter **LISTP** to tell \$DISKUT1 to print the output listings requested in step 3 on the system printer.
- Enter **LISTP printername** to tell \$DISKUT1 to print the output listings requested in step 3 on the specified printer.

```
COMMAND (?): LISTP MPRTR
```

```
COMMAND (?):
```

3. Do you want to list all logical volumes defined on your Series/1, or just those beginning with a specified prefix?

- Enter **LAV** to list all volumes defined.
- Enter **LAV xxxxxx**, where xxxxxx is a one to six character prefix, to list only those volumes whose names begin with the prefix.

In the example, if we had entered **LAV MY**, \$DISKUT1 would have listed only the last volume (MYVOL).

To cancel the listing while it is being printed or displayed, press the attention key and enter **CA**.

```
COMMAND (?): LAV
```

VOLUME NAME	DEV ADDR	FIRST REC	SIZE
EDX002	0003	661	10000
MTMSTR	0003	50301	1500
ASMLIB	0003	10661	5500
MYVOL	0003	48301	2000

```
USING VOLUME EDX002
```

```
COMMAND (?):
```

4. Is this all you wanted to list?

- If so, enter **EN** to end \$DISKUT1.
- If not, continue with 3.

```
COMMAND (?): EN
```

```
$DISKUT1 ENDED 08:10:30
```

Listing Volume and Data Set Directory Information (*continued*)

List All Volumes on One Device

Use this procedure to list the volume directories of the volumes on an individual disk or diskette device. The listing contains the following information about the volumes:

- Name
- Location of the first record
- Size
- Directory information.

Before You Start:

Determine the hexadecimal address of the disk or diskette device.

1. Load the utility \$INITDSK:

- Using the \$L operator command, press the attention key and enter \$L \$INITDSK.
- Using the session manager, select option 3.7.

```
> $L $INITDSK
LOADING $INITDSK 83P,08:04:28, LP= 9100,...
$INITDSK - DISK INITIALIZATION UTILITY
COMMAND (?): _
```

2. Enter LV to list the volume directory.

```
COMMAND (?): LV
DEVICE ADDRESS: _
```

3. Enter the hexadecimal device address.

The resulting listing shows the hexadecimal device address, the name of each volume on the disk or diskette, the location of the first record of each volume, and the number of 256-byte records in each volume (size). If a volume contains a copy of the supervisor that can be used to IPL the system, the name of the data set containing the supervisor is listed as shown in the example. If a volume is specified as a performance volume on a DISK configuration statement, that indication is also listed.

```
DEVICE ADDRESS: 3
DEVICE ADDRESS: 0003
VOLUME  FIRST REC.  SIZE  **IPL=$EDXNUC **
EDX002      121    10000
MTMSTR      49761    1500
ASMLIB      10121    5500
MYVOL       47761    2000
LIST DIRECTORY INFO (Y/N)? _
```

Storing and Handling Information

Listing Volume and Data Set Directory Information (*continued*)

4. Do you want to list the directory information or free space chain?

- If so, enter a Y.
- If not, enter an N and continue with step 6.

```
LIST DIRECTORY INFO (Y/N)? Y
DEVICE SIZE(RECORDS):      54000
DIRECTORY REC. NO:        541
DIRECTORY SIZE(RECS):      120
UNUSED DIRECTORY BYTES:    30416
NO. VOLUME ENTRIES:       8
MAX. VOLUME ENTRIES:      958
FREE SPACE ENTRIES:       2

DEVICE HAS NO FIXED HEADS
LIST FREE SPACE CHAIN (Y/N)? _
```

5. Do you want to list the free space chain? \$INITDSK asks this question only if the volume has free space available. If you are not asked this question, continue with step 6.

- If so, enter a Y.
- If not, enter an N.

In the accompanying drawing the number under the heading FIRST REC is the number of the first record of that portion of free space relative to the start of the volume. The number under the heading SIZE is the number of 256-byte records available.

```
LIST FREE SPACE CHAIN (Y/N)? Y
FIRST REC.      SIZE
  52761         700
  24121        21460
COMMAND (?): _
```

6. Is this all you wanted to list?

- If so, enter EN to end \$INITDSK.
- If not, continue with step 2 on page OP-183.

```
COMMAND (?): EN
$INITDSK ENDED 08:30:24
```

Listing Volume and Data Set Directory Information (*continued*)

List All Data Sets on All Volumes

Use this procedure to list data set directory information for all volumes on your system. The listing contains the following information:

- Name
- Data type (data or program)
- Location of the first record in the data set
- The number of 256-byte records in the data set
- The name of the volume the data set is in.

Before You Start:

Verify that the correct diskette is in the diskette unit if you are interested in specific data sets.

1. Load the utility \$DISKUT1:

- Using the \$L operator command, press the attention key and enter \$L \$DISKUT1.
- Using the session manager, select option 3.1.

```
> $L $DISKUT1
LOADING $DISKUT1 49P,08:04:28, LP= 0000,...
$DISKUT1 - DATA SET MANAGEMENT UTILITY I
USING VOLUME EDX002
COMMAND (?): _
```

2. Do you want the \$DISKUT1 output displayed on the display screen, or printed on a selected printer \$DISKUT1 automatically displays the output on the display screen unless you tell it to print the output on the system printer (printer named \$SYSPRTR), or another selected printer.

- Skip to step 3 on page OP-186 if you want \$DISKUT1 to display the output on the display terminal.
- Enter **LISTP** to tell \$DISKUT1 to print the output listings requested in step 3 on the system printer.
- Enter **LISTP printername** to tell \$DISKUT1 to print the output listings requested in step 3 on a selected printer.

```
COMMAND (?): LISTP MPRTR
COMMAND (?):
```

Storing and Handling Information

Listing Volume and Data Set Directory Information (*continued*)

3. Do you want to list all data sets or just those beginning with a specified prefix?

- To list all data sets on all volumes, enter **LAD**.
- To list only those whose names begin with the prefix, enter **LAD** **xxxxxxx**, where **xxxxxxx** is a one-to eight-character prefix.

Entering an 8-character data set name as a prefix lists the volume name and information about that data set only. This is useful if you don't know the volume name for a data set.

In the example, if we had entered **LAD MY**, **\$DISKUT1** would have listed only the data sets beginning with **MY**.

To cancel the listing while it is being printed or displayed, press the attention key, press the enter key and enter **CA**. (On a 3101, press the attention key and enter **CA**).

4. Is this all you wanted to list?

- If so, enter **EN** to end **\$DISKUT1**.
- If not, continue with 3.

```
COMMAND (?): LAD
NAME TYPE FIRST REC. SIZE EOD/PGMSZ VOLUME
$EDXNUC PGM 64 400 350 EDX002
$COPYUT1 PGM 464 57 57 EDX002
.
.
PROGRAM1 PGM 8 262 200 MYVOL
DATASET1 DATA 270 30 5 MYVOL
MYDATA1 DATA 300 30 28 MYVOL
MYPROG1 PGM 330 202 158 MYVOL
.
.
COMMAND (?):
```

```
COMMAND (?): EN
$DISKUT1 ENDED 08:30:24
```

Listing Volume and Data Set Directory Information *(continued)*

List One or More Data Sets on a Volume

Use this procedure to list the data set directory information for the data sets on an individual volume. You can get three different listings with this procedure. You can list the directory information for all data sets on a volume, for only one data set on a volume, or for all data sets in cylinder, track, sector (CTS), or relative block addressing (RBA) mode.

Cylinder, track, and sector (CTS) are physical divisions of a 4962 disk, 4963 disk, or a diskette recording surface and are used to access specific locations on it. Relative block addressing (RBA) is used on the 4967 disk device to access specific portions of its recording surface. Figure 12 shows the directory information listed for the three listing formats.

Information Listed	All data sets (LA)	One data set (LM)	CTS/RBA (LACTS)
Name (NAME)	X	X	X
Type (DATA, PGM, PGMOVL)	X	X	X
Location of first record (FIRST REC.)	X	X	
Number of 256-byte records (SIZE)	X	X	
Actual number of records with data (EOD/PGMSZ)	X	X	
Device address of the device containing the data set (IODA)		X	
Beginning and ending CTS or RBA			X

Figure 12. Data Set Directory Information

Storing and Handling Information

Listing Volume and Data Set Directory Information (*continued*)

Before You Start:

Observe the following requirements:

You need the name of the volume before starting this procedure.

If you are interested in the data sets on a particular diskette, insert the diskette into a diskette unit.

1. Load the utility \$DISKUT1:

- Using the \$L operator command, press the attention key and enter \$L \$DISKUT1.
- Using the session manager, select option 3.1.

```
> $L $DISKUT1
LOADING $DISKUT1 49P,08:04:28, LP= 0000,...
$DISKUT1 - DATA SET MANAGEMENT UTILITY I
USING VOLUME EDX002
COMMAND (?): _
```

2. Is \$DISKUT1 using the correct volume name?

- If so, continue with step 3 on page OP-189.
- If not, change the name of the volume by typing CV xxxxxx, where xxxxxx is the name of the volume you want to use.

```
USING VOLUME EDX002
COMMAND (?): CV MYVOL
USING VOLUME MYVOL
COMMAND (?): _
```

Listing Volume and Data Set Directory Information *(continued)*

3. Do you want the \$DISKUT1 output displayed on the display screen, or printed on a selected printer? \$DISKUT1 automatically displays the output on the display screen unless you tell it to print the output on the system printer (the printer named \$SYSPRTR) or another selected printer.

- Skip to step 4 on page OP-190 if you want \$DISKUT1 to display the output on the display terminal.
- Enter **LISTP** to tell \$DISKUT1 to print the output listings requested in step 4 on page OP-190 on the system printer.
- Enter **LISTP printername** to tell \$DISKUT1 to print the output listings requested in step 4 on page OP-190 on the selected printer.

```
COMMAND (?): LISTP MPRTR
```

```
COMMAND (?):
```

Storing and Handling Information

Listing Volume and Data Set Directory Information (*continued*)

4. Enter the appropriate command for the desired directory listing:

- Enter **LM** to list the directory information for only one data set. Continue with step 5 on page OP-191.

```
COMMAND (?): LM
MEMBER NAME:
```

- Enter **LA** to list all data sets on the volume. Skip to step 6 on page OP-191.
- Enter **LA xxxxxxxx**, where xxxxxxxx is a one to eight character prefix, to list only those whose names begin with the prefix. Skip to step 6 on page OP-191. (In the example, if we had entered **LA MYDA**, \$DISKUT1 would have listed only two data sets - MYDATA1 and MYDATA2.)

To cancel the listing while it is being printed or displayed, press the attention key, press the enter key and enter **CA**. (On a 3101, press the attention key and enter **CA**.)

```
COMMAND (?): LA

USING VOLUME MYVOL
NAME TYPE FIRST REC. SIZE EOD/PGMSZ
PROGRAM1 PGM 8 262 200
DATASET1 DATA 270 30 5
MYDATA1 DATA 300 30 28
MYPROG1 PGM 330 202 158
MYDATA2 DATA 532 200 50
MYPROG2 PGM 732 360 360
```

909 FREE RECORDS IN LIBRARY

```
COMMAND (?):
```

- Enter **LACTS** to list CTS/RBA information for all data sets on the volume. Skip to step 6 on page OP-191.
- Enter **LACTS xxxxxxxx**, where xxxxxxxx is a one to eight character prefix as described above. Skip to step 6 on page OP-191.

```
COMMAND (?): LACTS

USING VOLUME MYVOL
NAME TYPE ORG(CTS) END(CTS)
PROGRAM1 PGM 0030028 0050107
DATASET1 DATA 0050108 0050123
MYDATA1 DATA 0050124 0050139
MYPROG1 PGM 0050140 0050147
MYDATA2 DATA 0050148 0050202
MYPROG2 PGM 0050203 0050215
```

909 FREE RECORDS IN LIBRARY

```
COMMAND (?):
```

Listing Volume and Data Set Directory Information (*continued*)

5. Enter the name of the data set. (This step only applies to listing directory information for a single data set.)

```
MEMBER NAME: PROGRAM1
USING VOLUME MYVOL

NAME TYPE FIRST REC. SIZE EOD/PGMSZ
PROGRAM1 PGM 8 262 200
IODA= 0003 (HEX), CTS= 2680107,2690228

COMMAND (?):
```

6. Is this all you wanted to list?

- If so, enter **EN** to end \$DISKUT1.
- If not, continue with 2 on page OP-188 or 4 on page OP-190.

```
COMMAND (?): EN
$DISKUT1 ENDED 08:30:24
```

List Data- or Program-Type Data Sets on a Volume

Use this procedure to list the directory information for either data- or program-type data sets on a volume. Depending on the type of listing you request, it contains either the name, location of the first record, and the size of each data set, or the name, and the beginning and ending CTS or RBA addresses of each data set. CTS stands for cylinder, track, and sector. Cylinder, track, and sector are physical divisions of a 4962 disk, 4963 disk, or a diskette recording surface and are used to access specific locations on it. RBA stands for relative block addressing. Relative block addressing is used on the 4967 disk device to access specific portions of its recording surface.

Before You Start

Determine the name of the volume you wish to list.

1. Load the utility \$DISKUT1:
 - Using the \$L operator command, press the attention key and enter **\$L \$DISKUT1**.
 - Using the session manager, select option 3.1.

```
> $L $DISKUT1
LOADING $DISKUT1 49P,08:04:28, LP= 0000,...
$DISKUT1 - DATA SET MANAGEMENT UTILITY I
USING VOLUME EDX002
COMMAND (?): _
```

Storing and Handling Information

Listing Volume and Data Set Directory Information (*continued*)

2. Is \$DISKUT1 using the correct volume name?

- If so, continue with step 3.
- If not, change the name of the volume by typing **CV xxxxxx**, where xxxxxx is the name of the volume you want to use. In the example we changed the volume name to MYVOL.

```
USING VOLUME EDX002
```

```
COMMAND (?): CV MYVOL
```

```
USING VOLUME MYVOL
```

```
COMMAND (?): _
```

3. Do you want the \$DISKUT1 output displayed on the display screen, or printed on a specified printer? \$DISKUT1 automatically displays the output on the display screen unless you tell it to print the output on the system printer (the printer named \$SYSPRTR), or another specified printer.

- Skip to step 4 if you want \$DISKUT1 to display the output on the display terminal.
- Enter **LISTP** to tell \$DISKUT1 to print the output listings requested in step 4 on the system printer.
- Enter **LISTP printername** to tell \$DISKUT1 to print the output listings requested in step 4 on the specified printer.

```
COMMAND (?): LISTP MPRTR
```

```
COMMAND (?):
```

4. Do you want the list in standard format, or CTS/RBA format?

- If you want the list in the standard format, continue with step 4(a).
- If you want the list in CTS or RBA format, continue with step 4(b).

To cancel the listing while it is being printed or displayed, press the attention key, press the enter key and enter CA. (On a 3101, press the attention key and enter CA.)

Listing Volume and Data Set Directory Information (*continued*)

4(a). Enter one of the following commands to generate a standard format data set directory listing of only the data- or program-type data sets on the volume.

- Enter **LD** to list all data-type data sets on the volume. Continue with step 5.
- Enter **LD xxxxxxxx**, where xxxxxxxx is a one to eight character prefix, to list only the data-type data sets whose names begin with the prefix. (In the example, if we had entered **LD MY**, \$DISKUT1 would have listed only the last two data sets - MYDATA1 and MYDATA2.) Continue with step 5.
- Enter **LP** to list all program-type data sets on the volume. Continue with step 5.
- Enter **LP xxxxxxxx**, where xxxxxxxx is a one to eight character prefix, to list only the program-type data sets whose names begin with the prefix. Continue with step 5.

```
COMMAND (?): LD
```

NAME	FIRST REC.	SIZE	EOD/PGMSZ
DATASET1	270	30	5
MYDATA1	300	30	28
MYDATA2	532	200	190

```
909 FREE RECORDS IN LIBRARY
```

```
COMMAND (?):
```

Storing and Handling Information

Listing Volume and Data Set Directory Information (*continued*)

4(b). Enter one of the following commands to generate a CTS/RBA format data set directory listing of only the data- or program-type data sets on the volume. The listing reflects the format appropriate for the device.

- Enter **LDCTS** to list all data-type data sets on the volume.
- Enter **LDCTS xxxxxxxx**, where xxxxxxxx is a one to eight character prefix, to list only the data-type data sets whose names begin with the prefix. (In the example, if we had entered **LDCTS MY**, \$DISKUT1 would have listed only the last two data sets - MYDATA1 and MYDATA2.)
- Enter **LPCTS** to list all program-type data sets on the volume.
- Enter **LPCTS xxxxxxxx**, where xxxxxxxx is a one to eight character prefix, to list only the program-type data sets whose names begin with the prefix.

```
COMMAND (?): LDCTS

USING VOLUME MYVOL
NAME      ORG(CTS)  END(CTS)
DATASET1  2690229  2690258
MYDATA1   2690259  2700028
MYDATA2   2710051  2720110

          909 FREE RECORDS IN LIBRARY

COMMAND (?):
```

5. Is this all you wanted to list?

- If so, enter **EN** to end \$DISKUT1.
- If not, continue with step 2 or step 4.

```
COMMAND (?): EN

$DISKUT1 ENDED 08:30:24
```

Listing the Contents of a Data Set

Use one of the following three procedures to get a source listing or a formatted dump of a data set. The output can be listed on a display terminal or printed on a printer.

- Use the procedure “List the Contents of a Data-Type Data Set” to get a source format listing of the contents of a data-type data set.
- Use the procedure “Dump the Contents of a Data-Type Data Set” on page OP-198 to get a decimal, EBCDIC, or hexadecimal format listing of the contents of a data-type data set.
- Use the procedure “Dump the Contents of a Data-Type Data Set” on page OP-198 to get a decimal, EBCDIC, or hexadecimal format listing of the contents of a program-type data set.

List the Contents of a Data-Type Data Set

Use this procedure to get a source listing of the contents of a data-type data set. The output can be listed on either a display terminal, or a printer.

Before You Start:

Determine the following:

- The data set and volume names of the data set being listed.
- The name of the display terminal or printer where the output is to be listed. This name is not needed if you want the output listed on the display terminal you are using or printed on the system printer named \$SYSPRTR. (See “Listing System Terminal Assignments” on page OP-44 to get the names of the display terminals and printers on your system.)
- The decimal number of the first and last records of the portion you are listing if you are listing only part of the data set. These numbers are relative to the start of the data set. The valid range is 1 through the number of source records in the data set. See “Initialize a Diskette Volume Directory” on page OP-106 to determine how many 256-byte records are in the data set (EOD/PGMSZ column). Multiply this figure by 2 to get the number of source records in the data set. If you do enter a number that is too large, \$DISKUT2 returns the following error message telling you what the valid range is for the data set:

```
INVALID NUMBER  
MUST BE IN RANGE OF 1 TO nn
```

\$DISKUT1 also allows you to use absolute record numbers. See the *Operator Commands and Utilities Reference* for an explanation of using absolute record numbers instead of record numbers relative to the start of the data set.

Storing and Handling Information

Listing the Contents of a Data Set *(continued)*

1. Load the utility \$DISKUT2:

- Using the \$L operator command, press the attention key and enter **\$L \$DISKUT2**.
- Using the session manager, select option 3.2.

```
> $L $DISKUT2
LOADING $DISKUT2 51P,08:01:20, LP=0000,...

$DISKUT2 - DATA SET MANAGEMENT UTILITY II
USING VOLUME EDX002
COMMAND (?): _
```

2. Is \$DISKUT2 using the correct volume name?

- If so, continue with step 3.
- If not, change the name of the volume by entering **CV xxxxxx**, where xxxxxx is the name of the volume containing the data set you are listing.

```
COMMAND (?): CV MYVOL
USING VOLUME MYVOL
COMMAND (?): _
```

3. Enter the appropriate command for the type of listing or dump you are doing:

- Enter **LP dsname termname** to list the contents of a data set on a printer or display terminal. **Dsname** is the name of the data set you are listing, and **termname** is the symbolic name of the printer or display terminal where you want the listing printed. If you do not enter a terminal name, \$DISKUT2 prints the listing on the system printer (\$SYSPRTR).
- Enter **LU dsname termname** to list the contents of a data set on the display terminal. If you do not enter a display terminal name, \$DISKUT2 displays the listing on the terminal that you are using (the terminal that loaded \$DISKUT2).

```
COMMAND(?): LU MYSRC TERM2
LIST ALL OF THE DATA SET? _
```

To cancel the list while it is being printed or displayed, press the attention key, press the enter key, and enter **CA** (On the 3101, press the attention key and enter **CA**).

Listing the Contents of a Data Set *(continued)*

4. Do you want to list the entire data set, or just part of it?

- If so, enter a **Y** and continue with step 7.
- If not, enter an **N**.

```
LIST ALL OF THE DATA SET? N
FIRST RECORD: _
```

5. Enter the decimal record number of the first source record you are listing.

```
FIRST RECORD: 5
LAST RECORD: _
```

6. Enter the decimal record number of the last source record you are listing.

```
LAST RECORD: 9
READA EQU *
      GETVALUE A, 'A = ', DWORD
      SPACE 1
READB EQU *
      GETVALUE B, 'B = ', DWORD

LIST COMPLETE
COMMAND (?): _
```

7. Do you want to list another portion of this or another data set?

- If so, continue with step 2 on page OP-196.
- If not, enter **EN**.

```
COMMAND (?): EN
$DISKUT2 ENDED AT 08:35:20
```

Storing and Handling Information

Listing the Contents of a Data Set (*continued*)

Dump the Contents of a Data-Type Data Set

Use this procedure to get a formatted dump of the contents of a data-type data set. The listing can be in either decimal, EBCDIC, or hexadecimal format and the output can be listed on either a display terminal or a printer.

Before You Start:

Determine the following:

- The data set and volume names of the data set being listed.
- The name of the display terminal or printer where the output is to be listed. This name is not needed if you want the output listed on the display terminal you are using or printed on the system printer named \$SYSPRTR. (See "Listing System Terminal Assignments" on page OP-44 to get the names of the display terminals and printers on your system.)
- The decimal number of the first 256-byte record you are listing. The number is relative to the start of the data set. Valid numbers are from 1 to the number of records in the data set. (See "List Data- or Program-Type Data Sets on a Volume" on page OP-191 to determine how many records are in the data set. Use the number in the SIZE column.)

\$DISKUT2 also allows you to use absolute record numbers. See the *Operator Commands and Utilities Reference* for an explanation of using absolute record numbers instead of record numbers relative to the start of the data set.

- The decimal number of the last record of the portion being listed. The number is relative to the start of the data set. Valid numbers are from the number of the first record being listed, to the number of the last record in the data set.
- The decimal number of the first word in the first record being listed. Valid numbers are 1-128.
- The decimal number of words that you want dumped from each record. The valid numbers are 1-128.
- The format you want the listing printed in. It can be either decimal, EBCDIC, or hexadecimal.

Listing the Contents of a Data Set *(continued)*

1. Load the utility \$DISKUT2:
 - Using the \$L operator command, press the attention key and enter \$L \$DISKUT2.
 - Using the session manager, select option 3.2.

```
> $L $DISKUT2
LOADING $DISKUT2 51P,08:01:20, LP=0000,...

$DISKUT2 - DATA SET MANAGEMENT UTILITY II
USING VOLUME EDX002
COMMAND (?): _
```

2. Is \$DISKUT2 using the correct volume name?
 - If so, continue with step 3.
 - If not, change the name of the volume by entering CV xxxxxx, where xxxxxx is the name of the volume containing the data set you are listing.

```
COMMAND (?): CV MYVOL
USING VOLUME MYVOL
COMMAND (?): _
```

3. Enter the appropriate command for the type of listing or dump you are doing:
 - Enter DP **dsname termname** to print the contents of a data set on a printer or display terminal. **Dsname** is the name of the data set you are listing, and **termname** is the symbolic name of the printer or display terminal where you want the listing printed. If you do not enter a terminal name, \$DISKUT2 prints the listing on the system printer (\$SYSRTR).
 - Enter DU **dsname** to list the contents of a data set on the display terminal. **Dsname** is the name of the data set you are listing. If you do not enter a display terminal name, \$DISKUT2 displays the listing on the terminal that you are using (the terminal that loaded \$DISKUT2).

```
COMMAND(?): DU
PGM OR DS NAME: _
```

To cancel the list while it is being printed or displayed, press the attention key, press the enter key, and enter CA (On the 3101, press the attention key and enter CA).

Storing and Handling Information

Listing the Contents of a Data Set *(continued)*

4. Enter the name of the data set.

```
PGM OR DS NAME: MYSRC  
MYSRC IS A DATA SET  
FIRST RECORD (0 TO CANCEL COMMAND): _
```

5. Enter the decimal number of the first 256-byte record you want printed or a 0 to cancel the command. The valid record numbers are 1 to the number of the last record in the data set.

```
FIRST RECORD (0 TO CANCEL COMMAND): 3  
LAST RECORD: _
```

If you enter a 0, return to step 2 on page OP-199 to start over or skip to step 11 on page OP-201 to end \$DISKUT2.

6. Enter the decimal record number of the last record you are listing. The valid numbers are from the number entered in step 5, to the number of the last record in the data set.

```
LAST RECORD: 5  
FIRST WORD: _
```

7. Enter the decimal number of the first word that you are printing from the first record. The valid numbers are from 1 to 128.

```
FIRST WORD: 1  
WORDS / RECORD: _
```

8. Enter the decimal number of words you want printed from each record. The valid numbers are from 1 to 128.

```
WORDS / RECORD: 10  
(D)EC, (E)BCDIC OR (H)EX? _
```

Listing the Contents of a Data Set *(continued)*

9. Enter the letter for the desired format:
- Enter a **D** for a decimal format listing.
 - Enter an **E** for an EBCDIC format listing.
 - Enter an **H** for a hexadecimal format listing.

(D)EC, (E)BCDIC OR (H)EX? E

```
RECORD      3
  1 D9C5 C1C4 C140 4040 40C5 D8E4 4040 4040 | READA   EQU |
  9 4040 405C |      *   |
RECORD      4
  1 4040 4040 4040 4040 40E2 D7C1 C3C5 40F1 |      SPACE 1 |
  9 4040 4040 |      |
RECORD      5
  1 4040 4040 4040 4040 40C7 C5E3 E5C1 D3E4 |      GETVALU |
  9 C540 40C2 |      E B   |
```

DUMP COMPLETE
ANOTHER AREA? _

10. Do you want to list another portion of this data set?

- If so, enter a **Y** and continue with step 5 on page OP-200.
- If not, enter an **N**.

ANOTHER AREA? N
COMMAND (?):

11. Do you want to list another portion of this or another data set?

- If so, continue with step 2 on page OP-199.
- If not, enter **EN**.

COMMAND (?): EN
\$DISKUT2 ENDED AT 08:35:20

Storing and Handling Information

Listing the Contents of a Data Set *(continued)*

Dump the Contents of a Program-Type Data Set

Use this procedure to list the contents of a program-type data set on either a display terminal or a printer. The listing can be in either decimal, EBCDIC, or hexadecimal format.

Before You Start:

Determine the following:

- The data set and volume names of the data set being listed.
- The name of the display terminal or printer where the output is to be listed. This name is not needed if you want the output listed on the display terminal you are using or printed on the system printer named \$SYSPRTR. (See "Listing System Terminal Assignments" on page OP-44 to get the names of the display terminals and printers on your system.)
- The hexadecimal address where you want the dump to start. Valid numbers are from 0 to the hexadecimal size of the data set. (\$DISKUT2 tells you the hexadecimal size of the data set when you enter the data set name in step 4.)
- The decimal number of words you want dumped if you are not dumping the entire data set. Valid numbers are from 0 to the number of words in the data set. If you enter a number that is too large, \$DISKUT2 returns the following error message telling you the actual number of records in the data set:

```
INVALID NUMBER  
MUST BE IN RANGE OF 0 TO nn
```

- The format in which you want the listing printed. It can be either decimal, EBCDIC, or hexadecimal.

1. Load the utility \$DISKUT2:

- Using the \$L operator command, press the attention key and enter \$L \$DISKUT2.
- Using the session manager, select option 3.2.

```
> $L $DISKUT2  
LOADING $DISKUT2 51P,08:01:20, LP=0000,...  
  
$DISKUT2 - DATA SET MANAGEMENT UTILITY 11  
USING VOLUME EDX002  
COMMAND (?): -
```

Listing the Contents of a Data Set (*continued*)

2. Is \$DISKUT2 using the correct volume name?

- If so, continue with step 3.
- If not, change the name of the volume by entering **CV xxxxxx**, where xxxxxx is the name of the volume containing the data set you are listing.

```
COMMAND (?): CV MYVOL
USING VOLUME MYVOL
COMMAND (?): _
```

3. Enter the appropriate command for the type of listing or dump you are doing:

- Enter **DP dsname termname** to print the contents of a data set on a printer or display terminal. **Dsname** is the name of the data set you are listing, and **termname** is the symbolic name of the printer where you want the listing printed. If you do not enter a terminal name, \$DISKUT2 prints the listing on the system printer (\$SYSPRTR).
- Enter **DU dsname** to list the contents of a data set on the display terminal. **Dsname** is the name of the data set you are listing. If you do not enter a display terminal name, \$DISKUT2 displays the listing on the terminal that you are using (the terminal that loaded \$DISKUT2).

```
COMMAND(?): DU
PGM OR DS NAME: _
```

To cancel the list while it is being printed or displayed, press the attention key, press the enter key, and enter **CA** (On the 3101, press the attention key and enter **CA**).

4. Enter the name of the program.

```
PGM OR DS NAME: MYPGM
MYPGM IS A PROGRAM OF HEX SIZE 000002BE
ADDRESS: _
```

Storing and Handling Information

Listing the Contents of a Data Set (continued)

5. Enter the hexadecimal address where you want the dump to start. The valid addresses are 0 to the last address in the data set.

```
ADDRESS: 80
DUMP TO PROGRAM END? _
```

6. Do you want to dump to the end of the program?

- If so, enter **Y** to print the rest of the data set.
- If not, enter an **N** to print only part of the data set.

```
DUMP TO PROGRAM END? N
HOW MANY WORDS ? _
```

7. Enter the decimal number of words that you are printing. The valid numbers are from 0 to the number of words in the data set.

```
HOW MANY WORDS ? 30
```

```
0080 7DE2 E3D6 D77D 7C40 0018 01A2 80A2 01A2 | 'STOP'@ ... .. |
0090 FFFF 018A 802D 018E 009E 0001 0605 7CC1 | .....@A |
00A0 407E 4040 802D 0192 00AE 0001 0404 C240 | = ... ..B |
00B0 7E40 8026 0A09 7CC1 404E 40C2 | = ...@A + B |
```

```
DUMP COMPLETE
ANOTHER AREA? _
```

8. Do you want to list another portion of this data set?

- If so, enter a **Y** and continue with step 5 .
- If not, enter an **N**.

```
ANOTHER AREA? N
COMMAND (?):
```

9. Do you want to list another portion of this or another data set?

- If so, continue with step 2 on page OP-203.
- If not, enter **EN**.

```
COMMAND (?): EN
$DISKUT2 ENDED AT 08:35:20
```

Determining Volume Size and Free Space

Use this procedure to list the following information about an individual volume.

- The amount of free space.
- The location of the free space.
- The free space chain. The free space chain shows you the first record location and size of each portion of available free space.

Before You Start:

Observe the following requirements:

- If the volume is on a diskette, insert the diskette in a diskette unit.
- Get the name of the volume.

1. Load the utility \$DISKUT1:

- Using the \$L operator command, press the attention key and enter \$L \$DISKUT1.
- Using the session manager, select option 3.1.

```
> $L $DISKUT1
LOADING $DISKUT1 49P,08:04:28, LP= 0000,...
$DISKUT1 - DATA SET MANAGEMENT UTILITY I
USING VOLUME EDX002
COMMAND (?): _
```

2. Is \$DISKUT1 using the correct volume name?

- If so, continue with step 3 on page OP-206.
- If not, change the name of the volume by typing CV xxxx, where xxxx is the name of the volume you want to use.

```
USING VOLUME EDX002
COMMAND (?): CV MYVOL
USING VOLUME MYVOL
COMMAND (?): _
```

Storing and Handling Information

Determining Volume Size and Free Space *(continued)*

3. Enter **LS** to list the amount of available free space.

```
COMMAND (?): LS
USING VOLUME MYVOL

LIBRARY
  AT RECORD      48301
  SIZE           2000 RECORDS
  UNUSED         909 RECORDS

DIRECTORY
  SIZE           54 MEMBERS ( 7 RECORDS)
  UNUSED         48 MEMBERS

NO. MEMBERS -      6

NO. FREE SPACE ENTRIES - 1

LIST FREE SPACE CHAIN?
```

4. Do you want to list the free space chain? (This tells you the locations of the free space and the size of each portion. In our example volume, all the free space is in one piece.)

- If so, enter a **Y**.
- If not, enter an **N**.

```
LIST FREE SPACE CHAIN? Y

FIRST REC.      SIZE
1092            909

COMMAND (?):
```

5. Do you want to list the free space in another volume?

- If so, continue with step 2 on page OP-205.
- If not, enter **EN** to end \$DISKUT1.

```
COMMAND (?): EN

$DISKUT1 ENDED 08:06:30
```

Creating a Stand-Alone or \$TRAP Dump Diskette

Use this procedure to create stand-alone or \$TRAP dump diskettes. These diskettes are used to make a copy (dump) of the contents of the Series/1 storage. They are usually used as part of a problem determination procedure.

You need one diskette to dump the storage of a processor containing 512K of storage or less. You need two diskettes to dump the storage of a processor containing more than 512K of storage. If you are preparing to take a dump of a processor with more than 512K of storage, both diskettes must be prepared using the \$DASDI utility Option 0 as described in this procedure.

See “Taking a Stand-Alone Dump of Storage” on page OP-290 and “Printing a Stand-Alone or Formatted \$TRAP Dump” on page OP-293 for the procedures for using stand-alone and \$TRAP dump diskettes.

Before You Start:

Determine the following:

- The number of two-sided diskettes that you want to format. You do not have to use new diskettes. However, any data on the diskettes will be lost. (The utility used to format these diskettes allows you to format up to two diskettes without reloading the utility.)
- Insert one of the diskettes into a diskette drive. If you are using a 4966 Magazine Unit, the diskette must be in slot number 1, 2, or 3.
- The hexadecimal address of the diskette unit. See “Listing the Devices on Your System” on page OP-318 for the procedure for obtaining this number.

1. Load the utility \$DASDI:

- Using the \$L operator command, press the attention key and enter \$L \$DASDI.
- Using the session manager, select option 3.6.

```
> $L $DASDI
LOADING $DASDI      7P,08:30:06, LP= 0000, PART=2
DIRECT ACCESS DEVICE INITIALIZATION
DISK INITIALIZATION OPTIONS:
0=CREATE STAND-ALONE DUMP/$TRAP DISKETTE
1=DISKETTE INITIALIZATION
2=4962 DISK INITIALIZATION
3=4963 DISK INITIALIZATION
4=4967 DISK INITIALIZATION
5=DDSK DISK INITIALIZATION
9=EXIT DISK INITIALIZATION
ENTER DISK INITIALIZATION OPTION: _
```

2. Enter 0 to create a dump diskette.

```
ENTER DISK INITIALIZATION OPTION: 0
DO YOU WISH TO FORMAT 2 DISKETTES (Y/N)? _
```

Storing and Handling Information

Creating a Stand-Alone or \$TRAP Dump Diskette (*continued*)

3. Do you want to format 2 diskettes without reloading \$DASDI?

- If so, enter a Y.
- If not, enter an N.

```
DO YOU WISH TO FORMAT 2 DISKETTES (Y/N)? Y
DISKETTE INITIALIZATION
DISKETTE TO BE FORMATTED FOR USE WITH: STAND-ALONE DUMP
IS DISKETTE A 2-SIDED DISKETTE (Y/N)? _
```

4. Is there a 2-sided diskette in the diskette unit?

- If so, enter a Y.
- If not, insert a 2-sided diskette and enter Y. (If you do not have a 2-sided diskette available, enter an N and skip to step 8 on page OP-209 to end the utility.)

```
IS DISKETTE A 2-SIDED DISKETTE (Y/N)? Y
*****
* DISKETTE FORMATTING PROGRAM *
* IF FORMATTING IS IN PROGRESS, DO NOT *
* CANCEL ($C) THIS PROGRAM INSTEAD, PRESS *
* ATTN AND ENTER $DASDI TO TERMINATE. *
*****
ENTER DISKETTE ADDRESS IN HEX: _
```

5. Enter the hexadecimal address of the diskette device you are using. In our example, we used device address 02.

```
ENTER DISKETTE ADDRESS IN HEX: 02
DEVICE VARIED OFFLINE
DEVICE ADDRESS 02, STAND-ALONE DUMP FORMAT, SINGLE DENSITY,
256 BYTES/SECTOR
WARNING: FORMATTING WILL DESTROY ALL DATA ON THE DISKETTE.
CONTINUE (Y/N)? _
```

6. Do you have the correct diskette in the diskette unit?

- If so, enter a Y.
- If not, insert the correct diskette and enter a Y. (If you do not have a valid diskette available, enter an N and skip to step 8 on page OP-209 to end the utility.)

```
FORMATting WILL DESTROY ALL DATA ON THE DISKETTE.
CONTINUE (Y/N)? Y
IBMEDX VARIED ONLINE
```

Creating a Stand-Alone or \$TRAP Dump Diskette (*continued*)

7. If this is the second diskette, or if you entered an N in step 3 on page OP-208, skip to step 8.

If you entered a Y in step 3 on page OP-208:

- To format the second diskette; remove the first diskette, insert the second one, close the diskette drive door, and press the enter key. Return to step 4 on page OP-208.
- To end the formatting after completing the first diskette, press key PF3.

8. Do you want to initialize more dump diskettes?

- If so, continue with step 2 on page OP-207.
- If not, enter a 9 to end \$DASDI.

```
FORMATTING COMPLETE
DUMP DISKETTE BUILT
```

```
*** REMOVE THE FIRST DISKETTE AND INSERT SECOND DISKETTE ***
```

```
PRESS "ENTER" TO CONTINUE OR "PF3" TO EXIT OPTION
```

```
FORMATTING COMPLETE
DUMP DISKETTE BUILT
```

```
DIRECT ACCESS DEVICE INITIALIZATION
DISK INITIALIZATION OPTIONS:
0=CREATE STAND-ALONE DUMP/$TRAP DISKETTE
1=DISKETTE INITIALIZATION
2=4962 DISK INITIALIZATION
3=4963 DISK INITIALIZATION
4=4967 DISK INITIALIZATION
5=DDSK DISK INITIALIZATION
9=EXIT DISK INITIALIZATION
ENTER DISK INITIALIZATION OPTION: 9
```

```
$DASDI ENDED 08:05:30
```

Storing and Handling Information

Entering/Editing Data with \$FSEDIT

Use the following procedure to enter or edit the contents of a data set with the full-screen editor \$FSEDIT. It shows you how to load \$FSEDIT, read a data set into the work data set, save the updated work data set, and end \$FSEDIT.

After \$FSEDIT is loaded, the program function (PF) keys have special meanings. \$FSEDIT also uses a set of editing commands. “\$FSEDIT PF Keys and Edit Commands” on page OP-353 contains tables showing the meanings of the PFkeys and the editing commands. (Refer to the *Operator Commands and Utilities Reference* for a more detailed explanation of \$FSEDIT.)

Before You Start:

Observe the following requirements:

- If you are using the \$L command to load \$FSEDIT, you need the data set and volume name of a previously allocated work data set.

Note: If you are loading \$FSEDIT from an application program or a \$JOBUTIL procedure, you must pass \$FSEDIT a 1-word parameter of zeroes. \$FSEDIT copies the data set being edited into a work data set. This protects the integrity of the original data until you are ready to replace it with the new data. The work data set must be large enough to hold the data set you are creating or editing.

- If you are using the session manager to load \$FSEDIT, the session manager uses one of its work data sets for the \$FSEDIT work data set.
- If you are creating a new data set, you need the data set and volume names where you will store the data.
- If you are editing an existing data set, you need the data set and volume names of the existing data set.

Notes:

1. The \$FSEDIT menus contain a lot of information and the full menu is repeated on your screen for almost every operation. The examples in the following procedure show only the portion of the menu that is pertinent to the current step.
2. The \$FSEDIT utility uses a data set as a work area. During an edit session, the system saves the information you entered on the screen in a buffer area. Once this buffer area fills up, the system saves the contents in the work data set. If you IPL your system before you save the contents of the buffer in the work data set, the contents are lost. To avoid losing information in the buffer area, use option 4 (WRITE) before IPLING your system. If you write the work data set to a tape, previously existing data sets in that tape are no longer accessible.

Entering/Editing Data with \$FSEDIT (*continued*)

1. Load the utility \$FSEDIT:

- Using the \$L operator command, press the attention key and enter \$L \$FSEDIT *dsname*,*volname*, where *dsname* is the work data set name, and *volname* is the name of the volume it is on.
- Using the session manager, select option 1. (The session manager allocates a work data set for you.)

```
> $L $FSEDIT MYWORKDS,MYVOL
```

```
DS1 HAS NOT PREVIOUSLY BEEN USED AS A WORK DATA SET.  
IS IT OK TO USE IT NOW? _
```

- ### 2. If this is the first time that the work data set has been used, \$FSEDIT verifies if it is OK to use it. If you do not receive this question, skip to step 3 on page OP-212. (If you enter an N, \$FSEDIT ends.)

```
DS1 HAS NOT PREVIOUSLY BEEN USED AS A WORK DATA SET.  
IS IT OK TO USE IT NOW? Y
```

```
$FSEDIT PRIMARY OPTION MENU -----STATUS =  
PRESS PF3 TO EXIT
```

```
OPTION ==> _
```

```
DATASET NAME =====> (CURRENTLY IN WORK FILE)  
VOLUME NAME =====>
```

```
HOST DATASET =====>
```

```
ENTER A VOLUME NAME AND PRESS ENTER FOR A DIRECTORY LIST
```

- 1 BROWSE
- 2 EDIT
- 3 READ (HOST/NATIVE)
- 4 WRITE (HOST/NATIVE)
- 5 SUBMIT
- 6 PRINT
- 7 MERGE
- 8 END
- 9 HELP

Storing and Handling Information

Entering/Editing Data with \$FSEDIT (continued)

3. Are you creating a new data set or editing an existing one?

- If you are creating a new data set, select option 2 without specifying a data set or volume name. (If the work data set contains a copy of a data set from a previous edit, use the CLEAR primary command before starting step 5 on page OP-213.) Skip to step 5 on page OP-213.
- If you are editing an existing data set, enter a 3 on the OPTION line, and enter the data set and volume names of the data set you are editing on the appropriate lines.

```
$FSEDIT PRIMARY OPTION MENU -----STATUS =
                                           PRESS PF3 TO EXIT
OPTION ==> 3
DATASET NAME =====> MYDATA1
VOLUME NAME =====> MYVOL
HOST DATASET =====>
ENTER A VOLUME NAME AND PRESS ENTER FOR A DIRECTORY LIST
1 .... BROWSE
2 .... EDIT
3 .... READ (HOST/NATIVE)
.
.
.
3 LINES READ FROM MYDATA1,MYVOL
LAST UPDATE AT: 08:18, DATE 10/20/82
```

4. Enter a 2 on the OPTION line to edit copy in the work data set.

```
$FSEDIT PRIMARY OPTION MENU -----STATUS =
                                           PRESS PF3 TO EXIT
OPTION ==> 2
DATASET NAME =====> MYDATA1
VOLUME NAME =====> MYVOL
HOST DATASET =====>
ENTER A VOLUME NAME AND PRESS ENTER FOR A DIRECTORY LIST
1 .... BROWSE
2 .... EDIT
.
.
.
EDIT --- MYWORKDS,MYVOL          0( 3)--- COLUMNS 001 072
COMMAND INPUT ==>                SCROLL ==>HALF
***** ***** TOP OF DATA *****
00010 THIS IS THE FIRST LINE
00020 THIS IS THE SECOND LINE
00030 THIS IS THE THIRD LINE
***** ***** BOTTOM OF DATA *****
```

Entering/Editing Data with \$FSEDIT (*continued*)

5. Enter or edit the data set.
(See Figure 26 on page OP-354 for a summary of the edit commands, or refer to the *Operator Commands and Utilities Reference* for a description of each command.)

```
EDIT --- MYWORKDS,MYVOL      0( 4)--- COLUMNS 001 072
COMMAND INPUT ==>          SCROLL ==>HALF
***** ***** TOP OF DATA *****
00010 THIS IS THE FIRST LINE
00020 THIS LINE WAS EDITED
00030 THIS IS THE THIRD LINE
00040 THIS IS A NEW LINE
***** ***** BOTTOM OF DATA *****
```

6. Enter the command **WRITE** on the COMMAND INPUT line to save the contents of the work data set in the original data set. If you want to save the work data set in a new data set, enter **WRITE dataset,volume**, where *dataset,volume* is the data set and volume name of the new data set.

```
EDIT --- MYWORKDS,MYVOL      0( 4)--- COLUMNS 001 072
COMMAND INPUT ==> WRITE          SCROLL ==>HALF
***** ***** TOP OF DATA *****
00010 THIS IS THE FIRST LINE
00020 THIS LINE WAS EDITED
00030 THIS IS THE THIRD LINE
00040 THIS IS A NEW LINE
***** ***** BOTTOM OF DATA *****
```

Storing and Handling Information

Entering/Editing Data with \$FSEDIT (continued)

7. Enter **M** to return to the primary option menu.

```
EDIT --- MYWORKDS,MYVOL    0( 4)--- COLUMNS 001 072
COMMAND INPUT ==> M          SCROLL ==>HALF
***** ***** TOP OF DATA *****
00010 THIS IS THE FIRST LINE
00020 THIS LINE WAS EDITED
00030 THIS IS THE THIRD LINE
00040 THIS IS A NEW LINE
***** ***** BOTTOM OF DATA *****

$FSEDIT PRIMARY OPTION MENU -----STATUS =
                                PRESS PF3 TO EXIT
OPTION ==> _

DATASET NAME =====> MYDATA1
VOLUME NAME =====> MYVOL

HOST DATASET =====>

ENTER A VOLUME NAME AND PRESS ENTER FOR A DIRECTORY LIST

  1 .... BROWSE
  2 .... EDIT
.
.
.
4 LINES WRITTEN TO MYDATA1,MYVOL
```

8. Enter **8** to end \$FSEDIT.

```
$FSEDIT PRIMARY OPTION MENU -----STATUS =
                                PRESS PF3 TO EXIT
OPTION ==> 8

DATASET NAME =====> MYDATA1
VOLUME NAME =====> MYVOL

HOST DATASET =====>

ENTER A VOLUME NAME AND PRESS ENTER FOR A DIRECTORY LIST

  1 .... BROWSE
  2 .... EDIT
.
.
.
$FSEDIT ENDED AT 08:30
```

Chapter 6. Running Programs and Job Procedures

Running programs and job procedures involves loading them, checking and controlling their progress, and, if necessary, cancelling them.

A program is a collection of program statements (computer instructions) that causes the Series/1 to do a specified task or job. A job procedure is a collection of \$JOBUTIL utility commands that run one or more programs, usually without operator intervention. Job procedures are usually used for programs that are run on a regular basis.

Programs and job procedures are stored in data sets. When EDX asks for the name of a program or job procedure, it is looking for the name of the data set containing the program or job procedure.

This chapter contains procedures for doing the following jobs:

- “Preparing to Run a Program” on page OP-216
- “Loading Programs” on page OP-217
- “Checking and Controlling Programs” on page OP-227
- “Cancelling Programs” on page OP-236
- “Deleting Job Queue Jobs” on page OP-237
- “Coding a \$JOBUTIL Procedure” on page OP-241
- “Running the Job Queue Processor \$JOBQ” on page OP-247.

Running programs also includes controlling program output. This job is covered in Chapter 7, “Controlling Program Output” on page OP-251.

Running Programs and Job Procedures

Preparing to Run a Program

Before you run a program, you must make sure that everything is ready. The following list helps verify that you are prepared to load your program. If you answer “no” to any of the questions that are applicable to the program you are running, you should correct the situation before trying to run the program. (See “Job Instructions” on page OP-315 for a suggested form to use in preparing to run programs and job procedures.)

- Are all devices switched on? (See “Switching Power On” on page OP-9 if you do not know how to switch on the Series/1.)
- Is required input data prepared and at hand?
- Are the correct diskettes in the diskette units? (See “Using Diskettes” on page OP-12 if you are not familiar with inserting diskettes into the diskette units.)
- Are all required volumes and data sets allocated? (See “Allocating Volumes and Data Sets” on page OP-121 for procedures that show how to allocate volumes and data sets.)
- Are the correct tapes mounted on the tape drives? (Refer to the *IBM Series/1 Operator's Guide*, GA34-0039 for instructions on mounting tapes.)
- Are the correct forms in the printer? (Refer to the *IBM Series/1 Operator's Guide*, GA34-0039 for instructions on loading paper.)
- If you are using \$JOBUTIL or the job queue processor (\$JOBQ, \$JOBQUT, \$SUBMIT), are the procedures set up? (See “Coding a \$JOBUTIL Procedure” on page OP-241 for instructions on writing a \$JOBUTIL procedure.)
- Do you have:
 - The name of the data set containing the program you are running?
 - The name of the volume containing the program? (See “List All Data Sets on All Volumes” on page OP-185 if you do not know volume name for the program data set.)
 - The amount of additional storage, if any, needed by the program? (Refer to your local procedures for this information.)
 - The names of any data sets used by the program? (Refer to your local procedures for this information.)

Loading Programs

You execute a program by loading it. When you load a program, the Event Driven Executive copies it, puts it in storage, and then runs it. Load a program in one of six ways:

1. By using the \$L operator command to load a single program.
2. By using session manager option 6 to load a single program.
3. By using the job stream processor utility, \$JOBUTIL, to load a procedure containing one or more programs.
4. By using the job queue processor, \$SUBMIT, to load and control multiple JOBUTIL procedures without tying up the display terminal.
5. By using one program to load another without external intervention.
6. By using the initial program load (IPL) procedure to load a program during IPL.

This chapter shows you how to load programs using the first four methods. The fifth, using one program to load another, is explained under the LOAD instruction in the *Language Reference*. The sixth, loading a program during initial program load, is explained in the *Customization Guide*.

Using the \$L Operator Command

Use the following procedure to load an individual program with the \$L operator command.

Before You Start:

Obtain the following information:

- The name of the data set that contains the program.
- The name of the volume containing the data set, unless it is on the IPL volume.
- The amount of additional storage needed by the program, if any. (Refer to your local procedures for running the program for this information.)
- The data set and volume names of any data sets that are used by the program, whose names are not already coded in the program. Refer to your local procedures for these names. (The system requests these names only if they are requested by the program.)

Running Programs and Job Procedures

Loading Programs (*continued*)

1. Press the attention key and enter \$L.

```
> $L  
PGM(NAME,VOLUME,STORAGE): _
```

2. Enter the name of the program data set, the name of the volume it is on, and the number of bytes of additional storage, if needed. The volume name does not have to be entered if the program is on the IPL volume. If the program is on the IPL volume and you need additional storage, you can enter **name,,storage** where **name** is the name of the program data set and **storage** is the number of bytes of additional storage.

```
PGM(NAME,VOLUME,STORAGE): MYPROG,MYVOL,200  
INPUT (NAME,VOLUME): _
```

3. Enter the data set and volume name of the first data set required. If more than one is required, repeat this step until all names have been supplied. The names must be entered in the order the program uses them, as determined by your local procedures.

```
INPUT (NAME,VOLUME): DATASET1,MYVOL  
LOADING MYPROG 3P,08:10:26, LP=0000, PART=2  
.  
.  
MYPROG ENDED AT 08:30:00
```

If the program does not require additional data sets, the system finishes loading the program after step 2. The **LOADING xxxx** message is followed by any other messages and prompts that are issued by the program.

When the program finishes, the system displays the program ended message.

Loading Programs (*continued*)

Using the Session Manager

Use the following procedure to load a program with the session manager. (See Chapter 4, "Using the Session Manager" on page OP-71 for a complete explanation of how to use the session manager.)

You may run a program either in the background or foreground option. Choose the foreground option for small jobs. You cannot use your terminal until a foreground job completes. Choose the background option for large jobs which run on another terminal.

Select foreground by coding a blank or an "F." Select background by coding a "B."

Before You Start:

Obtain the following information:

- The name of the data set that contains the program.
- The name of the volume containing the data set, unless it is on the IPL volume.
- The amount of additional storage needed by the program, if any. (Refer to your local procedures for the program for this information.)
- The data set and volume names of any data sets that must be passed to the program. Refer to your local procedures for these names. (The system requests these names only if they are requested by the program.)

Running Programs and Job Procedures

Loading Programs (*continued*)

1. If the session manager is not active at your terminal, load it and select option 6, EXECUTE PROGRAM/UTILITY.

```
$SMMPRIM: SESSION MANAGER PRIMARY OPTION MENU
ENTER/SELECT PARAMETERS: PRESS PF3 TO EXIT
SELECT OPTION ==> 6
.
.
.
$SMM06 SESSION MANAGER EXECUTE PROGRAM/UTILITY-----
ENTER/SELECT PARAMETERS: PRESS PF3 TO RETURN

PROGRAM/UTILITY (NAME,VOLUME) ==>

PARAMETERS ==>

DATA SET 1 (NAME,VOLUME / * = DS1 NOT USED) ==>
DATA SET 2 (NAME,VOLUME / * = DS2 NOT USED) ==>
DATA SET 3 (NAME,VOLUME / * = DS3 NOT USED) ==>

FOREGROUND OR BACKGROUND (F/B) ==>
(DEFAULT IS FOREGROUND)

NOTE: IF A DATA SET (DS1, DS2 OR DS3) IS NOT USED,
AN ASTERISK (*) MUST BE ENTERED IN THE DATA SET
FIELD.
```

Loading Programs (*continued*)

2. Enter the information required to run the program:

- A The data set and volume name for the data set containing the program. If the data set is on the IPL volume, you do not have to enter the volume name.
- B The number of bytes of additional storage needed by the program. If the program does not require additional storage, this parameter can be left blank.
- C The data set and volume names of up to three data sets used by the program. If the program does not use additional data sets, enter an * on each of these prompts.
- D The foreground or background option.

3. The session manager creates a \$JOBUTIL procedure to load your program. After you enter the information in step 2, the session manager builds the \$JOBUTIL procedure and starts it. \$JOBUTIL then loads your program.

```
$SMM06 SESSION MANAGER EXECUTE PROGRAM/UTILITY-----  
ENTER/SELECT PARAMETERS:          PRESS PF3 TO RETURN
```

- A PROGRAM/UTILITY (NAME,VOLUME) ==> MYPROG1,MYVOL
- B PARAMETERS ==> 200
- C DATA SET 1 (NAME,VOLUME / * = DS1 NOT USED) ==> DATA1,MYVOL
DATA SET 2 (NAME,VOLUME / * = DS2 NOT USED) ==> *
DATA SET 3 (NAME,VOLUME / * = DS3 NOT USED) ==> *
- D FOREGROUND OR BACKGROUND (F/B) ==> F
(DEFAULT IS FOREGROUND)

NOTE: IF A DATA SET (DS1, DS2 OR DS3) IS NOT USED,
AN ASTERISK (*) MUST BE ENTERED IN THE DATA SET
FIELD.

```
LOADING $JOBUTIL          4P,08:29:18, LP= 0400, PART=2  
REMARK  
EXECUTE PROGRAM/UTILITY: MYPROG1,MYVOL  
*** JOB - MYPROG1, - STARTED AT 08:29:20 04/25/82 ***  
  
JOB          MYPROG1,MYVOL ($SMP06) USERID=MYID  
LOADING MYPROG1          3P,08:29:22, LP=0800, PART= 2
```

Running Programs and Job Procedures

Loading Programs (*continued*)

4. When the program ends, the system displays messages similar to those in the example.

Press the enter key to return to the session manager option 6 secondary option menu shown in step 2 on page OP-221.

```
MYPROG1 ENDED AT 09:00:00
$JOBUTIL ENDED AT 09:00:00
DEPRESS ENTER KEY TO RETURN
```

Using the Command Stream Processor, \$JOBUTIL

One or more programs can be run sequentially without operator intervention, using a predefined job command stream. If you have a job stream procedure set up, you can start it by using either the \$L operator command or session manager option 7. If you want to run multiple \$JOBUTIL procedures, see “Loading a procedure with the Job Submission Utility - \$SUBMIT” on page OP-225.

See “Coding a \$JOBUTIL Procedure” on page OP-241 for instructions on setting up a job command stream procedure.

Loading a Procedure with the \$L Operator Command

Use the following procedure to load a job procedure with the \$L operator command.

Before You Start:

Obtain the following information:

- The name of the data set containing the procedure. The name of the data set is the same as the name of the procedure.
- The name of the volume containing the data set. If the data set is on the IPL volume, the volume name is optional.

Loading Programs (*continued*)

1. Load the job stream processor utility, \$JOBUTIL, by pressing the attention key and entering \$L \$JOBUTIL.

```
> $L $JOBUTIL
DS1(NAME,VOLUME): _
```

2. Enter the data set and volume name of the procedure. If the procedure is on the IPL volume, the volume name is optional.

\$JOBUTIL loads each program in the procedure. As each program ends, \$JOBUTIL loads the next, until all programs in the procedure have run.

```
DS1(NAME,VOLUME): MYPROC,MYVOL
```

```
LOADING $JOBUTIL      4P,08:30:14, LP=0000, PART=2
*** JOB - MY JOB - STARTED AT 08:30:15 05/03/82 ***
```

```
JOB      MY JOB
REMARK   THIS IS MY JOB TO LOAD MY PROGRAM
LOADING MYPROG1      3P,08:30:16, LP=0400, PART=2
MYPROG1 ENDED AT 08:45:00
LOADING MYPROG2      3P,08:30:16, LP=0400, PART=2
MYPROG2 ENDED AT 09:00:04
$JOBUTIL ENDED AT 09:00:05
```

Loading a Procedure with the Session Manager

Use the following procedure to load a job procedure with the session manager.

You may run a program either in the background or foreground option. Choose the foreground option for small jobs. You cannot use your terminal until a foreground job completes. Choose the background option for large jobs which run on another terminal.

Select foreground by coding a blank or an "F." Select background by coding a "B."

Before You Start:

Obtain the following information:

- The name of the data set containing the procedure. The name of the data set is the same as the name of the procedure.
- The name of the volume containing the data set. If the data set is on the IPL volume, the volume name is optional.

Running Programs and Job Procedures

Loading Programs (*continued*)

1. If the session manager is not active at your terminal, load it and select option 7, EXECUTE A \$JOBUTIL PROCEDURE.

```
$SMMPRIM: SESSION MANAGER PRIMARY OPTION MENU
ENTER/SELECT PARAMETERS: PRESS PF3 TO EXIT
SELECT OPTION ==> 7

.
.
.
$SMM07 SESSION MANAGER EXECUTE $JOBUTIL PROC-----
ENTER/SELECT PARAMETERS: PRESS PF3 TO RETURN

PROC (NAME,VOLUME) ==> _
FOREGROUND OR BACKGROUND (F/B) ==>
(DEFAULT IS FOREGROUND)
```

2. Enter the data set and volume name of the procedure. Specify whether the program will run foreground or background. If the procedure is on the IPL volume, the volume name is optional.

```
$SMM07 SESSION MANAGER EXECUTE $JOBUTIL PROC-----
ENTER/SELECT PARAMETERS: PRESS PF3 TO RETURN

PROC (NAME,VOLUME) ==> MYPROC,MYVOL
FOREGROUND OR BACKGROUND (F/B) ==> F
```

3. The session manager starts \$JOBUTIL, which loads the procedure.

\$JOBUTIL loads each program in the procedure. As each program ends, \$JOBUTIL loads the next, until all programs in the procedure have run.

After the procedure completes, press the enter key to return to the session manager option menu for option 7.

```
LOADING $JOBUTIL 4P,08:30:44, LP= 0400, PART= 2
EXECUTE $JOBUTIL PROC: MYPROC,MYVOL
*** JOB - MYPROC - STARTED AT 08:30:46 05/03/82

JOB MYPROC,MYVOL ($SMP07) USERID=XXX
*** JOB - MY JOB - STARTED AT 08:30:49 05/03/82
JOB MY JOB
REMARK THIS IS MY JOB TO LOAD MY PROGRAM
LOADING MYPROG1 3P,08:30:48, LP= 0800, PART= 2
MYPROG1 ENDED AT 08:45:05
LOADING MYPROG1 3P,08:45:10, LP= 0800, PART= 2
MYPROG2 ENDED AT 09:00:00
$JOBUTIL ENDED AT 09:00:01
DEPRESS ENTER KEY TO RETURN
```

Loading Programs (*continued*)

Loading a procedure with the Job Submission Utility - \$SUBMIT

The job submission utility, \$SUBMIT, gives you the ability to submit and control up to 50 \$JOBUTIL job procedures at one time without tying up the terminal while they are running.

\$SUBMIT uses the job queue processor (\$JOBQ). The job queue processor is used for programs that do not require interactive operator input.

Use the following procedure to submit a job to the job queue processor.

Before You Start:

Obtain the following information:

- The name of the data set and volume containing the \$JOBUTIL procedure you are submitting. (The data set name is the same as the \$JOBUTIL procedure name.)
- The priority you are assigning to the job. Valid priorities are 0 - 3. If you do not enter a priority, \$SUBMIT assigns a priority of 2. (0 is the highest priority. EDX runs jobs with a priority of 0 first.)

1. Load the utility \$SUBMIT.

- Using the \$L operator command, press the attention key and enter **\$L \$SUBMIT**.
- Using the session manager, select option 10.2.

(This procedure assumes that the job queue processor (\$JOBQ) is already running. If \$JOBQ is not running, \$SUBMIT gives you an opportunity to start it. If you are not sure how to answer the prompts for starting \$JOBQ, see "Running the Job Queue Processor \$JOBQ" on page OP-247)

```
> $L $SUBMIT
LOADING $SUBMIT 21P,08:10:00, LP=A000...

$SUBMIT - EDX JOB QUEUE SUBMISSION PROGRAM
ENTER '?' AT ANY TIME FOR HELP

COMMAND (?): _
```

2. Do you want to submit a job to run, or do you want that job held so you can release it later?

- Enter **SJ**, to submit a job to run.
- Enter **SH**, to submit a job in a hold status.

```
COMMAND (?): SJ

ENTER COMMAND DATA SET (NAME,VOLUME): _
```

3. Enter the names of the data set and volume containing the procedure you are running.

```
COMMAND DATA SET (NAME,VOLUME): MYPROC,MYVOL
ENTER A PRIORITY FROM 0 TO 3 (DEFAULT 2): _
```

Running Programs and Job Procedures

Loading Programs (*continued*)

4. Enter the number of the priority you want assigned to the job. Valid priorities are 0, 1, 2, or 3. (0 is the highest priority - the system runs jobs with a priority of 0 first.) If you press the enter key without entering anything, \$SUBMIT assigns a priority of 2.

```
JOB PRIORITY: 1
JOB NUMBER 2 : SUBMITTED
COMMAND DATA : MYPROC,MYVOL
JOB PRIORITY : 1
COMMAND (?): -
```

5. Do you want to submit another procedure?

- If so, continue with step 2 on page OP-225.
- If not, enter **EN** to end \$SUBMIT.

```
COMMAND (?): EN
$SUBMIT ENDED AT 08:31:00
```

Checking and Controlling Programs

Some operating environments take care of themselves and do not require any intervention unless there is a problem. Others require some degree of intervention. This intervention includes checking to see what programs are running, controlling jobs being run by the job queue processor, and controlling printed output.

This section contains procedures for checking to see what programs are running and for controlling the job queue processor.

- “Listing Partitions and Active Programs”
- “Controlling Job Queue Jobs” on page OP-228
- “Responding to a \$JOBUTIL Pause” on page OP-235.

Listing Partitions and Active Programs

Use this procedure to display the starting address and size of each storage partition, and the names and starting addresses of all programs running in each partition.

Do you want to display a listing of all partitions defined on your system, or just the partition to which your terminal is assigned?

- To display all partitions, press the attention key and enter **\$A ALL**.
- To display only the partition to which your terminal is assigned, press the attention key and enter only **\$A**.

```
> $A ALL  
  
PROGRAMS AT 08:04:42  
IN PARTITION #1  
MYPROG 9200  
PART. ADDR: 9200 HEX; SIZE: 28160 DEC. BYTES  
  
PROGRAMS AT 08:04:42  
IN PARTITION #2  
$SMUXXX 0000 *  
$FSEDIT 0400 *  
PART. ADDR: 0000 HEX; SIZE: 65536 DEC. BYTES
```

In the example screen, partition 1 contains one program, MYPROG, which starts at hexadecimal address 9200. (The supervisor occupies the first part of partition 1.)

Partition 2 contains 2 programs, \$SMUXXX and \$FSEDIT. \$SMUXXX starts at hexadecimal address 0000, and \$FSEDIT starts at 0400. The listing was run from a terminal assigned to partition 2 and the asterisk shows that the two programs were started from that terminal. The partition starts at hexadecimal address 0000 and is decimal 65,536 bytes in size.

Running Programs and Job Procedures

Checking and Controlling Programs *(continued)*

Controlling Job Queue Jobs

The job queue processor provides the ability to control when jobs are permitted to run. You can control the job queue processor as follows:

- “Display the Status of the Job Queue”
- “Release a Held Job” on page OP-229
- “Holding a Job” on page OP-230
- “Suspending and Resuming Job Queue Processing” on page OP-231
- “Changing the Job Queue Logging Terminal” on page OP-233.

Display the Status of the Job Queue

Use the following procedure to display the status of the job queue.

1. You can use either the \$SUBMIT utility or the \$JOBQUT utility to display the status of the job queue. (This procedure assumes that the job queue processor (\$JOBQ) is already running. If \$JOBQ is not running, \$SUBMIT and \$JOBQUT give you an opportunity to start it. If you are not sure how to answer the prompts for starting \$JOBQ, see “Running the Job Queue Processor \$JOBQ” on page OP-247.)

- To use the \$SUBMIT utility enter \$L \$SUBMIT or select option 10.2 of the session manager.
- To use the \$JOBQUT utility enter \$L \$JOBQUT or select option 10.1 of the session manager.

```
> $L $xxxxxx
LOADING xxxxxxxx 21P,08:10:00, LP=A000, PART=1

$xxxxxx - EDX JOB QUEUE xxxxxxxxxxxx PROGRAM
ENTER '?' AT ANY TIME FOR HELP

COMMAND (?): _
```

2. Enter STQ to display the status of the job queue.

```
COMMAND (?): STQ

JOB QUEUE PROCESSING IS IN EFFECT
LOGGING TERMINAL IS $$SYSLOG
JOB ID | COMMAND DATA SET | PRIORITY | STATUS
3      | MYPROC1,MYVOL     | 0        | EXECUTING
4      | MYPROC2,MYVOL     | 3        | HELD
5      | MYPROC3,MYVOL     | 1        | WAITING
NUMBER OF JOBS QUEUED IS 3

COMMAND (?): _
```

Checking and Controlling Programs (*continued*)

3. Do you want to continue to use \$SUBMIT or \$JOBQUT?

- If so, continue with that procedure.
- If not, enter **EN** to end \$SUBMIT or \$JOBQUT. The job queue processor remains active.

```
COMMAND (?): EN
$xxxxxx ENDED AT 08:30:45
```

Release a Held Job

Use the following procedure to submit a job that is in hold status.

Before You Start:

Obtain the number of the job you are releasing. If you do not know the number, see “Display the Status of the Job Queue” on page OP-228.

1. Load the utility \$SUBMIT:

- Using the \$L operator command, press the attention key and enter **\$L \$SUBMIT**.
- Using the session manager, select option 10.2.

(This procedure assumes that the job queue processor (\$JOBQ) is already running. If \$JOBQ is not running, \$SUBMIT gives you an opportunity to start it. See “Running the Job Queue Processor \$JOBQ” on page OP-247 for the answers to the prompts.)

```
> $L $SUBMIT
LOADING $SUBMIT 21P,08:10:00, LP=A000,...

$SUBMIT - EDX JOB QUEUE SUBMISSION PROGRAM
ENTER '?' AT ANY TIME FOR HELP

COMMAND (?): _
```

2. Enter **RJ** to release a held job.

```
COMMAND (?): RJ
JOB NUMBER: _
```

3. Enter the number of the job being released.

```
JOB NUMBER: 2
JOB 2 RELEASED
COMMAND (?): _
```

Running Programs and Job Procedures

Checking and Controlling Programs (*continued*)

4. Do you want to continue to use \$SUBMIT?

- If so, continue with that procedure.
- If not, enter **EN** to end \$SUBMIT. The job queue processor remains active.

```
COMMAND (?): EN
$SUBMIT ENDED AT 08:30:45
```

Holding a Job

Use this procedure to place a submitted job into hold status. The job cannot be running (executing).

Before You Start:

Obtain the number of the job you are holding. If you do not know the number, see “Display the Status of the Job Queue” on page OP-228.

1. Load the utility \$SUBMIT:

- Using the \$L operator command, press the attention key and enter **\$L \$SUBMIT**.
- Using the session manager, select option 10.2.

(This procedure assumes that the job queue processor (\$JOBQ) is already running. If \$JOBQ is not running, \$SUBMIT gives you an opportunity to start it. If you are not sure how to answer the prompts for starting \$JOBQ, see “Running the Job Queue Processor \$JOBQ” on page OP-247.)

```
> $L $SUBMIT
LOADING $SUBMIT 21P,08:10:00, LP=A000,...

$SUBMIT - EDX JOB QUEUE SUBMISSION PROGRAM
ENTER '?' AT ANY TIME FOR HELP

COMMAND (?): _
```

2. Enter **HJ** to hold a job queue.

```
COMMAND (?): HJ

JOB NUMBER: _
```

Checking and Controlling Programs *(continued)*

3. Enter the number of the job being held.

```
JOB NUMBER: 2
```

```
JOB 2 HELD  
COMMAND (?): _
```

4. Do you want to continue to use \$SUBMIT?

- If so, continue with that procedure.
- If not, enter **EN** to end \$SUBMIT. The job queue processor remains active.

```
COMMAND (?): EN  
$SUBMIT ENDED AT 08:30:45
```

Suspending and Resuming Job Queue Processing

Use this procedure to temporarily stop (suspend) job queue processing or restart it after it has been suspended.

1. Load the utility \$JOBQUT:

- Using the \$L operator command, press the attention key and enter **\$L \$JOBQUT**.
- Using the session manager, select option 10.1.

(This procedure assumes that the job queue processor (\$JOBQ) is already running. If \$JOBQ is not running, \$JOBQUT gives you an opportunity to start it. If you are not sure how to answer the prompts for starting \$JOBQ, see "Running the Job Queue Processor \$JOBQ" on page OP-247.)

```
> $L $JOBQUT  
LOADING $JOBQUT 21P,08:10:00, LP=A000,...  
$JOBQUT - EDX JOB QUEUE CONTROLLER PROGRAM  
ENTER '?' AT ANY TIME FOR HELP  
COMMAND (?): _
```

Running Programs and Job Procedures

Checking and Controlling Programs (*continued*)

2. Do you want to suspend job queue processing, or restart it?

- To suspend job queue processing continue with step 2(a).
- To resume job queue processing continue with step 2(b).

2(a). Enter **SJQ** to suspend job queue processing.

```
COMMAND (?): SJQ
JOB QUEUE PROCESSING HAS BEEN SUSPENDED
COMMAND (?): _
```

2(b). Enter **RJQ** to resume job queue processing.

```
COMMAND (?): RJQ
JOB QUEUE PROCESSING HAS BEEN RESUMED
COMMAND (?): _
```

3. Do you want to continue to use \$JOBQUT?

- If so, continue with that procedure.
- If not, enter **EN** to end \$JOBQUT. The job queue processor remains active.

```
COMMAND (?): EN
$JOBQUT ENDED AT 08:30:45
```

Checking and Controlling Programs (*continued*)

Changing the Job Queue Logging Terminal

Use the following procedure to change the terminal that the job queue log is printed on, or to stop job queue logging.

Before You Start:

Determine the name of the terminal on which you want to print the job queue log. (See “Listing System Terminal Assignments” on page OP-44 if you don’t know the name of the terminal you want to use.)

1. Load the utility \$JOBQUT:

- Using the \$L operator command, press the attention key and enter \$L \$JOBQUT.
- Using the session manager, select option 10.1.

(This procedure assumes that the job queue processor (\$JOBQ) is already running. If \$JOBQ is not running, \$JOBQUT gives you an opportunity to start it. If you are not sure how to answer the prompts for starting \$JOBQ, see “Running the Job Queue Processor \$JOBQ” on page OP-247.)

```
> $L $JOBQUT
LOADING $JOBQUT 21P,08:10:00, LP=A000, PART=1
$JOBQUT - EDX JOB QUEUE CONTROLLER PROGRAM
ENTER '?' AT ANY TIME FOR HELP
COMMAND (?): _
```

2. Enter CLT to change the logging terminal.

```
COMMAND (?): CLT
THE LOGGING TERMINAL IS NOW $SYSLOG
NEW LOGGING TERMINAL NAME OR '*' FOR THIS TERMINAL: _
```

Running Programs and Job Procedures

Checking and Controlling Programs (*continued*)

3. Enter the name of the terminal you want used as the logging terminal or an * if you want the display terminal you are currently using to be the log terminal.

```
NEW LOGGING TERMINAL NAME OR '*' FOR THIS TERMINAL: *  
LOGGING TERMINAL CHANGED TO TERM02  
PUT TERMINAL IN ROLL MODE (Y/N)? _
```

4. Do you want to put the display terminal into roll mode? (If you put the display terminal into roll mode, you do not have to press the enter key every time the display screen is full.)

```
PUT TERMINAL IN ROLL MODE (Y/N)? Y  
COMMAND (?): _
```

- If so, enter a Y.
- If not, enter an N.

5. Do you want to continue to use \$JOBQUT?

```
COMMAND (?): EN  
$JOBQUT ENDED AT 08:30:45
```

- If so, continue with that procedure.
- If not, enter EN to end \$JOBQUT. The job queue processor remains active.

Checking and Controlling Programs (*continued*)

Responding to a \$JOBUTIL Pause

A job procedure can stop (pause) to ask that \$JOBUTIL commands be entered from the terminal. Your local procedures will tell you what commands should be entered. This procedure shows you how to enter those commands.

Note: In some instances, such as entering an EXEC command, \$JOBUTIL generates an extra pause, with the message PAUSE OPERATOR REQUESTED PAUSE, to give you an opportunity to enter additional commands.

1. If a \$JOBUTIL job procedure pauses press the attention key and enter:

- **ABORT** to end the \$JOBUTIL procedure.
- **ENTER** to enter \$JOBUTIL commands, continue with step 2 .
- **GO** to end the pause mode and continue the procedure.

```
PAUSE - * - ATTN: GO/ENTER/ABORT
PAUSE
> ENTER
ENTER COMMAND _
```

2. Enter the command.

```
ENTER COMMAND PARM
ENTER OPERAND _
```

3. Enter the operand for the command. If the command does not have an operand, press the enter key without typing anything.

```
ENTER OPERAND 050482
ENTER COMMAND _
```

4. Do you have another command to enter?

- If so, continue with step 2.
- If not, enter **GO** to restart the procedure.

```
ENTER COMMAND GO
```

Running Programs and Job Procedures

Cancelling Programs

Use this procedure to cancel a running application program when there is no other way to end it.

You can also use this procedure to cancel a system utility program, unless you received a message stating that the utility cannot be cancelled with \$C when you loaded it.

Do not use this procedure as a normal means of stopping program execution.

Before You Start:

Observe the following requirements:

- The program and your display terminal must be running in the same partition. (See “Changing a Terminal’s Partition Assignment” on page OP-50 to change your terminal’s partition.)
- Get the name of the program you are cancelling.
- Get the load point (starting address) of the program, if more than one copy of it is in the same partition. (See “Listing Partitions and Active Programs” on page OP-227 if you do not know the name or load point.)

1. Are you cancelling a program that is using your display terminal screen?

- If the program is not using the display screen, continue with step 1(a).
- If the program is using the display screen, continue with step 1(b).

1(a). To cancel a system utility that is not using the display screen of your terminal, or an application program running in the same partition as your terminal, press the attention key, enter \$C, and continue with step 2.

```
> $C
PROGRAM NAME: _
```

1(b). To cancel a system utility that is using the display screen of your terminal:

1. Press the attention key.
2. Press the enter key.
3. Enter \$C.

```
> $C
PROGRAM NAME: _
```

Cancelling Programs *(continued)*

2. Enter the name of the application program or system utility you are cancelling.

```
PROGRAM NAME: MYPROG  
LOAD POINT= _
```

3. If more than one copy of the program is running in the partition, the supervisor requests the load point of the program you are cancelling. Enter the hexadecimal starting address of the program you are cancelling.

```
LOAD POINT= 0200  
MYPROG CANCELLED AT 08:30:24
```

Deleting Job Queue Jobs

Use the following procedures to delete jobs running under the job queue processor and to end the job queue processor. The only way to delete a job that is running is to use the \$C operator command as described under "Cancelling Programs" on page OP-236.

Deleting One Job

Use this procedure to delete one job in the job queue.

Before You Start:

Obtain the number of the job you are cancelling. The job must be either waiting for execution or held. (If you do not know the number or status, see "Display the Status of the Job Queue" on page OP-228.)

Running Programs and Job Procedures

Deleting Job Queue Jobs (*continued*)

1. Load the utility \$SUBMIT:

- Using the \$L operator command, press the attention key and enter \$L \$SUBMIT.
- Using the session manager, select option 10.2.

(This procedure assumes that the job queue processor (\$JOBQ) is already running. If \$JOBQ is not running, \$SUBMIT gives you an opportunity to start it. If you are not sure how to answer the prompts for starting \$JOBQ, see "Running the Job Queue Processor \$JOBQ" on page OP-247.)

```
> $L $SUBMIT
LOADING $SUBMIT 21P,08:10:00, LP=A000...

$SUBMIT - EDX JOB QUEUE SUBMISSION PROGRAM
ENTER '?' AT ANY TIME FOR HELP

COMMAND (?): _
```

2. Enter DJ to delete a job.

```
COMMAND (?): DJ
JOB NUMBER: _
```

3. Enter the number of the job being deleted.

```
JOB NUMBER: 2
DELETE JOB 2 (Y/N)? _
```

4. Have you entered the correct job number?

- If so, enter a Y.
- If not, enter an N.

```
DELETE JOB 2 (Y/N)? Y
JOB 2 DELETED
COMMAND (?): _
```

5. Do you want to delete another job?

- If so, continue with step 2.
- If not, enter EN to end \$SUBMIT. The job queue processor remains active.

```
COMMAND (?): EN
$SUBMIT ENDED AT 08:30:45
```

Deleting Job Queue Jobs (*continued*)

Deleting All Job Queue Jobs

Use this procedure to delete all jobs in a held or waiting status. No jobs can be running when this command is entered.

1. Load the utility \$JOBQUT:

- Using the \$L operator command, press the attention key and enter **\$L \$JOBQUT**.
- Using the session manager, select option 10.1.

(This procedure assumes that the job queue processor (\$JOBQ) is already running. If \$JOBQ is not running, \$JOBQUT gives you an opportunity to start it. If you are not sure how to answer the prompts for starting \$JOBQ, see "Running the Job Queue Processor \$JOBQ" on page OP-247.)

```
> $L $JOBQUT
LOADING $JOBQUT 21P,08:10:00, LP=A000...

$JOBQUT - EDX JOB QUEUE CONTROLLER PROGRAM
ENTER '?' AT ANY TIME FOR HELP

COMMAND (?): _
```

2. Enter **DJQ** to delete all jobs.

```
COMMAND (?): DJQ
DELETE ALL JOBS IN THE QUEUE (Y/N)? _
```

3. If you want to delete all jobs, enter a **Y**.

If you want to change your mind, enter an **N**.

```
DELETE ALL JOBS IN THE QUEUE (Y/N)? Y
QUEUE CLEARED
COMMAND (?): _
```

4. Do you want to continue to use \$JOBQUT?

- If so, continue with that procedure.
- If not, enter **EN** to end \$JOBQUT. The job queue processor remains active.

```
COMMAND (?): EN
$JOBQUT ENDED AT 08:30:45
```

Running Programs and Job Procedures

Deleting Job Queue Jobs *(continued)*

Ending Job Queue Processing

Use this procedure to end job queue processing. If a job is running, \$JOBQUT waits until it finishes before ending job queue processing.

1. Load the utility \$JOBQUT:

- Using the \$L operator command, press the attention key and enter \$L \$JOBQUT.
- Using the session manager, select option 10.1.

(This procedure assumes that the job queue processor (\$JOBQ) is already running. If \$JOBQ is not running, \$JOBQUT gives you an opportunity to start it. If you are not sure how to answer the prompts for starting \$JOBQ, see "Running the Job Queue Processor \$JOBQ" on page OP-247.)

```
> $L $JOBQUT
LOADING $JOBQUT 21P,08:10:00, LP=A000...

$JOBQUT - EDX JOB QUEUE CONTROLLER PROGRAM
ENTER '?' AT ANY TIME FOR HELP

COMMAND (?): _
```

2. Enter TJQ to end job queue processing.

```
COMMAND (?): TJQ
TERMINATE JOB QUEUE PROCESSING (Y/N)? _
```

3. If you want to continue and end job queue processing, enter a Y.

If you want to change you mind, enter an N.

```
TERMINATE JOB QUEUE PROCESSING (Y/N)? Y
DELETE ALL JOBS IN THE QUEUE (Y/N)? _
```

4. Do you want to delete the queue of jobs? (If you save the queue, the jobs can be run when you restart job queue processing.)

- If so, enter a Y.
- If not, enter an N.

```
DELETE ALL JOBS IN THE QUEUE (Y/N)? Y
JOB QUEUE DATA SET RE-INITIALIZED
JOB QUEUE PROCESSING TERMINATED
$JOBQUT ENDED AT 08:30:45
```

Coding a \$JOBUTIL Procedure

A \$JOBUTIL procedure allows you to run a series of programs without operator intervention. A procedure is a series of commands that tell the utility \$JOBUTIL what you want done. Use a \$JOBUTIL procedure to do the following: (The names of the associated \$JOBUTIL commands are in parenthesis.)

- Allocate a data set (AL)
- Delete a data set (DE)
- Give instructions to an operator (REMARK)
- Load and run a program (PROGRAM, EXEC)
- Pause for operator intervention (PAUSE)
- Print a log of the procedure commands (LOG)
- Provide a parameter to a program (PARM)
- Start a new page in the log listing (EJECT)
- Specify the name of a data set used by a program (DS)
- Turn off the program load messages (NOMSG).

Additionally, the flow or operation of the procedure can be controlled in the following ways:

- Call a second procedure (PROC)
- Identify the end of a procedure (EOJ, EOP)
- Jump, or branch, to another part of the procedure (JUMP, LABEL)
- Print the name of the job , the date, and the time the job started (JOB).

This section helps you plan a procedure, explains how to enter the procedure commands into a data set, and provides a sample procedure. See “\$JOBUTIL Command Syntax” on page OP-376 for the command syntax of the individual commands, and refer to the *Operator Commands and Utilities Reference* for a complete description of \$JOBUTIL.

Planning a Job Procedure

Planning a job procedure is much like planning a simple application program. You have to decide what is to be done and what supporting information is needed. Use the following form to help plan your procedures. The command that supports each item is in parenthesis. (See Appendix B, “Blank Forms” on page OP-397 for a copy of this form.)

Running Programs and Job Procedures

Coding a \$JOBUTIL Procedure (continued)

FOR THE PRIMARY PROCEDURE:

Job identification (JOB): Example procedure

Stored in: MYPROC2, MYVOL on disk _____ diskette # D111

Command logging: (LOG) ON OFF _____ Terminal name \$SYSPRTR

Name and volume of nested procedures called by this procedure:

MYPROC2, MYVOL

FOR EACH PROGRAM:

Name of the program to be run (PROGRAM & EXEC): MYPROG, MYVOL (in nested procedure)

Additional storage required by the program: EXEC STORAGE= _____

Turn off the log message for this program (NOMSG): Yes _____ No

Data sets that must be allocated before the program can be run (AL):

name	volume	size	type	name	volume	size	type
<u>MYDS1</u>	<u>MYVOL</u>	<u>200</u>	<u>D</u>	<u>(allocate in primary procedure)</u>			

JUMP to LABEL LABEL1 if allocate is not equal (NE) to -1

Data sets that must be deleted before the program can be run (DE):

Data sets that must be deleted after the program is run (DE):

MYDS1, MYVOL (delete in primary procedure)

Data sets whose names must be passed when program is loaded:

MYDS1, MYVOL

JUMP to LABEL _____ if condition code from _____ is:
LT _____ GT _____ EQ _____ GE _____ LE _____ NE _____

Parameters required by the program (PARM): 050482

Operator intervention required before/after the program

is run (PAUSE): Yes _____ No

If yes, commands to be entered:

Special processing:

LABEL1 issues REMARK that allocate failed

Jump around failed message if it isn't issued

Figure 13. \$JOBUTIL Procedure Guide

Coding a \$JOBUTIL Procedure (*continued*)

Planning Considerations:

The following items should be considered as you plan your procedure:

- A procedure can call a second, or nested, procedure. A nested procedure cannot call a third.
- When \$JOBUTIL returns from a nested procedure, the return code from the last program executed in the nested procedure is available for use by the JUMP command.
- A JUMP command only branches forward, toward the end of the procedure.
- The corresponding LABEL command for a JUMP command must be in the same procedure, not a nested procedure.
- Printing of job stream commands can be turned on or off at anytime during the procedure.
- The system programs capable of receiving parameters in the format used by \$JOBUTIL

are: \$COBOL, \$EDXASM, \$EDXLINK, \$FORT, Pascal, \$PLI, \$MEMDISK, \$PREFIND, \$S1ASM, and \$UPDATE.

- The AL command returns a \$DISKUT3 return code that can be used by the JUMP command.

Entering a Procedure

Enter a procedure into a data set using either \$EDIT1N or \$FSEDIT. A procedure for entering data using \$FSEDIT is under “Entering/Editing Data with \$FSEDIT” on page OP-210. \$EDIT1N and \$FSEDIT are described in the *Operator Commands and Utilities Reference*.

Command Syntax

The \$JOBUTIL commands consist of a command name, operands, and comments. The command name is an abbreviation representing the action to be taken by \$JOBUTIL. The operands are information needed by \$JOBUTIL to do the action, such as the name of a program to be loaded. And the comments explain what the command is doing. They are optional and are for reference only.

Enter the commands in the following format:

Command Start in position 1.

Operands Start in position 10.

Comments Start in position 18, or after, and can extend to position 71. (At least one blank must separate the comment from the operands.) If a command does not have any operands, its comments can start in column 10.

An asterisk (*) in position one makes the line a comment.

Running Programs and Job Procedures

Coding a \$JOBUTIL Procedure (*continued*)

Coding Order of the Commands

The following command sequence requirements should be observed when coding your \$JOBUTIL procedure:

- PROGRAM and EXEC commands are coded in pairs. After reading a PROGRAM command, \$JOBUTIL must read an EXEC command before reading another PROGRAM command.
- If used, the following commands must be coded between a PROGRAM command and an EXEC command:
 - NOMSG - no message logging
 - PARM - pass parameter
 - DS - identify data set.
- If used, the following commands should not be coded between a PROGRAM command and an EXEC command:
 - AL - allocate data set
 - DE - delete data set
 - PROC - execute procedure
- Only one level of nested procedures is permitted. A PROC command cannot be coded in a procedure that is called by a PROC command.
- A JUMP command only branches forward, toward the end of the procedure.
- The corresponding LABEL command for a JUMP command must be in the same procedure, not a nested procedure.
- An EOJ command must be coded at the end of a primary procedure.
- An EOP command must be coded at the end of a nested procedure.
- A REMARK command can be coded whenever needed.

See “\$JOBUTIL Command Syntax” on page OP-376 for the syntax of the individual commands.

Coding a \$JOBUTIL Procedure *(continued)*

Example Procedure

Refer to the example primary procedure in Figure 14 on page OP-246 and the example nested procedure in Figure 15 on page OP-246 when writing your own procedure.

The following list explains each line of the primary example.

The primary procedure:

- 1** Prints the name of the job. The date and time are also printed if timer support is included in the system.
- 2** Turns on command logging to the system printer \$SYSPRTR.
- 3** Issues a message to the operator.
- 4** Allocates a data set.
- 5** Checks to see if the data set was allocated successfully. If the data set is not allocated successfully, it jumps around the request to load a nested procedure.
- 6** Prints messages telling the operator that the data set was allocated successfully and that the nested procedure is being called.
- 7** Requests that a nested procedure be loaded.
- 8** Prints a message telling the operator that the nested procedure has ended and that the data set is being deleted.
- 9** Deletes the data set.
- 10** Jumps past the message telling the operator that the allocate failed.
- 11** Tells the operator that the procedure is finished.
- 12** Ends.

The following list explains each line of the nested example.

The nested procedure:

- 1** Prints the name of the nested procedure.
- 2** Identifies the data set and volume name of a program.
- 3** Specifies a parameter to be passed to the program.
- 4** Specifies the name of a data set used by the program.
- 5** Loads the program.
- 6** After the program finishes, ends.

Running Programs and Job Procedures

Coding a \$JOBUTIL Procedure (continued)

```
1---5---0---1---1 1 2---2---3---3---4---4---5---5---6---6---77
1---5---0---5---8-0---5---0---5---0---5---0---5---0---5---01
1 JOB      EXAMPLE PROCEDURE
2 LOG      $SYSPRTR      LOG TO THE SYSTEM PRINTER
3 REMARK   ALLOCATING MYDS1 ON MYVOL
* ALLOCATE A 200 RECORD DATA SET OF TYPE DATA
4 AL       MYDS1,MYVOL 200 D
* JUMP TO LABEL1 IF ALLOCATE UNSUCCESSFUL
5 JUMP     LABEL1 NE -1
6 REMARK   ALLOCATE SUCCESSFUL, LOADING MYPROC2
7 PROC     MYPROC2,MYVOL  LOAD NESTED PROCEDURE
8 REMARK   RETURNED FROM MYPROC2, DELETING MYDS1
9 DE       MYDS1,MYVOL  DELETE DATA SET
10 JUMP    LABEL2          JUMP AROUND FAILED MESSAGE
    LABEL  LABEL1          COME HERE IF ALLOCATE FAILS
    REMARK ALLOCATE FAILED
    LABEL  LABEL2          COME HERE IF MYPROC2 RAN
11 REMARK  END OF PROCEDURE
12 EOJ
```

Figure 14. Example \$JOBUTIL Procedure

```
1---5---0---1---1 1 2---2---3---3---4---4---5---5---6---6---77
1---5---0---5---8-0---5---0---5---0---5---0---5---0---5---01
1 JOB      EXAMPLE NESTED PROCEDURE
2 PROGRAM  MYPROG,MYVOL  IDENTIFY PROGRAM TO RUN
3 PARM     050482        PASS PARAMETER
4 DS       MYDS1,MYVOL  PASS DATA SET
5 EXEC     LOAD PROGRAM
6 EOP      END NESTED PROCEDURE
```

Figure 15. Example Nested \$JOBUTIL Procedure

Running the Job Queue Processor \$JOBQ

The job queue processor must be running before you use it to run and control \$JOBUTIL job command procedures. You can start the job queue processor (\$JOBQ) by using the \$L operator command or by loading the \$SUBMIT utility or the \$JOBQUT utility. (Use \$SUBMIT to submit and control individual jobs and use \$JOBQUT to control the job queue.)

The following three procedures show how to start the job queue processor:

- “Starting the Job Queue Processor with \$L” shows how to start it using the \$L operator command.
- “Starting the Job Queue Processor with \$SUBMIT or \$JOBQUT” on page OP-248 shows how to start it the first time you load \$SUBMIT or \$JOBQUT.

Notes:

1. If you want the job queue processor to run in a partition other than the one in which your display terminal is running, you should start it the first time you use \$SUBMIT or \$JOBQUT.
2. The job queue processor can be running while you do other jobs on the Series/1.

Starting the Job Queue Processor with \$L

Use the following procedure to start the job queue processor (\$JOBQ) in the partition in which you display terminal is running. If you want to select another partition, use the procedure “Starting the Job Queue Processor with \$SUBMIT or \$JOBQUT” on page OP-248.

Press the attention key and enter \$L \$JOBQ to start the job queue processor in the partition in which your display terminal is running.

```
> $L $JOBQ
LOADING $JOBQ      21P,08:10:01, LP= A0000, PART=1
```

Running Programs and Job Procedures

Running the Job Queue Processor \$JOBQ (continued)

Starting the Job Queue Processor with \$SUBMIT or \$JOBQUT

Use the following procedure to start the job queue processor (\$JOBQ) the first time you load \$SUBMIT or \$JOBQUT:

Before You Start:

Determine the number of the partition in which you want the job queue processor (\$JOBQ) to run. If you are not sure which partition has room for \$JOBQ, press the enter key without typing a partition number and EDX will select a partition for you.

1. Load the \$SUBMIT utility or the \$JOBQUT utility

- by entering **\$L \$SUBMIT** or selecting option 10.2 of the session manager.
- by entering **\$L \$JOBQUT** or selecting option 10.1 of the session manager.

(Except for the utility name in the system messages, the prompts and answers are the same for both utilities.)

```
> $L $xxxxxx
LOADING $xxxxxx    21P,08:10:01, LP= A0000, PART=1

$xxxxxx - EDX JOB QUEUE xxxxxxxxxxxx PROGRAM

ENTER '?' AT ANY TIME FOR HELP

JOB QUEUE PROCESSOR ($JOBQ) MUST BE
LOADED TO CONTINUE. LOAD IT (Y/N)? _
```

2. Enter a Y to start the job queue processor.

If you do not want to start it at this time, enter an N.

```
JOB QUEUE PROCESSOR ($JOBQ) MUST BE
LOADED TO CONTINUE. LOAD IT (Y/N)? Y

PARTITION NUMBER (1-8) OR ENTER FOR ANY: _
```

3. Enter the decimal number of the partition where you want the job queue processor to run, or leave blank if you want EDX to select the partition.

```
PARTITION NUMBER (1-8) OR ENTER FOR ANY: 2

JOB QUEUE PROCESSOR LOADED INTO PARTITION 2

COMMAND (?): _
```

Running the Job Queue Processor \$JOBQ *(continued)*

4. Do you want to continue to use \$JOBQUT or \$SUBMIT?

- If so, continue with that procedure.
- If not, enter EN to end \$JOBQUT or \$SUBMIT. The job queue processor remains active.

```
COMMAND (?): EN
$xxxxxx ENDED AT 08:11:00
```


Chapter 7. Controlling Program Output

Controlling program output includes aligning and advancing the paper (forms), changing the terminal page-formatting parameters, and spooling program output. *Spooling* program output is putting it on a disk or diskette data set for printing or displaying at a later time.

Controlling printed output also includes putting paper or special forms into your printer. Because of the variety of printers that can be attached to the Series/1, this book does not show you how to load paper or change ribbons. Refer to the *Operator's Guide* or the individual hardware description manual for this information.

The first four topics are covered in this chapter under the following headings:

- “Aligning Forms” on page OP-252
- “Advancing Printer Forms” on page OP-252
- “Changing Page Formatting Parameters” on page OP-253
- “Spooling Program Output” on page OP-258

Remember that on the Series/1, printers are considered terminals. The procedure “Listing System Terminal Assignments” on page OP-44 and some of the procedures in Chapter 3, “Operating Your Display Terminal” on page OP-29, also apply to printers.

Controlling Program Output

Aligning Forms

Printer paper and special forms must be manually set to the physical top of the page and aligned left to right on the printer. Refer to the *Operator's Guide* or the individual printer description manual for instructions on aligning your paper forms.

Advancing Printer Forms

Use the following procedure to advance the forms in the system printer (\$SYSPRTR) one or more pages. It does not align the paper to print at the top of the physical page. Aligning the printer at the top of the page is a manual operation.

Notes:

1. If printer spooling is using the system printer, the \$E operator command used in this procedure has no effect.
2. The Event Driven Executive does not update the next line pointer when forms are manually advanced. Therefore, whenever you manually advance printer forms make sure they are advanced to the same line on the next page.

Press the attention key and type \$E n, where n is the number of pages to be advanced. If you do not enter the number of pages, \$E advances the forms to the top of the next page.



```
> $E 3
```

Changing Page Formatting Parameters

You can change the parameters that determine how output is printed on a printer or displayed on a display terminal screen. In the following definitions and procedures, "page" refers to printer paper, special printer forms, or a display terminal screen. These parameters are:

Attribute Specifies the display mode characteristics of the 3101 display. This parameter is specified as:

- HIGH, for high intensity display for both input and output.
- LOW, for a low or normal intensity display for both input and output.
- BLINK, for a blinking display for both input and output.
- BLANK, for no display of input or output characters.
- NO, for no attribute byte in the data stream.

Bottom margin Specifies the number of lines left blank at the bottom of the page. The bottom margin is specified as a decimal number between the value of the top margin plus history lines and page size minus 1.

Character set CHARSET - specifies which character set is to be used. This parameter applies to the 4975 model 1 and 4975 model 2 printers only. The valid character sets are:

AUGE Austria and Germany
BELG Belgium
BRZL Brazil
DNNR Denmark and Norway
FRAN France
FRCA French Canadian
INTL International (multi-national)
ITAL Italy
JAEN Japan English
KANA Japan katakana
PORT Portugal
SPAN Spain
SPNS Spanish speaking (other than Spain)
SWFI Sweden and Finland
UKIN United Kingdom
USCA United States and Canada

History lines Specifies the number of lines from the previous page that are printed at the top of the current page. The parameter applies to display terminals operating in full-screen mode only. Also, a 3101 Display Terminal cannot have history lines.

Left margin Specifies the character position where a line of printing starts. The left margin is specified as a decimal number between 0 and one less than the maximum number of characters that can fit on one line.

Lines per inch LPI - Specifies the number of lines of print per inch. LPI is specified as **6** for 6 lines per inch, or **8** for 8 lines of print per inch.

Controlling Program Output

Changing Page Formatting Parameters (*continued*)

Note: You cannot change the LPI for a printer being used by a spool writer.

- Output pause** Applies to screen devices; should be specified as N (no) for printers.
- Overflow lines** Specifies if output lines which exceed the right margin are continued on the next line. The overflow lines parameter is specified as Y (yes) or N (no).
- Page size** Specifies the maximum number of lines that one page of the form or paper can hold. This is not the number of lines that will really be printed unless the top and bottom margins are set to zero (0).
- Print density** PDN - specifies whether the output is printed in expanded (EXPD), normal (NORM), or compact (COMP) density. Density is the number of characters per inch as follows:

For the 5224, 5225, 4975 model 1 and model 2:

- Expanded density is 10 characters per inch for a total of 132 characters in the standard 13.2 inch line.
- Normal density is 15 characters per inch for a total of 198 characters in the standard 13.2 inch line.
- Compact density is 17 characters per inch for a total of 230 characters in the standard 13.2 inch line.

Note: To change the print density on a 5219 printer, you must physically change the print wheel. When the PDEN, CHARSET or RESTORE operands are coded on the TERMCTRL instruction, they cause the 5219 printer to stop printing and signal the operator. At that time, the operator may change the print wheel. The operator must then press the start button to resume printing. Refer to the *IBM Series/1 5219 Printer Models D01 and D02 Setup Procedures/Operator Guide*, GA23-1019, for information on how to change the print wheel.

- Print mode** Specifies whether the printing is done in draft or text mode. This parameter applies to the 4975 model 2 printer only. Draft mode sets uniform letter spacing as specified in the print density (PDEN) parameter. Text mode specifies a variable letter spacing based on the width of the letter (for example, an *i* is narrower than an *m*). Therefore, the number of characters is also variable. There are two text modes: TEXT and TEXT1. TEXT specifies two passes of the print head per line, for dark finished copy. TEXT1 specifies only one pass of the head, for light finished copy. TEXT1 is quicker and is usually used when you do not care about print quality for final copies.
- Right margin** Specifies the character positions where a line of printing stops. The right margin is specified as a decimal number between the left margin and the maximum number of characters that can fit on one line.
- Spoolable** Specifies whether the printer can be used for printer spooling. This parameter is specified as Y (yes) or N (no).

Changing Page Formatting Parameters (*continued*)

- Stream** specifies whether attribute bytes are in the data stream. The stream parameter is specified as **Y** (yes) or **N** (no). (See the 3101 TERMCTRL statement in the *Language Reference* for additional explanation of this parameter.)
- Top margin** Specifies the number of lines left blank at the top of the page before the printing starts. The top margin is specified as a decimal number between 0 and page size minus 1.

Use the following procedure to change one or more page formatting parameters for a printer:

Before You Start:

Obtain the following information:

- The name of the printer. (See “Listing System Terminal Assignments” on page OP-44 if you do not know the name of the printer.)
- The new values for the parameters you are changing. You only have to enter values for the parameters you are changing. If you want to keep the current value, leave it blank and press the enter key. These parameters are:
 - Page Size
 - Top margin
 - Bottom margin
 - History lines
 - Left margin
 - Right margin
 - Overflow lines (Yes/No)
 - Output pause (Yes/No)
 - Spoolable (Yes/No)
 - Lines per inch (6/8)
 - 4975-1, or 4975-2 print density (EXPD/NORM/COMP)
 - 4975-2 print mode (DRAFT/TEXT1/TEXT)
 - Character set (4975-1 or 4975-2)
 - 3101 attribute (HIGH/LOW/BLINK/BLANK/NO)
 - 3101 stream (Yes/No).

1. Load the utility \$TERMUT1:

- Using the \$L operator command, press the attention key and type **\$L \$TERMUT1**.
- Using the session manager, select option 4.1.

```
> $L $TERMUT1
LOADING $TERMUT1      34P,08:30:29, LP= 00...
*** TERMINAL CONFIGURATOR ***
COMMAND (?): _
```

Controlling Program Output

Changing Page Formatting Parameters (*continued*)

2. Enter CT to change the page formatting parameters.

Note: Use the CT command to modify parameters, but be sure the printer is turned on. If it is not, any changes made will not take effect and \$TERMUT1 enters a wait state. You must then turn the printer on and reenter the CT command.

```
COMMAND (?): CT
ENTER TERMINAL NAME: _
```

3. Enter the name of the terminal. (If you are changing the page formatting parameters for a 3101 Display Terminal, skip to step 4(a).)

```
ENTER TERMINAL NAME: $SYSRTR
PAGE SIZE (NOW IS 24):
```

4. \$TERMUT1 displays the example prompts, one at a time. Enter the new value, or leave blank and press the enter key to keep the current value. (If you are changing the parameters on a display terminal, you will not receive the LPI prompt.)

```
PAGE SIZE (NOW IS 24):
VALUE NOT CHANGED
TOP MARGIN (NOW IS 0): 5
BOTTOM MARGIN (NOW IS 23): 20
HISTORY LINES (NOW IS 0):
VALUE NOT CHANGED
LEFT MARGIN (NOW IS 0): 5
RIGHT MARGIN (NOW IS 131): 125
OVERFLOW LINES (NOW IS N): Y
OUTPUT PAUSE (NOW IS N):
VALUE NOT CHANGED
SPOOLABLE (Y/N) (NOW IS Y): N
LPI (6/8) (NOW IS 6): 8
```

- 4(a). \$TERMUT1 displays the example prompts for a 3101 Display Terminal, one at a time. Enter the new value, or leave blank and press the enter key to keep the current value.

```
OVERFLOW LINES (NOW IS N):
VALUE NOT CHANGED
OUTPUT PAUSE (NOW IS N): Y
SPOOLABLE (Y/N) (NOW IS Y): N
ATTRIBUTE (HIGH/LOW/BLINK/BLANK/NO) (NOW IS LOW): HIGH
STREAM (Y/N) (NOW IS Y): N
```

Changing Page Formatting Parameters *(continued)*

5. Do you have a 4975-1, 4975-2, or 5219 printer?

- If you do not, continue with step 6.
- If you have a 4975-1 or 5219, continue with step 5(a).
- If you have a 4975-2 continue with step 5(b).

5(a). If you are changing the parameters for a 4975-1 or 5219, \$TERMUT1 also asks if you want to change the print density or character set. Enter the new values, or leave blank to keep the current ones, and continue with step 6. Enter a ? to determine what the valid character set abbreviations are.

```
PDEN (EXPD/NORM/COMP) (NOW IS EXPD): NORM
CHAR SET (? FOR LIST) (NOW IS USCA):
                               VALUE NOT CHANGED
COMMAND (?): _
```

5(b). If you are changing the parameters for a 4975-2, \$TERMUT1 also asks if you want to change the print density, print mode, or character set. Enter the new values, or leave blank to keep the current ones, and continue with step 6. Enter a ? to determine what the valid character set abbreviations are.

```
PDEN (EXPD/NORM/COMP) (NOW IS EXPD): NORM
PMODE (DRAFT/TEXT1/TEXT) (NOW IS TEXT): DRAFT
CHAR SET (? FOR LIST) (NOW IS USCA):
                               VALUE NOT CHANGED
COMMAND (?): _
```

6. Do you want to change the parameters on another printer?

- If so, continue with step 2.
- If not, type EN to end \$TERMUT1.

```
COMMAND (?): EN
$TERMUT1 ENDED AT 08:55:35
```

Controlling Program Output

Spooling Program Output

Spooling program output is requesting that it be written to a disk or diskette data set. The output is then printed or displayed at a later time. Spooling is controlled using the Event Driven Executive spooling facility (\$SPOOL).

The spooling facility is usually used for output that is to be printed, but it can be used for output that is to be displayed on a display terminal. Printed output is spooled because most programs can generate output faster than a printer can print it. Spooling the output allows the program to finish sooner, freeing up its storage space for another program. Spooling is run independently of other jobs being run on your Series/1. In other words, spooling can be running at the same time other programs (application, system, or utility) are running, although other programs should not use the same output devices while spooling is running.

Program output can also be spooled by an application program. This subject is covered in the *Event Driven Executive Language Programming Guide*. However, an application program does not have to have spool instructions for its output to be spooled.

The following sections explain the concepts of spooling program output and have procedures for operating the spool facility.

- “Printer Spooling Terms”
- “Preparing to Run Spool” on page OP-260
- “Setting Spool Options” on page OP-261
- “Running the Spool Facility” on page OP-273

Printer Spooling Terms

The following terms are used in this description of spooling:

Control record An optional print control record that allows an application program to control the disposition and printing or displaying of spooled output.

Forms code Specifies a specific printer form that is to be used when the related spool job is printed. The forms code is part of the control record. If a forms code is specified for individual jobs, the spool writer prints all jobs with the same forms code in sequence. The writer does not alternate between form codes.

Restart Specifies whether \$SPOOL prints or deletes the contents of the spool data set when the spool facility is started. If restart is yes (Y), \$SPOOL prints any jobs left from the previous session. If restart is no (N), \$SPOOL deletes any jobs left from the previous session.

Spool data set A data set used to hold spooled output and control information before printing or displaying.

Spooling Program Output (*continued*)

- Spool device** Any printer or display terminal that has been designated as a spool output device. Program output directed to the spool device is spooled to the spool data set and printed or displayed by the spool facility at a later time. The name of a spool device is the symbolic system name of the device shown in the listing generated by the procedure "Listing System Terminal Assignments" on page OP-44.
- Spool facility** The Event Driven Executive program output spooling support programs (\$SPOOL).
- Spool group** A specific number of records (spool group size) allocated at one time to a spool job from the spool data set. The size of the spool group determines the number of spool groups in the spool data set.
- Spool job** The set of output records (spool groups) generated by a program and directed to a spool device. The status of a spool job can be any of the following:
- Active** The spool job is being created by an application program or utility.
 - Ready** The spool job has been created and is ready for printing or displaying by a spool writer.
 - Printing** A spool writer is printing or displaying the spool job.
 - Held** The spool job is held by the operator and not available for printing or displaying.
 - Incomplete** The spool job is incomplete because of abnormal system termination.
 - Kept** The spool job is not to be deleted after it is printed. It will not be deleted until you tell the spool facility to delete it.
- Spool session** The time that the spool facility is running between one load (\$L \$SPOOL) and end (\$S STOP) of the spool facility.
- Spool writer** An EDX program which prints spool groups on a specific spool device. The name of a spool writer is the same as the spool device name.

Controlling Program Output

Spooling Program Output (*continued*)

Preparing to Run Spool

The following items will guide you through the process of preparing to run the spool facility and running it.

1. To use the spool facility, it must have been included in your system during system generation. You can verify if it was included by checking your system generation records or trying to load \$SPOOL as described under "Starting the Spool Facility" on page OP-273.
2. Designate the spool device(s). Both EDX and the spool facility must recognize the output device as a spool device.
 - EDX must know that the device is spoolable. If it was not specified as such during system generation, you must use the \$TERMUT1 utility to do so. (See "Changing Page Formatting Parameters" on page OP-253 for the procedure.)
 - The spool facility must know that the device is a spool device. When received from IBM, the spool facility is set up to recognize the system printer named \$SYSRTR, as a spool device. If you are using other output devices for spooling, you must use the \$SPLUT1 utility to designate them as spool devices. (See "Specify Spool Devices" on page OP-271 for the procedure.)

Note: The program function key for printing a terminal screen (usually PF6) and the \$E operator command are not recognized while spooling is active with \$SYSRTR as a spool device.

3. The spool data set must exist and it must be large enough to contain the spooled data. If you name the data set SPOOL and put it on volume EDX003, you will not have to enter the name on commands referencing the data set.

The maximum data set size is determined by the number of spool jobs in the system and the number of lines of output in each job. Output can accumulate in the spool data set rapidly, so we suggest you start with a data set of 8510 records and adjust it as required for your installation. If the spool data set fills up, following programs will wait until space is available in the data set. (See "Estimating Spool Data Set Size" on page OP-395 for formulas for calculating the spool data set size for compiler, assembler, and linkage editor output listings, and "Allocate a Data Set" on page OP-124 for the allocation procedure.)

4. Set up the spool facility options. The options do not have to be set every time you start the spool facility because it remembers how you had them set the last time it was used. You can verify the current settings by loading the utility \$SPLUT1 because it displays the current settings of the spool options each time it is loaded. (See "Starting the Spool Facility" on page OP-273 for instructions on loading \$SPLUT1.) The main options are:
 - The spool data set and volume names must be the same as the spool data set you have allocated. (See "Specify the Spool Data Set" on page OP-264 to set this name.)

Spooling Program Output (*continued*)

- The spool restart mode must be set for the condition you require. If restart mode is on, any jobs from a previous session will be printed when you start this spool session. If restart mode is off, any jobs from a previous session will be deleted. (See “Set the Restart Option” on page OP-262 to set the restart mode.)
- Auto start mode must be set for the condition you require. If it is Y, the writer for the related device starts automatically when the spool facility is started. The auto start option is set when you specify a spool device. Auto start is set as part of the procedure “Specify Spool Devices” on page OP-271.

The remainder of this chapter contains procedures for setting the spool options and running the spool facility.

Setting Spool Options

The following procedures set the various spool options:

- “Set the Restart Option” on page OP-262 is used to specify whether jobs left in the spool data set from a previous session are printed or deleted the next time the spool facility is started.
- “Specify the Spool Data Set” on page OP-264 is used to set or change the data set and volume names of the spool data set.
- “Set the Maximum Jobs Option” on page OP-266 is used to specify the maximum number of spool jobs that can be in the system at one time.
- “Set the Maximum Active Jobs Option” on page OP-267 is used to specify the maximum number of spool jobs that can be active at one time.
- “Set Spool Groups and Group Size Options” on page OP-268 is used to specify the number of records in each spool group. The number of records in each group and the size of the spool data set determines the number of groups the data set holds.
- “Set the Separator Page Option” on page OP-270 is used to specify whether a separator page showing the job name is printed between spool jobs.
- “Specify Spool Devices” on page OP-271 is used to specify the names of the spool devices. This procedure is also used to set the autostart option and specify the forms code. Autostart determines whether the writer for the device is automatically started when the spool facility is started. The forms code is used to send the operator a message that a special form should be in the printer.

Controlling Program Output

Spooling Program Output (*continued*)

Set the Restart Option

Use the following procedure to set the restart option. The restart option determines whether jobs left from a previous session are printed or displayed when \$SPOOL is loaded.

Set restart mode off (N) to delete all previous jobs from the spool data set. Restart should be set off:

- The first time a spool data set is used.
- To delete any jobs left over from a previous spool session.

Set the restart mode on (Y) to start spooling using a previously used spool data set. Restart should be set on to:

- Ensure that any data left from a previous session is not lost.
- Restart spooling to retrieve the output spooled up to the last checkpoint before a system failure. Checkpoint data is written to the spool data set each time a group of records is allocated to a spool job, each time a spool job becomes ready or is deleted, and each time a \$\$ operator command is processed.
- Transfer a spool data set on diskette from another Series/1.

Notes:

1. The specified restart mode remains in effect for all following spool sessions.
2. Restart mode must be set off (N) before you change any spooling capacity parameters.
3. You cannot set restart mode on (Y) during the same \$SPLUT1 session that you change any spooling capacity parameters (CD, DS, GS, MA, MJ, and SO commands). If you do, the changes will not take effect.
4. The changes take effect in the next spooling session.

Before You Start:

Observe the following requirements:

- If you are setting restart on (Y) you need the data set and volume names of the spool data set, unless you are using the currently specified data set.
- If you have changed any spooling capacity parameters (CD, DS, GS, MA, MJ, or SO), you must restart spooling before setting the restart option back on. Otherwise, the new capacity specifications will not take effect.

To avoid losing spool jobs, do the following:

1. Print any spool jobs that are in the spool data set.
2. Stop and restart \$SPOOL.

Spooling Program Output *(continued)*

1. Load the utility \$SPLUT1: (The example screen at the right shows only the fields of the sign-on screen that are applicable to this procedure. For an explanation of the entire screen, see “Spool Status Display Formats” on page OP-391.)

- Using the \$L operator command, press the attention key and enter \$L \$SPLUT1.
- Using the session manager, select option 4.7.

```
> $L $SPLUT1
LOADING $SPLUT1      33P, 08:30:35 LP= 00...

*****
**                               EDX SPOOL UTILITY
**  CHANGES EFFECTIVE THE NEXT TIME $SPOOL...
*****

DSNAME...           ...RESTR...
SPOOL...            N

COMMAND (?): _
```

2. Enter RS to change the restart parameter.

```
COMMAND (?): RS

RESTART Y/N? _
```

3. Do you want to set the restart option on?

- If so, enter a Y. This retains the current contents of the spool data set when \$SPOOL is loaded.
- If not, enter an N. This deletes any jobs remaining in the spool data set when spool is loaded. Skip to step 6 on page OP-264.

```
RESTART Y/N? Y
ALL PREVIOUS CHANGES IGNORED
RESTART TO SPOOL,EDX003 ? _
```

4. Do you want to restart to the listed data set?

- If so, enter a Y and skip to step 6 on page OP-264 .
- If not, enter an N and continue with 5 on page OP-264.

```
RESTART TO SPOOL,EDX003 ? N
ENTER (NAME,VOLUME): _
```

Controlling Program Output

Spooling Program Output (*continued*)

5. Enter the data set and volume name of the spool data set you are using.

```
ENTER (NAME,VOLUME): MYSPool,MYVOL
NO FURTHER CHANGES ALLOWED
DSNAME...      ...RESTRT...
SPOOL...      Y
COMMAND (?): _
```

6. End \$SPLUT1:

- Enter CA to cancel the utility, ignoring any changes made during this session.
- Enter EN to end the utility, recording the restart mode change. It will be effective the next time \$SPOOL is loaded.

```
COMMAND(?): EN
$SPLUT1 ENDED AT 09:15:30
```

Specify the Spool Data Set

Use the following procedure to specify the data set and volume names of the spool data set. This does not allocate the data set. It must be allocated using the procedure "Allocate a Data Set" on page OP-124.

Before You Start:

Observe the following requirements:

- You need the data set and volume names of the new spool data set before starting this procedure.
- The restart option must be set off. See "Set the Restart Option" on page OP-262 for the procedure.

Spooling Program Output (*continued*)

1. Load the utility `$$SPLUT1`:
 - Using the `$L` operator command, press the attention key and enter `$L $$SPLUT1`.
 - Using the session manager, select option 4.7.

(The example screen at the right shows only the fields of the sign-on screen that are applicable to this procedure. For an explanation of the entire screen, see "Spool Status Display Formats" on page OP-391.)

```
> $L $$SPLUT1
LOADING $$SPLUT1      33P, 08:30:35 LP= 00...

*****
**                               EDX SPOOL UTILITY
** CHANGES EFFECTIVE THE NEXT TIME $$SPOOL...
*****

DSNAME---VOLUME...

SPOOL      EDX003...

COMMAND (?): _
```

2. Enter `DS` to change the spool data set.

```
COMMAND (?): DS
ENTER(NAME,VOLUME): _
```

3. Enter the data set and volume name of the new spool data set.

```
ENTER(NAME,VOLUME): NEWSPL,MYVOL
DSNAME---VOLUME...
NEWSPL  MYVOL...

COMMAND (?): _
```

4. If you are changing other parameters, continue with that procedure. Otherwise, end `$$SPLUT1`:

- Enter `CA` to cancel the utility, ignoring any changes made during this session.
- Enter `EN` to end the utility, recording all changes made during this session. These changes will be effective the next time `$$SPOOL` is loaded.

```
COMMAND (?): EN
$$SPLUT1 ENDED AT 09:30:25
```

Controlling Program Output

Spooling Program Output (*continued*)

Set the Maximum Jobs Option

Use the following procedure to change the maximum number of spool jobs allowed at one time.

Before You Start:

Observe the following requirements:

- You need the decimal new number of maximum jobs before starting this procedure.
- The restart option must be set off. See “Set the Restart Option” on page OP-262 for the procedure.

1. Load the utility \$SPLUT1:

- Using the \$L operator command, press the attention key and enter \$L \$SPLUT1.
- Using the session manager, select option 4.7.

(The example screen at the right shows only the fields of the sign-on screen that are applicable to this procedure. For an explanation of the entire screen, see “Spool Status Display Formats” on page OP-391.)

```
> $L $SPLUT1
LOADING $SPLUT1      33P, 08:30:35 LP= 00...
*****
**                      EDX SPOOL UTILITY
** CHANGES EFFECTIVE THE NEXT TIME $SPOOL...
*****
DSNAME...           ...MAXJOBS...
SPOOL...            ...10...

COMMAND (?): _
```

2. Enter MJ to change the maximum number of spool jobs.

```
COMMAND (?): MJ
MAX JOBS: _
```

3. Enter the new number of maximum jobs.

```
MAX JOBS: 14
DSNAME...           ...MAXJOBS...
SPOOL...            ...14...

COMMAND (?): _
```

Spooling Program Output (*continued*)

4. If you are changing other parameters, continue with that procedure. Otherwise, end \$SPLUT1:
 - Enter CA to cancel the utility, ignoring any changes made during this session.
 - Enter EN to end the utility, recording all changes made during this session. These changes will be effective the next time \$SPOOL is loaded.

```
COMMAND (?): EN
$SPLUT1 ENDED AT 09:30:25
```

Set the Maximum Active Jobs Option

Use the following procedure to change the maximum number of active spool jobs allowed at one time.

Before You Start:

Observe the following requirements:

- You need the new number of maximum active jobs before you start this procedure.
- The restart option must be set off. See “Set the Restart Option” on page OP-262 for the procedure.

1. Load the utility \$SPLUT1:

- Using the \$L operator command, press the attention key and enter \$L \$SPLUT1
- Using the session manager, select option 4.7.

(The example screen at the right shows only the fields of the sign-on screen that are applicable to this procedure. For an explanation of the entire screen, see “Spool Status Display Formats” on page OP-391.)

```
> $L $SPLUT1
LOADING $SPLUT1      33P, 08:30:35 LP= 00...
*****
**                      EDX SPOOL UTILITY
** CHANGES EFFECTIVE THE NEXT TIME $SPOOL...
*****

DSNAME...           ...MAXACTV...

SPOOL...             ...3...

COMMAND (?): _
```

2. Enter MA to change the maximum number of active jobs.

```
COMMAND (?): MA
MAX ACTIVE JOBS: _
```

Controlling Program Output

Spooling Program Output (*continued*)

3. Enter the new number of maximum active jobs.

```
MAX ACTIVE JOBS: 5
DSNAME...          ...MAXACTV...
SPOOL...           ...5...

COMMAND (?): _
```

4. If you are changing other parameters, continue with that procedure. Otherwise, end \$SPLUT1:

- Enter **CA** to cancel the utility, ignoring any changes made during this session.
- Enter **EN** to end the utility, recording all changes made during this session. These changes will be effective the next time \$SPOOL is loaded.

```
COMMAND (?): EN
$SPLUT1 ENDED AT 09:30:25
```

Set Spool Groups and Group Size Options

Use the following procedure to change the number of records in each spool group. A spool group can have from 1 to 255 records. (The default group size is 100 records.)

Changing the group size alters the space and performance characteristics of spooling. A large group size can improve performance because fewer groups will need to be allocated. A small group size can improve disk space efficiency because there is potentially less unused space. The best group size for you depends on the average size of the spool jobs. Decrease the group size to minimize unused disk space, and increase it to improve performance.

The spool facility divides the group size into the total number of records in the spool data set to determine the maximum number of groups. Therefore, changing the spool group size may also change the maximum number of spool groups.

Before You Start:

Observe the following requirements:

- You need the number of records you want in each spool group before you start this procedure.
- The restart option must be set off. See “Set the Restart Option” on page OP-262 for the procedure.

Spooling Program Output (*continued*)

1. Load the utility \$SPLUT1:

- Using the \$L operator command, press the attention key and enter \$L \$SPLUT1.
- Using the session manager, select option 4.7.

(The example screen at the right shows only the fields of the sign-on screen that are applicable to this procedure. For an explanation of the entire screen, see “Spool Status Display Formats” on page OP-391.)

```
> $L $SPLUT1
LOADING $SPLUT1      33P, 08:30:35 LP= 00...

*****
**                               EDX SPOOL UTILITY
** CHANGES EFFECTIVE THE NEXT TIME $SPOOL...
*****

DSNAME...           ...GRPS-GRPSZ...

SPOOL...            ... 10  100...

COMMAND (?): _
```

2. Enter GS to change the spool group size.

```
COMMAND (?): GS
PHYSICAL RECORDS PER GROUP (1-255): _
```

3. Enter the number of records per group.

```
PHYSICAL RECORDS PER GROUP (1-255): 50
DSNAME...           ...GRPS-GRPSZ...
SPOOL...            ...20  50...

COMMAND (?): _
```

4. If you are changing other parameters, continue with that procedure. Otherwise, end \$SPLUT1:

- Enter CA to cancel the utility, ignoring any changes made during this session.
- Enter EN to end the utility, recording all changes made during this session. These changes will be effective the next time \$SPOOL is loaded.

```
COMMAND (?): EN
$SPLUT1 ENDED AT 09:30:25
```

Controlling Program Output

Spooling Program Output (*continued*)

Set the Separator Page Option

Use the following procedure to add or delete the optional separator page between spool jobs.

1. Load the utility \$SPLUT1: (The example screen at the right shows only the fields of the sign-on screen that are applicable to this procedure. For an explanation of the entire screen, see "Spool Status Display Formats" on page OP-391.)

- Using the \$L operator command, press the attention key and enter \$L \$SPLUT1.
- Using the session manager, select option 4.7.

```
> $L $SPLUT1
LOADING $SPLUT1      33P, 08:30:35 LP= 00...

*****
**                               EDX SPOOL UTILITY
** CHANGES EFFECTIVE THE NEXT TIME $SPOOL...
*****

DSNAME...           ...SEP...
SPOOL...            ...N...

COMMAND (?): _
```

2. Enter the command SO.

```
COMMAND (?): SO
SET SEPARATOR PAGE OPTION (Y/N is now N): _
```

3. Do you want a separator page?

- If so, enter a Y.
- If not, enter an N.

```
SET SEPARATOR PAGE OPTION (Y/N is now N): Y
DSNAME...           ...SEP...
SPOOL...            ...Y...

COMMAND (?): _
```

4. If you are changing other parameters, continue with that procedure. Otherwise, end \$SPLUT1L:

- Enter CA to cancel the utility, ignoring any changes made during this session.
- Enter EN to end the utility, recording all changes made during this session. These changes will be effective the next time \$SPOOL is loaded.

```
COMMAND (?): EN
$SPLUT1 ENDED AT 09:30:25
```

Spooling Program Output (*continued*)

Specify Spool Devices

Use the following procedure to specify the devices that are used for printer spooling. Any printer or display terminal can be used as a spool output device, if they are specified as spoolable devices. (See “Changing Page Formatting Parameters” on page OP-253 to configure the device as spoolable.)

Before You Start:

Observe the following requirements:

- You need the names by which the system knows the devices (See “Listing System Terminal Assignments” on page OP-44 for the procedure to list the names of the output devices attached to your Series/1.)
- The restart option must be set off. See “Set the Restart Option” on page OP-262 for the procedure.

1. Load the utility \$SPLUT1: (The example screen at the right shows only the fields of the sign-on screen that are applicable to this procedure. For an explanation of the entire screen, see “Spool Status Display Formats” on page OP-391.)

- Using the \$L operator command, press the attention key and enter \$L \$SPLUT1.
- Using the session manager, select option 4.7.

```
> $L $SPLUT1
LOADING $SPLUT1      33P, 08:30:35 LP= 00...

*****
**                               EDX SPOOL UTILITY
** CHANGES EFFECTIVE THE NEXT TIME $SPOOL...
*****

DSNAME...  ...DEVICES-AUTO-FORM
SPOOL...   ...$SYSRTR  N

COMMAND (?): _
```

2. Enter CD to change the current spool device or specify multiple spool devices.

```
COMMAND (?): CD
DEVICE NAME (ENTER BLANK TO END): _
```

3. Enter the device name, or, if you do not want to continue, leave blank, press the enter key, and continue with step 6 on page OP-272.

```
DEVICE NAME (ENTER BLANK TO END): $SYSRTR
WRITER AUTOSTART ? (Y/N): _
```

Controlling Program Output

Spooling Program Output (*continued*)

4. Do you want a spool writer started for this device when \$SPOOL is loaded? (If a writer is started when \$SPOOL is loaded, the first spool job sent to the writer will start printing when the job is generated. If a writer is not started, you will have to start it at a later time using the \$\$ \$WSTR command.)

- If so, enter a Y.
- If not, enter an N and continue with step 3 on page OP-271.

```
WRITER AUTOSTART ? (Y/N): Y
ENTER FORMS CODE: _
```

5. Enter the code for the form that is used. If you do not have a particular form, leave blank and press the enter key. You must enter the names of all devices being used for spooling. Continue with step 3 on page OP-271 to add another device, or leave blank and press the enter key if you are finished.

```
ENTER FORMS CODE: FM35
DEVICE NAME (ENTER BLANK TO END): _
```

6. If you are changing other parameters, continue with that procedure. Otherwise, end \$SPLUT1:

- Enter CA to cancel the utility, ignoring any changes made during this session.
- Enter EN to end the utility, recording all changes made during this session. These changes will be effective the next time \$SPOOL is loaded.

```
DSNAME... ..DEVICES-AUTO-FORM
SPOOL... ..$SYSPRTR Y FM35
...MPRTR N
COMMAND (?): EN
$SPLUT1 ENDED AT 09:30:25
```

Spooling Program Output (*continued*)

Running the Spool Facility

The following procedures control the running of the spool facility, individual jobs, and writers:

- “Starting the Spool Facility” is used to start the spool facility, \$SPOOL.
- “Stopping the Spooling Facility” on page OP-274 is used to stop the spool facility, \$SPOOL.
- “Displaying Spool Status Information” on page OP-275 is used to print or display the status of one spool job, selected spool options, or all jobs.
- “Altering Spool Job Printing Parameters” on page OP-276 is used to change spool jobs as follows:
 - Change the number of copies printed
 - Change the forms code
 - Change the name used on the separator page
 - Redirect one or all jobs to a new output device
 - Specify forms alignment.
- “Starting a Spool Writer” on page OP-278 is used to start an individual spool writer or restart one that was permanently stopped at the end of a job. You can also specify a new forms code when you restart the writer.
- “Stopping a Spool Writer” on page OP-279 is used to stop an individual spool writer.
- “Restarting a Spool Writer” on page OP-280 is used to restart a spool writer that was temporarily stopped. You can also specify where the current job is restarted and a new forms code.
- “Holding Spool Jobs” on page OP-282 is used to hold active or ready jobs so they will not be printed or displayed.
- “Releasing Held Jobs” on page OP-283 is used to release for output jobs that are in the held status.
- “Keeping and Releasing Kept Jobs” on page OP-284 is used to keep a copy of a job after it is printed or displayed. It is also used to delete a job that was kept. When the job is deleted, one final copy is printed.
- “Deleting Spool Jobs” on page OP-285 is used to delete ready or printing spool jobs.
- “Responding to Spool Forms Alignment Prompt” on page OP-286 is used to respond to a request to verify that the printer forms are properly set up in the printer.

Controlling Program Output

Spooling Program Output (*continued*)

Starting the Spool Facility

Use the following procedure to start the spool facility. It should not be started through the Session Manager because the Session Manager will not receive control again until after spooling ends, keeping you from using the display terminal until spooling ends.

Press the attention key and type
\$L \$SPOOL,EDX002.

```
> $L $SPOOL,EDX002
LOADING $SPOOL      26P,08:30:24, LP=0000, PART= 2
SPOOL INITIALIZATION COMPLETE
$$SYSPRTR WRITER STARTED
```

Stopping the Spooling Facility

Use the following procedure to stop the spooling facility. The spooling facility stops when any jobs in an active or printing status finish. Ready jobs are not printed, and no new jobs are accepted.

Press attention and type **\$\$ STOP.**

```
> $$ STOP
SPOOL STOP PENDING
$$SPOOL ENDED AT 08:30:45
```

Spooling Program Output (*continued*)

Displaying Spool Status Information

Use the following procedure to display the status information for one job, for the spool resources, or for all jobs and the resources. (See “Spool Status Display Formats” on page OP-391 for an explanation of the display formats.)

Before You Start:

Determine the ID of the spool job if you are displaying the status of only one job. If you do not know what the ID is, type ALL in step 2 and the system displays the status of all spool jobs.

1. Press the attention key and type
\$\$ DISP.

```
> $$ DISP  
JOBID/ALL/STAT: _
```

2. Which status information are you displaying?

```
JOBID/ALL/STAT: STAT  
JOBS: 1( 10) ACTV: 1( 4) GRPS: 1( 2) DS=SPPOOL ,EDX003
```

- Enter **nnn**, where nnn is the number of the job, to display the status of one job.
- Enter **ALL** to display the status of all spool jobs, writers, and resources.
- Enter **STAT** to display the status of the spool resources. (See “Spool Status Display Formats” on page OP-391 for an explanation of this display format.)

Controlling Program Output

Spooling Program Output (*continued*)

Altering Spool Job Printing Parameters

Use the following procedure to change the printing or displaying of spool jobs by:

- Changing the number of copies printed.
- Changing the forms code for the job.
- Changing the job name used on the spool job separator page.
- Redirecting one spool job to a different output device.
- Redirecting all spool jobs from one output device to a another.
- Specifying that forms alignment be verified before a job is printed or displayed.

Note: Spool jobs specified for a 4975-01A, must be printed on a 4975-01A. Jobs specified for other printers cannot be redirected to the 4975-01A.

Before You Start:

Depending on which parameters you are altering, obtain one or more of the following:

- The numeric spool job ID. See “Displaying Spool Status Information” on page OP-275 to determine this number.)
- The number of copies.
- The forms code.
- The separator page job name.
- The current writer name.
- The new writer name.

1. Press the attention key and type **\$\$ ALT**.

```
> $$ ALT  
JOBID/WRIT: _
```

2. Do you want to change printing for one job or redirect all spool jobs to another printer?

- Enter the 1 to 3 digit job ID to change printing of one job. Continue with step 3(a).
- Enter **WRIT** to redirect all spool jobs to another job. Continue with step 3(b).

Spooling Program Output (*continued*)

3(a). Which parameter are you altering? (If you are altering a parameter, this is the last step of this procedure.)

- Enter **COPY n**, where *n* is the number of pages to be printed.
- Enter **FORM code**, where *code* is the new forms code.
- Enter **NAME heading**, where *heading* is the new separator page heading.
- Enter **WRIT name**, where *name* is the name of the new spool writer.
- Enter **ALIGN Y** if you want the spooling facility to verify forms alignment before proceeding with the job.
- Enter **ALIGN N** to turn off current forms alignment.

```
JOBID/WRIT: 3  
COPY/FORM/NAME/WRIT/ALIGN: COPY 3  
JOB ALTERED
```

3(b). Enter the name of the current spool writer you are reassigning.

```
JOBID/WRIT: WRIT  
CURRENT WRITER: $$SYSPRTR  
NEW WRITER: _
```

4. Enter the name of the new writer for the jobs assigned to the current writer.

```
NEW WRITER: MPRTR  
JOB(S) ALTERED
```

Controlling Program Output

Spooling Program Output (*continued*)

Starting a Spool Writer

Use the following procedure to start a spool writer to a spool device so spooled output can be printed or displayed on the device. (See "Specify Spool Devices" on page OP-271 to specify automatic start of a spool writer.)

Before You Start:

Observe the following requirements:

- You need the name of the device for which you are starting the writer. The writer name and the device name are the same. (See "Listing System Terminal Assignments" on page OP-44 if you do not know the name of the device.)
- You need the 4-character forms code, if you are using one.
- \$SPOOL must be running. (See "Starting the Spool Facility" on page OP-273 to start printer spooling.)

1. Press the attention key and type **\$S WSTR**.

```
> $S WSTR  
WRITER NAME: _
```

2. Enter the name of the spool writer you are starting. (It is the same name as the device.)

```
WRITER NAME: MPRTR  
FORMS: _
```

3. Enter the 1 to 4 character forms code representing the special forms used with this writer. If there are none, leave blank and press the enter key.

```
FORMS: FM01  
WRITER STARTED
```

Spooling Program Output (*continued*)

Stopping a Spool Writer

Stopping a spool writer stops output at the associated spool device. You can stop a spool writer:

- Immediately at the start of the next line of the spool job or at the end of the current job.
- Temporarily, to be restarted with the `$$ WRES` command, or permanently. A spool writer that is permanently stopped is restarted with the `$$ WSTR` command.

Use the following procedure to stop a spool writer. It must be entered in single-line format. The following four procedures show the four combinations for stopping a spool writer.

Before You Start:

Determine the name of the spool writer before using one of the following procedures. The writer name is the same as the name of the device to which it writes. (See “Listing System Terminal Assignments” on page OP-44 if you do not know the name of the device.)

Stopping a Spool Writer Immediately but Temporarily

Press the attention key and type `$$ WSTP name` where **name** is the name of the spool writer. The writer can be restarted using the `$$ WRES` command as described under “Restarting a Spool Writer” on page OP-280.

```
> $$ WSTP MPRTR  
WRITER STOPPED
```

Stopping a Spool Writer Immediately and Permanently

Press the attention key and type `$$ WSTP name IMM TERM`, where **name** is the name of the spool writer. You must use the `$$ WSTR` command as described under “Starting a Spool Writer” on page OP-278 to start a writer that has been permanently stopped.

```
> $$ WSTP MPRTR IMM TERM  
WRITER STOPPED
```

Controlling Program Output

Spooling Program Output (*continued*)

Temporarily Stopping a Writer at the End of the Current Job

Press the attention key and type **\$\$ WSTP name JOB NOTERM**, where **name** is the name of the spool writer. The writer can be restarted using the **\$\$ WRES** command as described under "Restarting a Spool Writer."

```
> $$ WSTP MPRTR JOB NOTERM  
WRITER STOPPED
```

Permanently Stopping a Writer at the End of the Current Job

Press the attention key and type **\$\$ WSTP name JOB TERM**, where **name** is the name of the spool writer. You must use the **\$\$ WSTR** command as described under "Starting a Spool Writer" on page OP-278 to start a writer that has been permanently stopped.

```
> $$ WSTP MPRTR JOB TERM  
WRITER STOPPED
```

Restarting a Spool Writer

Use the following procedure to restart a spool writer that has been temporarily stopped with the **\$\$ WSTP** command. You can restart the writer to:

- Start at the beginning of the interrupted job.
- Start at the line following the last line printed.
- Start at a specified number of lines or pages before or after the last line printed.

When the restarted writer finishes printing or displaying the current job, it continues with another waiting job.

Before You Start:

Obtain the following information: need:

- The name of the writer being restarted. The writer name is the same as the device name.
- The forms code for the output, if one is used.
- The location where you want spool to start. (See step 3.)

Spooling Program Output (*continued*)

1. Press the attention key and type **\$\$ WRES**.

```
> $$ WRES  
WRITER NAME: _
```

2. Enter the name of the spool writer.

```
WRITER NAME: MPRTR  
IMM/JOB/B/F: _
```

3. At what point do you want to restart?

- Enter **IMM** to start at the next line.
- Enter **JOB** to start at the beginning of the interrupted job.
- Enter **B nnn L** to start *nnn lines before* the end of the last line printed.
- Enter **B nnn P** to start *nnn pages before* the end of the last line printed.
- Enter **F nnn L** to start *nnn lines after* the end of the last line printed.
- Enter **F nnn P** to start *nnn pages after* the end of the last line printed.

```
IMM/JOB/B/F: B 100 L  
FORMS: _
```

4. Enter the forms code for this writer. If there is none, leave blank and press the enter key.

```
FORMS: FM01  
WRITER RESTARTED
```

Controlling Program Output

Spooling Program Output (*continued*)

Holding Spool Jobs

Use the following procedure to specify that active or ready jobs are to be held from printing. You can hold all jobs or individual jobs. You cannot hold a job that is already printing, it must be stopped or deleted.

Before You Start:

Determine the ID of the specific spool job you wish to hold.

Note: If you wish to hold all jobs you do not need an ID. (See "Displaying Spool Status Information" on page OP-275 to determine the jobID.)

1. Press the attention key and type **\$\$ HOLD**.

```
> $$ HOLD  
JOBID: _
```

2. Are you holding one job, or all jobs?

- Enter the 1 to 3 character spool job ID to hold a single job.
- Enter **ALL** to hold all spool jobs.

```
JOBID: 1  
JOB(S) HELD
```

Spooling Program Output (*continued*)

Releasing Held Jobs

Use the following procedure to release jobs that have been held using the **\$\$ HOLD** command. You can release one job or all jobs. A released job resumes its place in the ready queue; that is, its print order is still determined by the order in which it originally became ready.

Before You Start:

Determine the ID of the specific job you wish to release.

Note: If you wish to release all jobs you do not need an ID. (See “Displaying Spool Status Information” on page OP-275 to determine the jobID.)

1. Press the attention key and type **\$\$ REL**.

```
> $$ REL
JOBID: _
```

2. Are you releasing one job, or all jobs?

- Enter the 1 to 3 character spool job ID to release a single job.
- Enter **ALL** to release all spool jobs.

```
JOBID: 1
JOB(S) RELEASED
```

Controlling Program Output

Spooling Program Output (*continued*)

Keeping and Releasing Kept Jobs

The spool facility automatically deletes a spool job after printing, unless the keep parameter is set to Y using the `$$ KEEP` command or by the application program generating the job. Use the following procedure to keep a job after printing or displaying or to specify that a kept job should be deleted after printing or displaying.

Note: With the `KEEP` parameter equal to Y, a job is kept after output, with the number of copies set to one. When that job is released, one additional copy is printed or displayed before the spool facility deletes the job.

Before You Start:

Determine the spool job ID for the job you wish to keep. (See "Displaying Spool Status Information" on page OP-275 to determine the ID.)

1. Press the attention key and type `$$ KEEP`.

```
> $$ KEEP  
JOBID: _
```

2. Enter the ID of the job you are keeping or releasing.

```
JOBID: 2  
Y/N: _
```

3. Are you keeping the job, or releasing a previously kept job?

```
Y/N: Y  
JOB KEEP SET
```

- Enter a **Y** to keep a job.
- Enter an **N** to release a previously kept job.

Spooling Program Output (*continued*)

Deleting Spool Jobs

Use the following procedure to delete ready or printing spool jobs. You can delete one job, all jobs beginning with a prefix, or all jobs. If a writer is printing or displaying a job you are deleting, spool stops the output operation and deletes the job.

Before You Start:

Obtain one of the following:

- The spool job ID if you are deleting one job.
- The prefix you want to use if you are deleting all jobs whose names begin with a prefix.

1. Depending on the type of delete you are doing, press the attention key and:

- Enter **\$\$ DALL** if you are deleting all spool jobs.
- Enter **\$\$ DE** if you are deleting one job. Continue with step 2(a).
- Enter **\$\$ DG** if you are deleting all jobs whose names begin with a prefix. Continue with step 2(b).

```
> $$ DALL  
DALL COMMAND COMPLETE
```

2(a). Enter the ID of the job being deleted.

```
> $$ DE  
JOBID: 2  
JOB DELETED
```

2(b). Enter the prefix for the jobs being deleted.

```
> $$ DG  
GENERIC JOB NAME PREFIX: MYJOB  
DG COMMAND COMPLETED
```

Controlling Program Output

Spooling Program Output (*continued*)

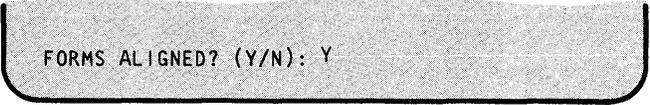
Responding to Spool Forms Alignment Prompt

Use the following procedure to respond to a spool forms alignment prompt. The prompt is displayed at the display terminal associated with the printer when a job specifying forms alignment is started.

Are the forms properly aligned in the printer?

- If so, enter a **Y** (The writer ejects the printer to the start of the next page and continues printing the job.)
- If not, enter an **N** (The writer ejects the printer to a new page so you can align the form.)

Enter a **Y** after aligning the form.



FORMS ALIGNED? (Y/N): Y

Chapter 8. Responding to Problems

You may experience 3 basic types of problems while operating a Series/1:

- Operator errors such as incorrect command syntax.
- Problems with the software (programs) you are running.
- Problems with the hardware.

Operator errors, such as incorrect command syntax or data, are easily analyzed and corrected. You will usually receive an error message indicating the error. An explanation of these messages is in the *Messages and Codes*.

This chapter helps you gather information about hardware and software problems and contains procedures for taking a stand-alone dump, printing a stand-alone or \$TRAP dump, dumping tapes to a printer, and determining the version and modification level of your system under the following headings:

- “Recording Problem Symptoms” on page OP-288
- “Restarting the System” on page OP-311
- “Taking a Stand-Alone Dump of Storage” on page OP-290
- “Printing a Stand-Alone or Formatted \$TRAP Dump” on page OP-293
- “Dumping Tapes to a Printer or Display Terminal” on page OP-302
- “Determining the Software Version Level” on page OP-306

Responding to Problems

Recording Problem Symptoms

Use the following form as a guide to record information about a problem with the operation of your system. The more information you can gather about the problem, the easier it will be to solve. A limited number of copies of the form are in Appendix B, "Blank Forms" on page OP-397.

Date 6/5/3 Time 10:00 the problem occurred.
Program name PAYROLL has been running for long time.

Record the console switch positions and mark any indicators that are on:

Load <input type="checkbox"/>	Wait <input checked="" type="checkbox"/>	Run <input checked="" type="checkbox"/>	IPL Source	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
			Primary <input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
			Alternate <input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
			Auto IPL <input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
			Normal <input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
			Diagnostic <input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Check <input type="checkbox"/>	Reset <input type="checkbox"/>	Store <input type="checkbox"/>	Data Buffer <input type="checkbox"/>	Console Interrupt <input type="checkbox"/>	Start <input type="checkbox"/>	Stop <input type="checkbox"/>	Level 0 <input type="checkbox"/>	Level 1 <input type="checkbox"/>	Level 2 <input type="checkbox"/>	Level 3 <input type="checkbox"/>	Stop On Address <input type="checkbox"/>	Instruct Step <input type="checkbox"/>	Check Restart <input type="checkbox"/>	Stop On Error <input type="checkbox"/>						

List unexpected results that occurred:

the same paychecks are being printed repeatedly.

List expected results before the problem occurred:

Paychecks were printing ok

Was any display, or printed output being generated just before the problem? Yes No

If display or printed output was being generated, is it correct? Yes No If no, specify what is wrong:

same output is being generated over and over.

Attach any output from the program such as printer output, magnetic tape or diskette output. attached: Yes No

List, or attach a copy of, any system messages.

List the I/O devices that were in operation:

\$SYSRTR, DISK, 4978 Display Terminal

Note any unusual conditions of I/O devices, such as whether any indicators are on that should be off or off that should be on.

Take a stand-alone dump.

Dump is on diskette number _____

Dump diskette is attached: Yes No

If not attached, it is _____

Figure 16 (Part 1 of 2). Problem Record Form

Recording Problem Symptoms (*continued*)

If the system is in a loop and you have a programmer console:



1. Press the Stop key



2. Press the Instruct Step key to place the processor in instruction-step mode.



3. Press the Start key and record the indicators that are on. Each time the Start key is pressed, one instruction is executed and the indicators show a new address in the instruction loop.

4. Repeat pressing the Start key and recording the indicators until the first pattern of indicators reappears. This may be as few as 2 patterns, or it may be many. This information will be of help in analyzing the loop.

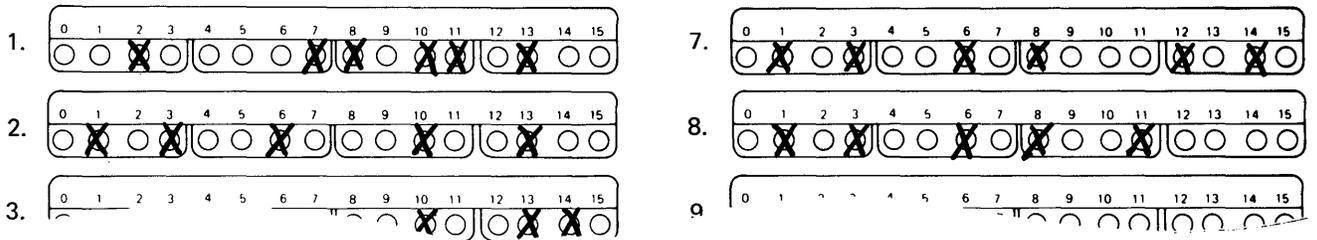


Figure 16 (Part 2 of 2). Problem Record Form

Responding to Problems

Taking a Stand-Alone Dump of Storage

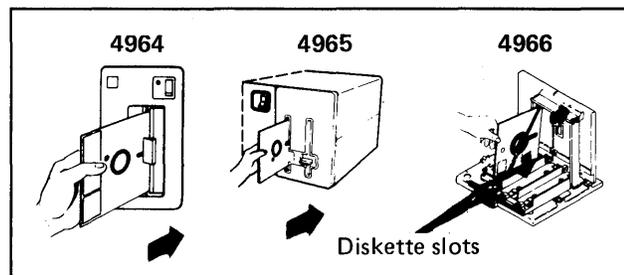
Use the following procedure to take a stand-alone dump. A stand-alone dump records a copy of processor storage. On a Series/1 with more than 512K of processor storage, you can also dump the unmapped portion of storage over 512K.

Before You Start:

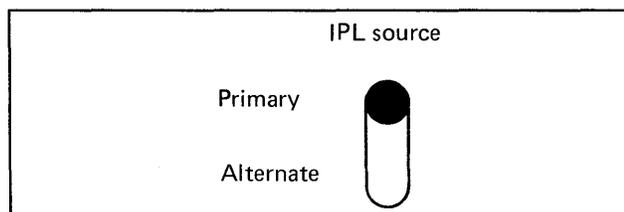
Observe the following requirements:

- To take a stand-alone storage dump of a processor that has a storage size of 512K or less, you need either the starter set diskette labeled XS4005 or a stand-alone dump diskette prepared as described under "Creating a Stand-Alone or \$TRAP Dump Diskette" on page OP-207.
- To take a stand-alone dump of a processor that has more than 512K of storage, you need two diskettes prepared as described under "Creating a Stand-Alone or \$TRAP Dump Diskette" on page OP-207. (These diskettes must have been created on a Version 4.1 level system and must have volume labels of IBMEDX.)
- If your Series/1 has a programmer's console, the **Mode** switch should **not** be set to Diagnostic mode.

1. Insert the diskette into any diskette slot that can be used for an IPL. **Do not** close the diskette drive door at this time.



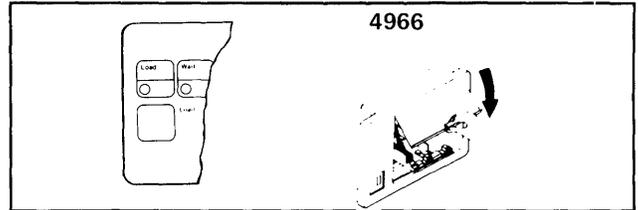
2. Set the **IPL Source** switch to IPL from the diskette. (This setting is unique to your system hardware.)



Taking a Stand-Alone Dump of Storage *(continued)*

- 3.** If you are using a 4964 or 4965 diskette drive, press the **Load** button.

If you are using a 4966 Diskette Magazine Unit, close the drive door.

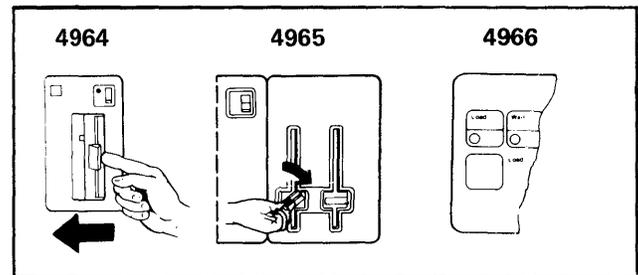


- 4.** If you are using a 4964 or 4965, close the diskette drive door.

If you are using a 4966, press the **Load** button.

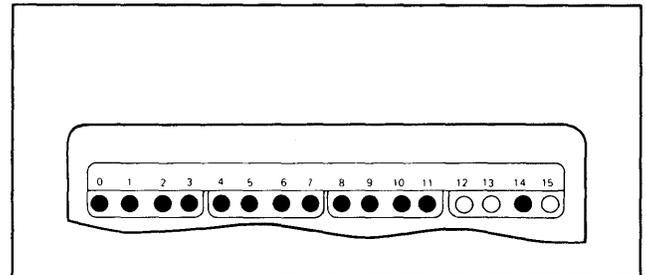
A program on the diskette starts making a copy of processor storage on the diskette.

If you are taking a storage dump of a processor with storage of 512K or less, continue with step 6.



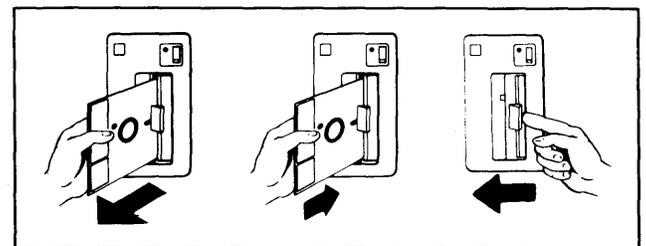
- 5(a).** If you are taking a storage dump of a processor with more than 512K of storage, the hardware does a seek-recalibrate when the first diskette finishes. It also displays X'FFF2' in the console lights.

To dump the rest of storage, continue with step 5(b). If you do not want to dump the additional storage, skip to step 7.



- 5(b).** Remove the diskette from the diskette drive, insert the next diskette in its place, and close the diskette drive door. Continue with step 6 unless the system displays X'1111' in the console lights.

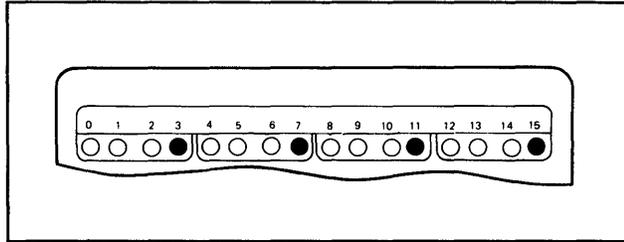
(Number the diskettes so you can print them in the correct order.)



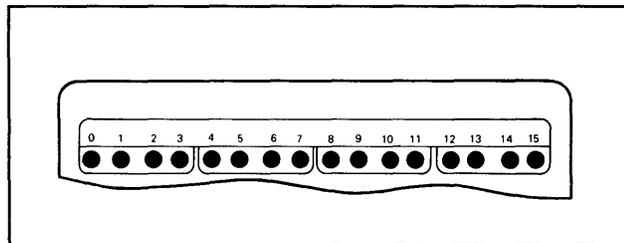
Responding to Problems

Taking a Stand-Alone Dump of Storage *(continued)*

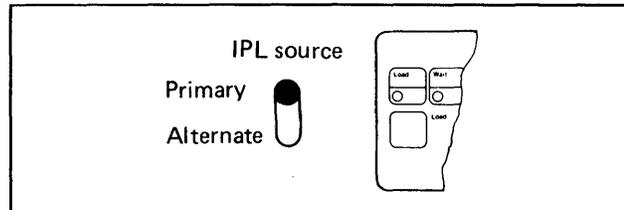
- 5(c).** If the system displays X'1111' in the console lights after you close the diskette drive door, you have inserted an improperly formatted diskette. Replace the incorrect diskette with one that is formatted for a stand-alone dump. (See "Creating a Stand-Alone or \$TRAP Dump Diskette" on page OP-207)



- 6.** When the system finishes, the hardware does a seek-recalibrate twice, causing the audible sound of the heads retracting to occur two times. If the system displays X'FFFF' in the console lights continue with step 7. If the system displays X'FFF2' return to step 5(b).



- 7.** IPL the system in the normal manner to resume system operation, or print the contents of the stand-alone dump diskette as described under "Printing a Stand-Alone or Formatted \$TRAP Dump" on page OP-293.



Printing a Stand-Alone or Formatted \$TRAP Dump

This procedure prints the contents of stand-alone or \$TRAP dump data sets.

Note: This procedure applies to printing from diskettes only. If you are printing from disk or copying the diskettes to disk before printing them, refer to the *Operator Commands and Utilities Reference*.

Before You Start:

Observe the following requirements:

- If the dump is on a stand-alone diskette, the diskette must be in a diskette drive. For a dump of a processor with more than 512K of storage, insert diskette number 1 into the drive.
- Obtain the name of the data set and volume containing the dump.

If you are printing a stand-alone dump made with the starter system diskette XS4005, the data set and volume names are \$\$EDXLIB, XS4005. If the stand-alone or \$TRAP dump was made with a diskette created using the \$DASDI utility, the name is \$\$EDXLIB, IBMEDX.

If you do not know the data set and volume name, see “Listing Volume and Data Set Directory Information” on page OP-181 to list all the names on your system.

- Obtain the name of the printer you are using to print the dump. (See “Listing System Terminal Assignments” on page OP-44 if you do not know this name.)

1. Load the utility \$DUMP:

- Using the \$L operator command, press the attention key and enter **\$L \$DUMP**.
- Using the session manager, select option 9.1.

```
> $L $DUMP
DUMPDS (NAME,VOLUME): _
```

2. Enter the data set and volume name of the dump data set.

```
DUMPDS (NAME,VOLUME) $$EDXLIB,IBMEDX
LOADING $DUMP          41P,08:30:45, LP= 00...
ENTER DEVICE NAME FOR OUTPUT: _
```

Responding to Problems

Printing a Stand-Alone or Formatted \$TRAP Dump (*continued*)

3. Enter the name of the device you are using to print the dump. If you press the enter key without entering a name, the listing will be displayed on your display terminal.

```
ENTER DEVICE NAME FOR OUTPUT: $$SYSPRTR
PARTIAL DISPLAY? (Y/N): _
```

4. Do you want to print a specified partition or the entire formatted storage?

- A specified partition, enter a Y.
- All of formatted storage, enter an N and continue with step 9.

Note: If you have less than 512K of storage, skip step 5 and continue with step 6.

```
PARTIAL DISPLAY? (Y/N):
USING STAND-ALONE DUMP OUTPUT
```

5. Reply N to copy diskettes to a disk data set. If you wish to copy diskettes to a disk data set, refer to the *Operator Commands and Utilities Reference* for more information.

```
USING STAND-ALONE DUMP OUTPUT
MAPPED STORAGE SPANS DISKETTE WHICH WILL CAUSE REPEATED
DISKETTE SWITCHING DURING CONTROL BLOCK FORMATTING
IF DUMP DATA SET IS COPIED TO DISK DATA SET, NO
DISKETTE SWITCHING
DISK DATA SET MUST BE      8200 RECORDS IN LENGTH
COPY DISKETTE(S) TO DISK DATA SET (Y/N)? N
```

Printing a Stand-Alone or Formatted \$TRAP Dump (*continued*)

- 6.** Do you want to include the formatted control blocks in the printout?
- If so, enter a **Y**, and \$DUMP prints the control blocks.
 - If not, enter an **N** and continue with step 7.

```
FORMAT CONTROL BLOCKS (Y/N)? Y
DUMP MAPPED STORAGE? (Y/N): _
```

- 7.** Do you want to dump mapped storage?
- If so, enter a **Y** and continue with step 8.
 - If not, enter an **N** and continue with step 10.

```
DUMP MAPPED STORAGE? (Y/N): Y
VALID PARTITIONS ARE:
```

- 8.** The system tells you which partitions are available for printing. Select the partition you want to print.

```
VALID PARTITIONS ARE:
  1, 2, 3, 4, 5, 6
ENTER PARTITION # : 3
ENTER START AND ENDING ADDRESS IN
HEX (2 VALUES): _
```

- 8(a).** Enter the hexadecimal starting and ending address of the storage area you are printing.

```
ENTER START AND ENDING ADDRESS IN
HEX (2 VALUES): 400 500
ANOTHER MAPPED AREA? (Y/N): _
```

Responding to Problems

Printing a Stand-Alone or Formatted \$TRAP Dump (*continued*)

9. Do you want to print another partition.
- If so, reply **Y** to this prompt and continue with step 8.
 - If not, reply **N** to this prompt and continue with the next step.

```
ANOTHER MAPPED AREA? (Y/N): _
```

10. Do you want to dump unmapped storage?
- If so, enter a **Y** and \$DUMP prints all of unmapped storage. If you are printing from a **disk** data set, skip to step 12.
 - If not, enter an **N** and continue with step 12.

```
DUMP UNMAPPED STORAGE? (Y/N): Y
```

```
$DUMP REQUIRES DISKETTE # 2. PLEASE MOUNT DISKETTE.  
PRESS "ENTER" TO CONTINUE OR "PF3" TO END DUMP.
```

11. Do you want to dump the contents of the second diskette?
- If so, remove the first diskette from the diskette drive, insert the second diskette in its place, close the diskette drive door, and press the enter key.
 - If not, press the PF3 key.

```
$DUMP REQUIRES DISKETTE # 2. PLEASE MOUNT DISKETTE.  
PRESS "ENTER" TO CONTINUE OR "PF3" TO END DUMP.
```

```
ANOTHER AREA? (Y/N): _
```

12. Do you want to print another area of storage?
- If so, enter a **Y**, and continue with step 7.
 - If not, enter an **N** and \$DUMP ends.

```
ANOTHER AREA (Y/N)? N
```

```
$DUMP ENDED AT 08:40:00
```

Listing Your Log Data Set on the Console

This procedure lists program and system checks and other errors, recorded in your log data set, on the console.

Before You Start:

Observe the following requirements:

- You must load \$LOG when the error occurs. \$LOG should not be active when you print the data set.
- You must know the name of the log data set you have allocated.

1. Load the utility \$DISKUT2:

- Using the \$L operator command, press the attention key and enter \$L \$DISKUT2.
- Using the session manager, select option 3.2.

```
> $L $DISKUT2
LOADING $DISKUT2 51P,00:30:15, LP= 9200, PART=1
COMMAND (?):
```

2. Enter the list command (LL) and the name of the log data set.

```
COMMAND(?): LL $LOGDS
ENTER DEVICE ADDRESS, NULL FOR ALL ENTRIES,
OR 'FFFF' FOR PROGRAM/SYSTEM CHECKS ONLY : _
```

Responding to Problems

Listing Your Log Data Set on the Console (*continued*)

3. Enter one of the following:

- To list a specific device, enter the device address.
- To list all entries (which is the default), press the enter key.
- To list program and system checks only, enter "FFFF."

To read the format of a log entry, refer to the *Problem Determination Guide*.

```
ENTER DEVICE ADDRESS, NULL FOR ALL ENTRIES,  
OR 'FFFF' FOR PROGRAM/SYSTEM CHECKS ONLY : FFFF  
  
      **PROGRAM CHECK**  
DATE:10/27/83      TIME: 08:25:31  
SAR = 904B  PSW = 8002 PSW ANALYSIS: SPECIFICATION CHECK  
.  
.  
COMMAND (?):
```

4. Do you want to list the log data set again?

- If so, continue with step 2 on page OP-297.
- If not, enter EN to end \$DISKUT2.

```
COMMAND(?): EN  
  
$DISKUT2 ENDED AT 01:02:00  
COMMAND (?):
```

Copying a \$TRAP Data Set to Two Diskettes

This procedure allows you to copy a \$TRAP disk data set which is greater than 512K, to 2 diskettes. Use this procedure:

- If you wish to print a copy of the dump on a different Series/1.
- If you want to send a copy of the dump and an APAR to IBM.

Instructions for filling out an APAR are explained in the *IBM Series/1 Software Service Guide*, GC34-0099.

Before You Start:

Obtain the following:

- Two \$TRAP diskettes. See “Creating a Stand-Alone or \$TRAP Dump Diskette” on page OP-207 for the procedure used to create dump diskettes.
- Determine the size of the \$TRAP disk data set. To do this, refer to the procedure “List One or More Data Sets on a Volume” on page OP-187.

Note: If you have more than one megabyte of storage, you will need more than 2 diskettes for this procedure.

1. Load the utility \$COPY:
 - Using the \$L operator command, press the attention key and enter \$L \$COPY.
 - Using the session manager, select option 3.5.

```
> $L $COPY
LOADING $COPY          45P, LP= 1300, PART= 2
```

2. Enter the CD (copy data set) command.

```
COMMAND (?): CD
SOURCE (NAME,VOLUME):
```

3. Enter the \$TRAP disk data set name and volume.

```
SOURCE (NAME,VOLUME): TRAP,TRPDMP
COPY ENTIRE DATA SET (Y/N)?
```

4. Reply N to this prompt if you do not wish to copy the entire data set.

```
COPY ENTIRE DATA SET (Y/N)? N
FIRST RECORD: -
```

Responding to Problems

Copying a \$TRAP Data Set to Two Diskettes (*continued*)

5. Enter 1 for the first record.

```
FIRST RECORD: 1
LAST RECORD: _
```

6. Enter 2,048 for the last record.

```
LAST RECORD: 2,048
TARGET (NAME,VOLUME) : _,_
```

7. Enter \$\$EDXLIB,IBMEDX for the target name and volume of the stand-alone dump diskette you are using.

```
TARGET (NAME,VOLUME): $$EDXLIB,IBMEDX
FIRST RECORD: _
```

8. Enter 91 for the first record.

```
FIRST RECORD: 91
ARE ALL PARAMETERS CORRECT (Y/N)? _
```

Copying a \$TRAP Data Set to Two Diskettes (*continued*)

9. Reply Y to this prompt after checking to be sure that you have entered all the previous responses in this procedure correctly.

```
ARE ALL PARAMETERS CORRECT (Y/N)? Y
COPY COMPLETE
      2,048 RECORDS COPIED
COMMAND (?):
```

10. The first 8 steps of this procedure copy the first 512K bytes of the \$TRAP data set. To copy the rest of the data set, replace the first diskette with the second diskette. (Use the same diskette drive for both diskettes.)

Repeat steps 2 through 9. Respond with the same replies with the exception of steps 5 and 6. This time the first record is **2,049**. The number you enter for the last record will change each time you do this procedure. Enter the last record of the \$TRAP disk data set you are currently using.

11. Once you have copied the entire disk data set to diskettes, use the EN command to end \$COPY.

```
COPY COMPLETE
      2,048 RECORDS COPIED
COMMAND (?): EN
$COPY ENDED AT 08:40:00
```

Responding to Problems

Dumping Tapes to a Printer or Display Terminal

This procedure prints the contents of all or part of a tape on a printer or displays it on a display terminal. The printout is in hexadecimal form plus the EBCDIC conversion.

Before You Start:

Obtain the following information:

- The hexadecimal address of the tape drive. (\$TAPEUT1 lists this address when you load it, or you can use the procedure "Listing the Devices on Your System" on page OP-318 to get it.)
- The maximum block size of records on the tape.
- The volume label (TAPEID) of the tape. (See the \$TAPEUT1 utility in the *Operator Commands and Utilities Reference* to determine this information.)
- The number of records you are printing.
- If you are not printing the dump on the printer named \$SYSPRTR or displaying it on the display terminal you are currently using, you need the name of the printer or display terminal. (See "Listing System Terminal Assignments" on page OP-44 if you do not know the name of the printer or display terminal.)

1. Is the tape positioned at the start of the data set you are dumping?

- If it is, skip to step 2.
- If it is not, enter \$VARYON *ioda* *file* to vary the tape online to the file number of the data set where:

ioda is the hexadecimal device address of the tape drive.

file is the decimal file number being accessed. In the example, we are accessing file number 2.

```
> $VARYON 4C 2
MYVOL ONLINE
```

2. Load the utility \$TAPEUT1:

- Using the \$L operator command, press the attention key and enter \$L \$TAPEUT1.
- Using the session manager, select option 3.10.

```
> $L $TAPEUT1
LOADING $TAPEUT1 21P,08:30:45, LP=00...
TAPEUT1 DUAL SL 1600 OFFLINE
.
.
.
COMMAND (?) _
```

Dumping Tapes to a Printer or Display Terminal (*continued*)

3. Where do you want to start the dump?

- To start at the current location, skip to step 8 on page OP-304.
- To move the tape to another record, enter **MT**.

```
COMMAND (?) MT
ENTER TAPEID (1-6 CHARS): _
```

4. Enter the 1-6 character ID (TAPEID) of the tape device. If you do not receive the "DEVICE NOT OFFLINE" message, skip to 6.

```
ENTER TAPEID (1-6 CHARS): TAPE01
DEVICE NOT OFFLINE!
DO YOU WISH TO CONTINUE? (Y/N): _
```

5. Do you want \$TAPEUT1 to vary the tape drive offline and continue? (This message is displayed only if the tape drive is online.)

- If so, enter a **Y**.
- If not, enter an **N**. Skip to step 19 on page OP-306 .

```
DEVICE NOT OFFLINE!
DO YOU WISH TO CONTINUE? (Y/N): Y
TYPE? FSR/BSR/FSF/BSF/WTM/REW/ROFF/OFF: _
```

6. Enter the command corresponding to the direction and distance you are moving the tape. (With the MT command, a file is one tapemark and there are at least two tapemarks per data set.)

- BSF n** Backspace *n* files.
BSR n Backspace *n* records.
FSF n Forward space *n* files.
FSR n Forward space *n* records.
REW Rewind to the start of the tape.

```
TYPE?FSR/BSR/FSF/BSF/WTM/REW/ROFF/OFF: BSF 4
BSF SUCCESSFUL
TYPE? FSR/BSR/FSF/BSF/WTM/REW/ROFF/OFF: _
```

7. Enter **END** to end the move tape function.

```
TYPE? FSR/BSR/FSF/BSF/WTM/REW/ROFF/OFF: END
COMMAND (?): _
```

Responding to Problems

Dumping Tapes to a Printer or Display Terminal (*continued*)

8. Enter **DP** to dump tape records to a printer or display terminal.

```
COMMAND (?) DP
ENTER MAXIMUM BLOCKSIZE: _
```

9. Enter the maximum blocksize on the tape. If \$TAPEUT1 reads a record larger or smaller than the specified size, it issues a message indicating that a wrong record length was encountered, gives the actual record size, and continues. \$TAPEUT1 pads smaller records with zeros before printing them. It truncates larger records to the maximum record size.

```
ENTER MAXIMUM BLOCKSIZE: 256
USE ATTN/CA TO CANCEL DUMP
ENTER TAPEID (1-6 CHARS): _
```

10. Enter the 1-6 character ID (TAPEID) of the tape device.

If you decide to cancel the dump while it is printing, press the attention key and enter **CA**.

```
USE ATTN/CA TO CANCEL DUMP
ENTER TAPEID (1-6 CHARS): TAPE01
ENTER NUMBER OF RECORDS TO DUMP: _
```

11. Enter the number of records you are printing.

```
ENTER NUMBER OF RECORDS TO DUMP: 5
PRINT DUMP ON $$SYSPRTR? (Y/N): _
```

12. Are you printing the dump on the printer named \$\$SYSPRTR?

- If so, enter a **Y** and continue with step 15 on page OP-305 .
- If not, enter an **N**.

```
PRINT DUMP ON $$SYSPRTR? (Y/N): N
ENTER TARGET TERMINAL NAME (*,NAME): _
```

Dumping Tapes to a Printer or Display Terminal (*continued*)

13. Do you want the dump displayed on the display terminal you are currently using?

- If so, enter an *****.
- If not, enter the name of the printer or display terminal where you want the dump printed.

```
ENTER TARGET TERMINAL NAME (*,NAME): *  
TAPE MARK ENCOUNTERED - CONTINUE? (Y/N):
```

14. Do you want to dump the next section of the tape?

- If so, enter a **Y**. Repeat this step until you have dumped the desired portion of the tape.
- If not, enter an **N**.

```
TAPE MARK ENCOUNTERED - CONTINUE? N  
DUMP ENDED  
COMMAND (?): _
```

15. Do you want to rewind the tape?

- If so, enter **MT**. Continue with step 16 .
- If not, enter **EN** to end the utility. Skip to step 20 on page OP-306 .

```
COMMAND (?): MT  
ENTER TAPEID (1-6 CHARS): _
```

16. Enter the 1 to 6 character ID (TAPEID) of the tape drive.

```
ENTER TAPEID (1-6 CHARS): TAPE01  
TYPE? FSR/BSR/FSF/BSF/WTM/REW/ROFF/OFF: _
```

17. Enter **REW** to rewind the tape.

```
TYPE? FSR/BSR/FSF/BSF/WTM/REW/ROFF/OFF: REW  
REW SUCCESSFUL  
TYPE? FSR/BSR/FSF/BSF/WTM/REW/ROFF/OFF: _
```

18. Enter **END** to end the move tape function.

```
TYPE? FSR/BSR/FSF/BSF/WTM/REW/ROFF/OFF: END  
COMMAND (?) _
```

Responding to Problems

Dumping Tapes to a Printer or Display Terminal *(continued)*

19. Enter EN to end \$TAPEUT1.

```
COMMAND (?) EN
$TAPEUT1 ENDED AT 08:55:00
```

20. To vary the tape back online, press the attention key and enter \$VARYON *ioda*, where *ioda* is the hexadecimal address of the tape device.

```
> $VARYON 4E
TAPE01 ONLINE
```

Determining the Software Version Level

You may need to know the version and modification level of your EDX system when contacting IBM about a problem or when installing an update. You can find out what it is for the system in general using the \$D dump storage operator command, or from a printout of a dump of partition 1. You can find out the version and modification level of each individual module by displaying the last word of the module. This section contains procedures for each of these methods.

Determining Version Level with \$D

Use this procedure to display the version and modification level for your EDX system or the contents of other storage locations on your display terminal.

Before You Start:

Obtain the following information:

- The hexadecimal origin address of the program whose storage area you are displaying. This address is X'0000' for the version and modification level. To determine the origin address of application programs, see "Listing Partitions and Active Programs" on page OP-227.
- The hexadecimal address in the program where the display is to start. This address is X'0298' for the version and modification level.
- The decimal number of words to be displayed. This number is 1 for the version and modification level. A maximum of 16 words can be displayed.

1. Press the attention key and enter the operator command \$D.

```
> $D
ENTER ORIGIN: _
```

Determining the Software Version Level *(continued)*

2. Enter the origin address. Enter **0000** to display the version and modification level.

```
ENTER ORIGIN: 0000
ENTER ADDRESS,COUNT: _
```

3. Enter the starting address and the number of words you want displayed. Enter **0298,1** to display the version and modification level.

```
ENTER ADDRESS,COUNT: 0298,1
0298: 4000
ANOTHER DISPLAY? _
```

Storage location X'0298' is interpreted as:

abcd where: a = The version level
b = The modification level
c = A unique identifier assigned to the program.
Location X'0298' is always 0.
d = The number of PTFs (Program Trouble Fixes)

4. Do you want to display the contents of another storage location starting at the same origin address?

```
ANOTHER DISPLAY? N
```

- If so, enter a **Y** and continue with step 3 .
- If not, enter an **N** to end the \$D function.

Responding to Problems

Determining the Software Version Level (*continued*)

Determining the Version Level from a Dump of Partition 1

A printout of a stand-alone or \$TRAP dump diskette contains the version and modification level at the beginning of the storage map. The version modification level is labeled EDXFLAGS on the printout as shown in Figure 17. It is interpreted in the same way as the \$D display.

```
STORAGE MAP:      $SYSCOM AT ADDRESS 1580
EDXFLAGS 4000     SVCFLAGS  8000
.
.
.
```

Figure 17. Storage Dump, EDXFLAGS

Determining the Version Level of Programs

Use this procedure to determine the version and modification level of programs. The version and modification level of an individual program module is contained in the last word of that module. This procedure shows you how to calculate the address of the last word and display its contents. To do this procedure you need a copy of the system generation listing for your system.

1. Turn to the last portion of the system generation listing. It is a map of the completed supervisor.

```
$EDXNUCT,EDX002 STORED
RESIDENT SEGMENT:
          LABEL      ADDR LNTH LABEL...
A SECTION = EDXSYS   0000 052E
  ENTRY = $START    0230      RETURN...
.
.
.
          ENTRY = CMDTABLE 0306
B SECTION = $EDXDEF C 052E 133A
.
.
.
```

Determining the Software Version Level *(continued)*

2. Calculate the address of the version level. For this example, we want the version level for the module named EDXSYS.
 - a. Find the SECTION label that shows the start of the program whose version level you are listing. In the example, listing in step 1 on page OP-308 EDXSYS is at **A**.
 - b. Find the start of the next SECTION and note its starting address. In the example in step 1 on page OP-308, the next section is \$EDXDEF **B** which starts at X'052E' **C**.
 - c. Subtract a X'0002' from the starting address to get the address of the version and modification level.

A	Program name	(EDXSYS) =	_____
B	Next program name	(\$EDXDEF) =	_____
C	Starting address	(X'052E') =	_____
	Less X'0002'	(X'0002')	X'0002'

	The address is:	(X'052C') =	_____

3. Load the utility \$DISKUT2:
 - Using the \$L operator command, press the attention key and enter \$L \$DISKUT2.
 - Using the session manager, select option 3.2.
4. Are you displaying the information on your display terminal or on a printer?
 - Enter **DU** to display the information on your terminal.
 - Enter **DP** to print the information on a printer.

```
> $L $DISKUT2
LOADING $DISKUT2 51P,08:45:30, LP= 920...
$DISKUT2 - DATA SET MGMT. UTILITY II
USING VOLUME EDX002
COMMAND (?): _
```

```
COMMAND (?): DU
PGM OR DS NAME: _
```

5. Enter the name of the supervisor.

```
PGM OR DS NAME: $EDXNUC
$EDXNUC IS A PROGRAM OF HEX SIZE 0000B834
ADDRESS: _
```

Responding to Problems

Determining the Software Version Level (*continued*)

6. Enter the address calculated in step 2 on page OP-309.

```
ADDRESS: 52C  
DUMP TO PROGRAM END: _
```

7. Enter an N.

```
DUMP TO PROGRAM END: N  
HOW MANY WORDS: _
```

8. Enter a 1 to display the word containing the version and modification level for this module.

```
HOW MANY WORDS: 1  
052C 4000  
DUMP COMPLETE  
ANOTHER AREA? _
```

9. Do you want to display the version level for another module in the supervisor?

- If so, enter a Y and continue with step 6.
- If not, enter an N.

```
ANOTHER AREA? N  
COMMAND (?): _
```

10. Enter EN to end \$DISKUT2.

```
COMMAND (?): EN  
$DISKUT2 ENDED AT 08:55:00
```

Restarting the System

You should always correct any errors before continuing operation. However, if this is not possible, you can continue operation with the following considerations:

- If the problem is traced to a single program, you can start your system running without that program.
- If the problem is traced to an individual device, such as a printer, restart operations without that device. For a printer, direct the output to another printer, if available.
- If the system is not operational because of problems such as a hard wait, machine check, uncontrolled loop, or hardware error, you must restart the system by doing an IPL as described under “Loading the System (Initial Program Load)” on page OP-22.
- If the problem affected the integrity of programs or data on the disk or diskette storage devices, you must restore the system from your backup copy as shown under “Restoring the System” on page OP-334.

Chapter 9. Keeping Records and Backing-up the System

The procedures you follow in the daily operation of your EDX system are established by the needs of your business and the application programs run on your system. These procedures vary from one business to another.

There are, however, procedures that are the same regardless of the needs of your business and the application programs used. These procedures involve keeping records of system activity and backup copies of the programs and data. This chapter covers these topics under the following headings:

- “Keeping Records” on page OP-314
- “Saving the EDX System” on page OP-319
- “Restoring the System” on page OP-334.

Keeping Records and Backing-up the System

Keeping Records

The efficiency of daily operations is increased by up-to-date records.

Keeping records helps by:

- Providing information for running the programs.
- Telling you how long it takes to run each program.
- Keeping track of which jobs have been run and which jobs still need to be run.
- Keeping track of data, diskettes, tapes, and backup copies.

The following suggested forms are supplied to help you design others that meet the specific needs of your business. (Full-page copies of each are supplied in the Appendix B, "Blank Forms" on page OP-397.)

Operations Log

An operations log provides a summary of the daily operation of your EDX system. The following sample log has space for the date, the name of the job, program, or procedure, the time started, the time ended, the operator's initials, and remarks about problems encountered while running the job.

Date	Job procedure description	Time started	Time ended	Opr.	Remarks
5/21	Billing	8:45	10:00	WJB	No problems
5/22	Payroll	9:00	11:00	BJB	Error in input file
5/22	Address labels	1:00	2:30	Sue	No problems

Figure 18. Operations Log

Keeping Records (continued)

Job Instructions

Most programs and \$JOBUTIL procedures have specific requirements, such as data sets that must be allocated, and input that must be prepared, that must be met before, or while, the job is run. System operation is more efficient if the operator has a set of instructions outlining these requirements. The following sample instruction form has places for recording some of the more common requirements.

Job name: <u>Billing</u>			
Program/procedure name: <u>BILL1</u>		On volume: <u>MYVOL</u>	
Stored on: Disk <input checked="" type="checkbox"/> Diskette # _____			
Other data sets required:			
Data set name	Volume name	Diskette #	Tape #
<u>ADDR1</u>	<u>VOL1</u>	<u>D007</u>	_____
<u>ADDR2</u>	<u>VOL1</u>	<u>D007</u>	_____
Input data required: (Attached <input checked="" type="radio"/> Obtain from)			
<u>Update ADDR2 with attached addresses</u>			
Jobs that must be run first:			
<u>Payments received</u>			
Prompts that can be expected:			
<u>None</u>			
Special printer forms:			
<u>Customer bills form ABØØ1</u>			
Notes/Special Instructions:			

Date run: <u>6/1/3</u> Time: <u>10:00</u> Operator: <u>Barry</u>			
Comments: <u>Job ran ok.</u>			

Figure 19. Job Instructions

Keeping Records and Backing-up the System

Keeping Records (*continued*)

Diskette and Tape Log

A log of the diskettes and tapes used with your system helps keep track of when they were initialized, their format, contents, and purpose. It also tells you how old the diskette is and when to verify if it is still usable. (You verify the quality of the recording surface by initializing a tape or diskette. Since initializing a tape or diskette deletes any data currently on it, be sure to make a backup copy first.)

When you initialize a diskette or tape, record the sequential serial number, the date, and a brief description of the contents on an external label affixed to the diskette or tape. Record this and the other information in the log. (The serial number is your means of relating the diskette or tape to its corresponding log entry.)

The following sample diskette log has spaces to record the serial number, the date created, a description of the contents, the serial number of the backup diskette or tape, the volume label, the format (EDX/exchange/bytes per sector) and number of bytes per sector, the names of volumes on the diskette, and the names of data sets on each volume.

The sample tape log has spaces to record the serial number, the date created, a description of the contents, the density (800/1600/DUAL), the label type (SL/NL/BLP), the name of the volume on the tape, the data sets on the volume, the expiration date and the file number of each data set.

Serial	Date	Description	Backed up on	Volume label	Fmt.	BPS	Volume name	Data set name
D001	6/15/3	Addresses	D005	ADDOL	EDX	256	DIV101	DEPT 301
								DEPT 302
							DIV102	DEPT 444

Figure 20. Diskette Log

Serial	Date	Description	Dens.	Type	Volume name	Data set		
						Name	Exp. date	File no.
T001	5/21/3	Accounts	1600	SL	MYVOL	CUSTØ1	84001	1
						CUSTØ2	83300	2
T002	6/1/3	Backup of disk	1600	NL	—	BACK1	83250	1

Figure 21. Tape Log

Keeping Records (*continued*)

Backup Log

Backup copies of the contents of your disk, and the diskettes and tapes used in daily operations, provide checkpoints for recovery should anything happen to the data stored on the originals.

A log of these backup copies helps keep track of them. The following sample log has places to record the serial number of the backup diskette or tape, the date created, a brief description, the serial number of the diskette or tape backed up (enter *disk* for a disk), the volume label if this is a diskette, and the volume and data set name(s) if this is a tape.

Note: You should also keep a copy of the worksheets used to generate your EDX system and the printed system generation listing, with the diskettes or tapes containing the backup copy of your EDX system.

Serial	Date	Description	Backup of	Diskette volume label	Tape	
					Volume	Data set
T003	6/10/3	Backup of 6/1 system	full system	disk	BKUP6	BACKUP
D005	6/15/3	Address file as of 6/15	D001	BK01		
D01				BK02		

Figure 22. Backup Log

Procedures Log

A log of the \$JOBUTIL procedures set up for your system helps keep track of the purpose of each procedure and the individual programs it runs.

The following sample log has places for the name of the job (use the name that is on the JOB command), the date it was created, the serial number of the diskette or *disk* if it is on a disk, the name of the data set it is on, the name of the volume containing the data set, and notes such as a description of what the job does or the names of the programs it runs.

Job name	Date	Dskt. serial number	Data set name	Volume name	Notes
Payroll link	6/1/3	D005	LINK1	LNKVOL	Payroll program prep. runs PAYROLL/\$EDXASM/\$EDXLINK
Payroll	6/2/3	D005	MYPROC	MYVOL	Runs payroll program

Figure 23. Procedures Log

Keeping Records and Backing-up the System

Keeping Records (*continued*)

Listing the Devices on Your System

Use this procedure to determine the names, addresses, and IDs of the devices that are either physically attached to your Series/1, or are supported by the EDX supervisor after initial program load (IPL). These two lists may not be the same. You can have devices attached to your Series/1 that are not supported by the generated system.

You should keep a copy of this list with the other backup information for your system. You may also want to keep copies with your daily operating procedures because some of the system utilities require the device address.

Before You Start:

Power on all Series/1 devices you wish to list.

1. Load the utility \$IOTEST:

- Using the \$L operator command, press the attention key and enter **\$L \$IOTEST**.
- Using the session manager, select option 9.3.

```
> $L $IOTEST
LOADING $IOTEST 36P,08:30:45 LP=5000...
ATTNLIST (ALTER) TO STOP LOOPING FUNCTIONS
COMMAND (?): _
```

2. Enter the command for the listing you want:

- Enter **LD** to list the devices attached to your system.
- Enter **LS** to list the devices supported by your supervisor.

If the listing fills the screen, see step 3 before pressing the enter key to view the rest of the listing.

```
COMMAND (?): LD
ACTUAL SERIES/1 HARDWARE CONFIGURATION
ADDRESS  ID  DEVICE TYPE
01      0206  4974 PRINTER
02      0106  4964 DISKETTE UNIT
03      00CA  4962 DISK MDL 3
04      0406  4979 DISPLAY STATION
21      0306  4973 PRINTER
:
:
COMMAND (?): _
```

3. Press the PF6 key to get a printed copy of the list.

6

Keeping Records (*continued*)

4. Enter EN to end \$IOTEST.

```
COMMAND (?): EN
$IOTEST ENDED AT 08:55:35
```

Saving the EDX System

Backup copies of your EDX system and other important volumes on the disk provide checkpoints for recovery should anything happen to the original copies. These copies can be kept on either diskettes or tape.

The first backup copy of your EDX system should be made immediately after the EDX system is generated. After that, copies should be made any time the system is regenerated or patched. Backup copies of application programs and data should be made on a regular basis or whenever they change significantly.

This section contains two procedures: one to copy a complete volume to diskette, and another to copy the complete disk, a volume, or a data set to tape.

Saving the System on Diskettes

Use this procedure to make an exact copy of a disk volume.

Saving a disk volume normally requires more than one diskette. The diskettes must be initialized with sequential volume labels (xxx000, xxx001,...) and must be inserted into the diskette drive, or 4966 diskette slot or magazines, in numerical order.

If you use 4966 diskette magazines, the diskettes must be in the correct sequence, with no empty slots in the magazine. You can use either, or both, of the diskette magazines.

Note: We suggest that a disk volume be compressed before it is saved. See "Compressing Volumes and Devices" on page OP-142 for the procedure.

Keeping Records and Backing-up the System

Saving the EDX System (*continued*)

Before You Start:

Determine the following information:

- The name of the volume being saved.
- The size of the volume being saved. (See “Determining Volume Size and Free Space” on page OP-205 to get this number.)
- The size of the volume directory on the volume being saved. (See “Determining Volume Size and Free Space” on page OP-205 to get this number.)
- The 3-character prefix you are using for the sequential diskette volume labels.
- 1-50 characters of identification information. This information is entered on the first diskette (xxx000). It should be specific enough, such as date and version level, to help verify the contents of the diskettes when they are used to restore the system.
- If you have to initialize the directories of the diskettes you are using, you will also need:
 - The hexadecimal device address of the diskette unit.
 - The owner identification.

1. Determine the number of diskettes required to hold the volume being saved. You can use either single- or double-sided diskettes initialized to the Event Driven Executive Format. However, all diskettes in the set must be the same type with the same number of bytes/sector (BPS). If you have enough diskettes, continue with step 2 on page OP-321.

Diskette type	Number of records at 128 BPS	Number of records at 256 BPS	Number of records at 1024 BPS
Diskette 1	946	1107	-
Diskette 2	1921	2217	-
Diskette 2-D	-	3845	4733

To calculate how many diskette are needed:

- a. Divide the number of records in the volume being saved by the number of records on each diskette.
- b. Increase the result of the division by 1 for the total number of diskettes required. (The first diskette is used for the directory and the data starts on the second, so you always require a minimum of two diskettes.)

$$\left(\frac{\text{volume size}}{\text{diskette size}} \right) + 1 = \text{number of diskettes}$$

If you do not have enough diskettes of the same type and record size, use the procedure “Initialize for EDX Format” on page OP-93 to initialize the required quantity.

Saving the EDX System (*continued*)

2. Initialize the diskette volume directories.

The full set of diskettes must have sequential volume labels of xxx000 through xxxnnn, where xxx is a three character prefix common to the set and nnn is the number of the last diskette in the set. For example, a ten diskette set might have volume labels of SAV000, SAV001,... SAV009.

Also, the first, or control diskette (volume label xxx000), can have a copy of the nucleus and IPL text so that it can be used to IPL the system.

If your set of diskettes is already set up with the correct volume IDs and, if required, the nucleus, continue with step 3 on page OP-322.

If not, use the \$INITDSK ID command to initialize the diskettes to the following specifications as shown in the example. (See "Initialize a Diskette Volume Directory" on page OP-106 for the step-by-step procedure.)

- For the entire set of diskettes, specify:
 - A single volume.
 - A volume label of xxxnnn. Where xxx is a 3-character prefix common to the set, and nnn is a sequential number starting with 000.
- For diskette xxx000:
 - Specify 4 data sets if diskette xxx000 will be used to IPL the system.
 - Specify 1 data set if diskette xxx000 will not be used to IPL the system.
 - Write verification is optional.
 - Allocate \$EDXNUC, if the diskette will be used to IPL the system.
 - Initialize IPL text if the diskette will be used to IPL the system.
- For diskettes xxx001-xxxnnn:
 - Specify 1 data set.
 - Write verification is optional.

```
> $L $INITDSK
COMMAND (?): ID

DEVICE ADDRESS: 02

INITIALIZE DEVICE MAY DESTROY ALL DATA
CONTINUE? Y

INITIALIZE DISKETTE AS MULTI-VOLUME TYPE? N

NEW VOLUME LABEL: SAV000

ENTER OWNER IDENTIFICATION: BACKUP
SINGLE VOLUME DISKETTE INITIALIZED
BACKUP ALLOCATED

INITIALIZE THE VOLUME? Y

MAXIMUM NUMBER OF DATASETS? 4

DO YOU WANT WRITE VERIFY FOR THIS VOLUME? N

ALLOCATE $EDXNUC? Y
VOLUME INITIALIZED

INITIALIZE IPL TEXT? Y
IPL TEXT WRITTEN
ALLOCATE ANOTHER VOLUME? _
```

Keeping Records and Backing-up the System

Saving the EDX System (*continued*)

3. Allocate a data set on each diskette using the \$DISKUT1 AL command as shown in the example. (See "Allocate a Data Set" on page OP-124 for the step-by-step procedure.)

- On the diskette with volume label xxx000, allocate one data set with the same name as the disk volume being saved and a size one record larger than the volume's directory.
- On the diskettes with volume labels xxx001-xxxxnnn, allocate two data sets.
 - The first is named \$CONTROL and is one record in size.
 - The second has the same name as the disk volume being saved with a size equal to the number of available records as shown in 1 on page OP-320.

```
> $L $DISKUT1
USING VOLUME EDX002
COMMAND(?): CV SAV000

COMMAND(?): AL

MEMBER NAME: EDX002

HOW MANY RECORDS? 64

DEFAULT TYPE = DATA - OK? Y
EDX002 CREATED

COMMAND(?): _
```

4. If you allocated the nucleus (\$EDXNUC) on diskette xxx000, use the \$COPYUT1 CM command to copy members \$MOVEVOL and \$LOADER from volume EDX002 as shown in the example. (See "Copy a Data Set with Allocation" on page OP-152 for the step-by-step procedure.)

If your system logging terminal (\$SYSLOG) is a 4978, copy \$4978IS0 and \$4978CS0 also.

After you have copied the members, end the \$COPYUT1 command.

```
> $L $COPYUT1
THE DEFINED SOURCE VOLUME IS EDX002, OK? Y

THE DEFINED TARGET VOLUME IS EDX002, OK? N

ENTER NEW TARGET VOLUME: SAV000

MEMBER WILL BE COPIED FROM EDX002 TO SAV000, OK? Y

COMMAND(?): CM

ENTER FROM (SOURCE) MEMBER: $LOADER

ENTER TO (TARGET) MEMBER OR * FOR
SAME NAME AS SOURCE: *
$LOADER COPY COMPLETE      nn RECORDS COPIED

COMMAND(?): EN
```

Saving the EDX System *(continued)*

5. Insert diskette xxx000 into the diskette drive, or all diskettes into 4966 magazines, and start the save process by loading the utility \$MOVEVOL:

- Using the \$L operator command: Press the attention key and enter \$L \$MOVEVOL
- Using the session manager: Select option 3.8. If you are using the session manger, steps 6 and 7 are combined in one parameter input menu.

```
> $L $MOVEVOL
DISK (NAME,VOLUME): _
```

6. Enter \$\$EDXLIB,volname, where volname is the name of the disk volume your are saving.

```
DISK (NAME,VOLUME): $$EDXLIB,EDX002
DISKETTE (NAME,VOLUME): _
```

7. Enter the name of the disk volume being saved and the volume label of the diskette (xxxnnn).

```
DISKETTE (NAME,VOLUME): EDX002,SAV000
LOADING $MOVEVOL 20P,08:35:45, LP=5200,...
DUMP VOLUME EDX002 FROM DISK TO DISKETTE: _
```

8. Did you enter the correct data set and volume names?

- If so, enter a Y.
- If not, enter an N and return to step 6.

```
DUMP VOLUME EDX002 FROM DISK TO DISKETTE: Y
PROCESSING DISKETTE VOLUME SAV000
ENTER VOLUME IDENTIFICATION (1-50 CHAR):
```

Keeping Records and Backing-up the System

Saving the EDX System (*continued*)

9. Enter 1 to 50 characters of identifying information about the saved volume, such as: the date, generation level, or contents. Be as specific as you can as this will help insure you use the most recent copy in the event you have more than one copy of the same volume.

```
ENTER VOLUME IDENTIFICATION (1-50 CHAR):  
5/20/82 EDX VERSION 5.0 CUSTOMIZED SUPERVISOR  
5 MORE DISKETTE(S) REQUIRED, CONTINUE? _
```

10. Do you have enough diskettes prepared?

- If so, enter a Y.
- If not, enter an N and \$MOVEVOL ends. You must restart at step 1 on page OP-320 and prepare enough diskettes.

```
5 MORE DISKETTE(S) REQUIRED, CONTINUE? Y  
PROCESSING DISKETTE VOLUME SAV000  
MOUNT NEXT DISKETTE  
REPLY -Y- WHEN DONE: _
```

11. Insert the next diskette and enter a Y. Continue inserting diskettes when requested until the entire volume is copied. (If the diskettes are in a 4966 diskette magazine, \$MOVEVOL automatically goes to the next diskette.)

If you enter an N, \$MOVEVOL gives you an opportunity to end the operation.

```
MOUNT NEXT DISKETTE  
REPLY -Y- WHEN DONE: Y  
PROCESSING DISKETTE VOLUME SAVnnn  
VOLUME DUMP OPERATION COMPLETE  
3600 RECORDS TRANSFERRED
```

```
$MOVEVOL ENDED 08:50:10
```

Saving the EDX System (*continued*)

Saving the System on Tape

This procedure saves an entire disk device, volume, or data set on tape.

If you are saving a copy of the entire disk, or the volume containing your supervisor, you should also create a diskette that can be used to IPL the system and keep it with the tape.

Steps 1 and 2 show how to create this diskette. If you do not require an IPL diskette containing your customized supervisor and tape support, start with step 3.

Note: We suggest that all disk volumes be compressed before they are saved. See “Compressing Volumes and Devices” on page OP-142 for the compress procedure.

When saving to tape, you must use the `$TAPEUT1 ST` command. The `ST` command has the following two functions:

- **Double-buffer** - uses two buffer areas to hold the output from tape or disk. By using two buffer areas, the tape drive or disk never has to wait for the system to empty the contents of the first buffer area. If sufficient storage is available (we recommend a 64K area), the system will always use the double- buffer function. In addition, when saving tapes using the double-buffer function, the system initializes and varies tapes online automatically.
- **Single-buffer** - uses one buffer area to hold output from tape or disk. As a result, the tape drive or disk has to wait for the system to empty the buffer area to disk.

If you have enough storage space in your partition, the system uses the double-buffer function. You must decide if you want automatic initialization. (You may reply `N` if you have already initialized your tape or you know you don't have the required space in your partition. Then you will have to initialize and vary tapes online manually.) If you choose automatic initialization, you will skip over several steps in this procedure.

Before You Start:

Observe the following requirements:

- Obtain an EDX format diskette, if you are backing up the volume containing the supervisor. (See “Initialize for EDX Format” on page OP-93 for the initialization procedure.)
- Obtain the hexadecimal device address of the tape, if it is not online. (`$TAPEUT1` displays this address when you load it.)
- Determine if you have room in the current partition for the double-buffer function. (We recommend that you have a 64K area of storage for the double-buffer function.) See “Listing Partitions and Active Programs” on page OP-227 for this procedure.
- If you cannot use the double-buffered function and auto initialization, initialize the tape with at least one data set. (See “Initializing Tapes” on page OP-112 to initialize the tape.)

Keeping Records and Backing-up the System

Saving the EDX System (*continued*)

- If you are saving the entire disk, obtain:
 - The hexadecimal device address of the disk. (See “Listing the Devices on Your System” on page OP-318 if you do not know this address.)
 - The tape data set name.
 - The tape device ID (TAPEID).
- If you are saving an individual volume, obtain:
 - The disk volume name.
 - The tape data set name.
 - The tape device ID (TAPEID).
- If you are saving an individual data set, obtain:
 - The name of the data set and volume on the disk.
 - The tape data set name.
 - The tape device ID (TAPEID).
- If you are backing up the volume containing the supervisor start with step 1, otherwise start with step 4.

1. Initialize and prepare a diskette as follows: (See “Initialize a Diskette Volume Directory” on page OP-106 for the step-by-step procedure.)

- Initialize as a single- or multi-volume diskette. (The example uses a single-volume diskette.)
- Assign any volume label.
- Initialize the volume to contain 20 data sets.
- Write verification is optional.
- Allocate \$EDXNUC.
- Initialize IPL text.

```
> $L $INITDSK
```

```
COMMAND (?): ID  
DEVICE ADDRESS: 02  
INITIALIZE DEVICE MAY DESTROY ALL DATA  
CONTINUE? Y  
INITIALIZE DISKETTE AS MULTI-VOLUME TYPE? N  
NEW VOLUME LABEL: SAV000  
  
ENTER OWNER IDENTIFICATION: BACKUP  
SINGLE-VOLUME TYPE DISKETTE INITIALIZED  
INITIALIZE THE VOLUME (Y/N)? Y  
MAXIMUM NUMBER OF DATASETS? 8  
DO YOU WANT WRITE VERIFY FOR THIS VOLUME (Y/N)? N  
ALLOCATE $EDXNUC (Y/N)? Y  
VOLUME INITIALIZED  
INITIALIZE IPL TEXT (Y/N)? Y  
IPL TEXT WRITTEN
```

```
COMMAND (?) : EN
```

```
$INITDSK ENDED AT 00:09:45
```

Saving the EDX System (*continued*)

2. Use the \$COPYUT1 CM command to copy \$LOADER and your nucleus, from volume EDX002 as shown in the example. If you have a 4978 display terminal as your system log terminal (\$SYSLOG), copy \$4978IS0 and \$4978CS0 also. (See "Copy a Data Set with Allocation" on page OP-152 for the step-by-step member copy procedure.)

```
> $L $COPYUT1
.
THE DEFINED SOURCE VOLUME IS EDX002 OK (Y/N)? Y
THE DEFINED TARGET VOLUME IS EDX002 OK (Y/N)? N
ENTER NEW TARGET VOLUME: SAV000
MEMBER WILL BE COPIED FROM EDX002 TO SAV000, OK? Y
COMMAND(?): CM
ENTER FROM (SOURCE) MEMBER: $LOADER
ENTER TO (TARGET) MEMBER OR * FOR
SAME NAME AS SOURCE: *
$LOADER COPY COMPLETE      nn RECORDS COPIED
COMMAND(?): _
```

3. Use the \$COPYUT1 CG command to copy all data sets starting with the prefix \$TAPE from volume EDX002 as shown in the example.

If you have sufficient storage and plan to use the auto initialization function skip to step 5.

```
COMMAND(?): CG
ENTER GENERIC TEXT: $TAPE
$TAPEUT1 COPY COMPLETE      nn RECORDS COPIED
.
.
COMMAND(?): _
```

Keeping Records and Backing-up the System

Saving the EDX System (*continued*)

4. Is the tape unit online and the tape positioned at the correct data set? (If you are not sure, load \$TAPEUT1, the first screen contains the answer.)

```
> $VARYON 4C 2 EX
IGNORE EXPIRATION DATE? (Y,N): _
```

- If so, continue with step 5.
- If not, press the attention key and enter \$VARYON *ioda* file *EX*, where:
 - *ioda* is the hexadecimal device address of the tape.
 - *file* is the decimal file number of the data set if the tape is labeled.
 - *EX* shows expiration date override. *EX* is required only if the tape is labeled and has an unexpired expiration date.

4(a). Do you want to be able to override an unexpired expiration date? (\$VARYON displays this message only if you entered the EX parameter.)

```
IGNORE EXPIRATION DATE? (Y,N): Y
MYVOL ONLINE
```

- If so, enter a Y.
- If not, enter an N.

5. Load the utility \$TAPEUT1 to start the save procedure:

```
> $L $TAPEUT1
LOADING $TAPEUT1 19P,08:30:45, LP= 7A00,
DUAL NL 1600 ONLINE
      DEVICE ADDRESS 4C
```

- Using the \$L operator command, press the attention key and enter \$L \$TAPEUT1.
- Using the session manager, select option 9.10.

```
COMMAND (?): _
```

6. Enter ST to save to tape.

```
COMMAND (?): ST
AUTO INITIALIZATION MODE? (Y/N): _
```

Saving the EDX System *(continued)*

7. Do you want automatic initialization?

- If so, enter a **Y** and continue with step 7(a).
- If not, enter a **N** and continue with step 8.

```
COMMAND (?): ST
AUTO INITIALIZATION MODE? (Y/N): _
```

7(a). Enter the tape identification and continue with step 8.

```
AUTO INITIALIZATION MODE? (Y/N): Y
ENTER TAPEID (1-6 CHARACTERS) :
```

8. Enter the number of the type of save you want and continue as follows:

- To save the entire disk, enter a **1** and continue with step 8(a).
- To save an individual volume, enter a **2** and continue with step 8(b).
- To save an individual data set, enter a **3** and continue with step 8(c).

Note: If you receive the insufficient storage message, exit from \$TAPEUT1. Reply **N** to the automatic initialization prompt in step 7, and manually initialize the tape.

```
*****
* WARNING: TO ENSURE PROPER *
* TAPE CONTENTS, THE SYSTEM *
* SHOULD BE INACTIVE WHILE *
* RUNNING THIS UTILITY *
*****

SAVE OPTIONS:
 1 = DEVICE
 2 = VOLUME
 3 = DATA SET
 4 = EXIT
ENTER SAVE OPTION: _
```

Keeping Records and Backing-up the System

Saving the EDX System *(continued)*

8(a). Enter the hexadecimal device address of the disk.

If you replied:

- **Y** to auto initialization, continue with steps 16 through 19.
- **N** to auto initialization. Continue with steps 9 through 15.

```
ENTER SAVE OPTION: 1
DISK IODA IN HEX(XX): 48
```

8(b). Enter the name of the disk volume being saved. If you replied:

- **Y** to auto initialization, continue with steps 16 through 19.
- **N** to auto initialization, continue with steps 9 through 15.

```
ENTER SAVE OPTION: 2
SOURCE DISK VOLUME NAME: EDX003
```

8(c). Enter the data set and volume name of the data set being saved. If you replied:

- **Y** to auto initialization, continue with steps 16 through 19.
- **N** to auto initialization, you will need to initialize and vary your tapes online manually. Continue with steps 9 through 15.

```
ENTER SAVE OPTION: 3
SOURCE (NAME,VOLUME): DATASET1,EDX003
```

9. Enter the name of the target tape data set and the volume label of the tape.

```
TARGET (NAME,VOLUME): BACKUP,MYVOL
xxx SAVE OF xxxx ONTO TAPE BACKUP,MYVOL OK?
(Y/N): _
```

Saving the EDX System *(continued)*

10. Did you enter the correct information:

- If so, enter a **Y**. (If the source requires only a single tape, continue with step 14.)
- If not, enter an **N**. (\$TAPEUT1 gives you a chance to correct the information.)

Note: If you want to cancel the copy operation while it is in progress, press the attention key and enter **CA**.

```
xxx SAVE OF xxxx ONTO TAPE BACKUP,TAPE01 OK?  
(Y/N): Y
```

```
USE ATTN/CA TO CANCEL THE SAVE  
END OF TAPE ENCOUNTERED. CONTINUE? _
```

11. Do you want to continue on another tape? (If you are saving an entire disk device, the contents may span more than one tape.)

- If so, enter a **Y**.
- If not, enter an **N**. Skip to step 15.

```
END OF TAPE ENCOUNTERED. CONTINUE? Y  
ENTER NEXT TAPE (NAME,VOLUME): _
```

12. Enter the name of the next tape data set and the volume label of the tape.

```
ENTER NEXT TAPE (NAME,VOLUME): BACKUP2,TAPE01  
REPLY Y WHEN TAPE BACKUP2,TAPE01 IS MOUNTED  
AND VARIED ONLINE? _
```

Keeping Records and Backing-up the System

Saving the EDX System (*continued*)

13. Mount the new tape and:

- Press the attention key.
- Press the enter key.
- Enter \$VARYON *ioda* file *EX*, where:
 - *ioda* is the hexadecimal device address of the tape.
 - *file* is the decimal file number of the data set if the tape is labeled.
 - *EX* shows expiration date override. *EX* is required only if the tape is labeled and has an unexpired expiration date.)
- Press the enter key again.

```
REPLY Y WHEN TAPE BACKUP2,TAPE01 IS MOUNTED
AND VARIED ONLINE? _
> $VARYON 4C 2 EX _
IGNORE EXPIRATION DATE? (Y,N): _
```

13(a). Do you want to be able to override an unexpired expiration date? (\$VARYON displays this message only if you entered the *EX* parameter.)

- If so, enter a **Y**.
- If not, enter an **N**.

```
IGNORE EXPIRATION DATE? (Y,N): Y
MYVOL2 ONLINE
? _
```

14. Enter a **Y** to continue.

If you receive the message END OF TAPE ENCOUNTERED, CONTINUE?, go to step 11. If not, continue with step 15.

```
? Y
xxxx SAVED

SAVE OPTIONS:
1 = DEVICE
2 = VOLUME
3 = DATA SET
4 = EXIT
ENTER SAVE OPTION: _
```

Saving the EDX System *(continued)*

15. Are you finished?

- If so, enter a **4** to end \$TAPEUT1.
- If not, continue with step 8.

```
ENTER SAVE OPTION: 4
$TAPEUT1 ENDED AT 09:45:00
```

16. Reply **Y** to continue.

Notes:

1. Reply **Y** if you want the system to write over old data even if the tape has not expired.
2. Reply **N** if you do not want to use the tape. Exit from \$TAPEUT1.

```
TAPE EXPIRATION DATE WILL NOT BE CHECKED PRIOR
TO INITIALIZATION CONTINUE? (Y/N): Y
```

17. Fill in the current expiration date.

```
TAPE EXPIRATION DATE WILL NOT BE CHECKED PRIOR
TO INITIALIZATION CONTINUE? (Y/N): Y
```

```
INPUT EXPIRATION DATE (YYDDD): _
```

18. You will receive a message that your tape has been initialized. Reply **Y** to the next prompt to continue.

```
TAPE##00,DATSAV TAPE INITIALIZED
```

```
xxx SAVE OF xxxx ONTO TAPE
TAPE##00,DATSAV OK? (Y,N): Y
```

Keeping Records and Backing-up the System

Saving the EDX System *(continued)*

19. Do you want to continue on another tape? (If you are saving an entire disk device, the contents may span more than one tape.)

- If so, enter a Y.
- If not, enter an N. Skip to step 20.

```
END OF TAPE ENCOUNTERED. CONTINUE? Y
ENTER NEXT TAPE (NAME,VOLUME): _
```

20. Are you finished?

- If so, enter a 4 to end \$TAPEUT1.
- If not, continue with step 6.

```
XXXX saved

SAVE OPTIONS:
1 = DEVICE
2 = VOLUME
3 = DATA SET
4 = EXIT
```

Restoring the System

This section contains procedures for restoring your system from a backup copy on either diskette or tape.

Restoring from Diskette

Use this procedure to restore your system from a backup copy on diskette. The copy must have been made using the \$MOVEVOL utility as described under "Saving the System on Diskettes" on page OP-319.

Before You Start:

Obtain the following information:

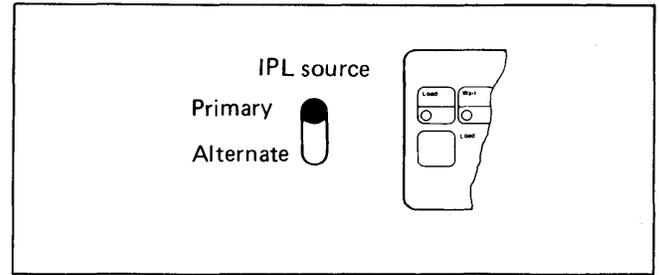
- The name of the volume being restored.
- The name of the source diskette data set.
- The volume label of the source diskette. (The volume label is xxxnnn, where xxx is common to all the diskettes in the set, and nnn is sequence numbers starting with 000.)

You must also insert the first diskette (volume label xxx000) into the diskette drive, or all diskettes into 4966 diskette magazines. If you are using magazines, the diskettes must be in the magazine in the correct sequence with no empty slots.

Restoring the System (*continued*)

1. IPL the system from the first diskette in the backup set, if you are restoring the volume containing the system nucleus and the current disk resident nucleus is not operational.

If you cannot IPL from disk and diskette xxx000 is not set up to IPL the system, you must use the starter system and generate a system that will copy the diskettes. (Refer to the *Installation and System Generation Guide* for system generation information.)



2. Load the utility \$MOVEVOL:

- Using the \$L operator command, press the attention key and enter \$L \$MOVEVOL.
- Using the session manager, select option 3.8. If you are using the session manager, steps 3 and 4 are combined in one parameter input menu.

```
> $L $MOVEVOL
DISK (NAME,VOLUME): _ _
```

3. Enter \$\$EDXLIB,volname, where volname is the name of the disk volume being restored.

```
DISK (NAME,VOLUME): $$EDXLIB,EDX002
DISKETTE (NAME,VOLUME): _
```

4. Enter the name of the diskette data set and the diskette volume label.

```
DISKETTE (NAME,VOLUME): EDX002,SAV000
LOADING $MOVEVOL 20P,08:35:45, LP=5200,...
DUMP VOLUME EDX002 FROM DISK TO DISKETTE: _
```

Keeping Records and Backing-up the System

Restoring the System *(continued)*

5. Enter an N because you are restoring the volume.

```
DUMP VOLUME EDX002 FROM DISK TO DISKETTE: N
RESTORE VOLUME FROM DISKETTE TO DISK VOLUME EDX002: _
```

6. Enter a Y to restore the volume.

```
RESTORE VOLUME FROM DISKETTE TO DISK VOLUME EDX002: Y
RESTORING DUMP OF EDX002 - DATE IS 05/19/82
CONTINUE? _
```

7. Do you have the correct backup copy?

- If so, enter a Y.
- If not, enter an N and return to step 3 on page OP-335.

```
RESTORING DUMP OF EDX002 - DATE IS 05/19/82
CONTINUE? Y
PROCESSING DISKETTE VOLUME SAV000
MOUNT NEXT DISKETTE
REPLY -Y- WHEN DONE: _
```

8. Insert the next diskette and enter a Y. Continue inserting diskettes when requested until the entire volume is copied. If the diskettes are in a 4966 diskette magazine, \$MOVEVOL automatically goes to the next diskette.

```
MOUNT NEXT DISKETTE
REPLY -Y- WHEN DONE: Y
PROCESSING DISKETTE VOLUME SAVnnn
VOLUME INSTALLED
      3600 RECORDS TRANSFERRED

$MOVEVOL ENDED 08:50:10
```

9. After IPL completes, load the utility \$INITDSK by typing \$L \$INITDSK.

```
> $L $INITDSK
LOADING $INITDSK      83P,08:30:20, LP= 0000,...
$INITDSK - DISK INITIALIZATION UTILITY
COMMAND (?): _
```

Restoring the System (*continued*)

10. Enter **II** nucleus volume where **nucleus** is the name of the nucleus being initialized, and **volume** is the name of the volume containing the nucleus.

```
COMMAND (?): II $EDXNUC EDX002
IPL TEXT WRITTEN
COMMAND (?): _
```

11. Enter **EN** to end \$INITDSK. You should now be able to IPL from the disk.

```
COMMAND (?): EN
$INITDSK ENDED AT 08:45:00
```

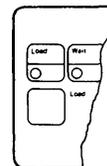
12. After you have restored the nucleus volume, set the IPL source switch to IPL from the disk, and IPL the system.

09100

IPL source

Primary

Alternate



Keeping Records and Backing-up the System

Restoring the System (*continued*)

Restoring from Tape

Use this procedure to restore an entire disk, an individual volume, or an individual data set from tape.

When restoring from tape, you must use the `$TAPEUT1 RT` command. The `RT` command has the following two functions:

- **Double-buffer** - uses two buffer areas to hold the output from tape or disk. By using two buffer areas, the tape drive or disk never has to wait for the system to empty the contents of the first buffer area. If sufficient storage is available (we recommend a 64K area), the system will always use the double-buffer function. In addition, when saving tapes using the double-buffer function, the system initializes and varies tapes online automatically.
- **Single-buffer** - uses one buffer area to hold output from tape or disk. As a result, the tape drive or disk has to wait for the system to empty the buffer area to disk.

If you have enough storage space in your partition, the system uses the double-buffer function. If the system does not use the double-buffer function you have to vary tapes online manually.

Before You Start:

Consider the following:

- If the tape is not online, you need the hexadecimal device address of the tape. (`$TAPEUT1` lists this address when you load it.)
- Determine if you have room in your partition for the double-buffer function. (We recommend that you have a 64K area of storage for the double-buffer function.) See "Listing Partitions and Active Programs" on page OP-227 for this procedure.
- If you are restoring the entire disk, you need:
 - The hexadecimal device address of the disk. (See "Listing the Devices on Your System" on page OP-318 to determine this address.)
 - The tape data set name. (The data set name can be obtained from the physical label on the tape or from your tape log.)
 - The tape ID (`TAPEID`).
- If you are restoring an individual volume, you need:
 - The disk volume name.
 - The tape data set name. (The data set name can be obtained from the physical label on the tape or from your tape log.)
 - The tape ID (`TAPEID`).
- If you are restoring an individual data set, you need:
 - The disk data set and volume name of the data set being restored.
 - The tape data set name. (The data set name can be obtained from the physical label on the tape or from your tape log.)
 - The tape ID (`TAPEID`).

Restoring the System (*continued*)

1. Are you restoring a back up copy of the system?

- If so, IPL the system from the diskette created for this purpose or from the current disk resident nucleus.

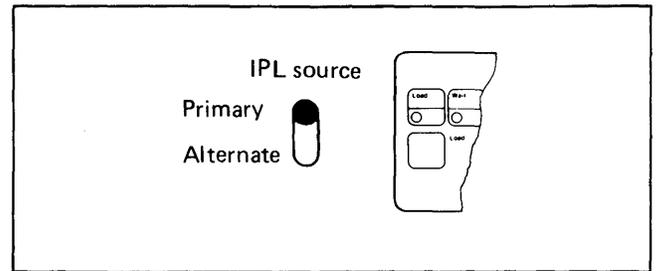
If you cannot IPL from disk and a diskette to IPL the system was not created, you must use the starter system and generate a system that will copy the tapes. (Refer to the *Installation and System Generation Guide* for system generation information.)

- If not, skip to step 2.

Note: If you have sufficient storage and plan to use the auto initialization function, skip to step 3. (To use the auto initialization function, we recommend a 64K area of storage.)

2. Is the tape online?

- If so, continue with step 3.
- If not, press the attention key and enter **\$VARYON ioda file** to vary the tape online to the file number of the data set being restored where:
 - *ioda* is the hexadecimal device address of the tape drive.
 - *file* is the decimal file number being accessed if you are using a labeled tape. (In the example, we are accessing file number 2.)



```
> $VARYON 4C 2
MYVOL ONLINE
```

Keeping Records and Backing-up the System

Restoring the System *(continued)*

3. Load the utility \$TAPEUT1:

- Using the \$L operator command, press the attention key and enter \$L \$TAPEUT1.
- Using the session manager, select option 9.10.

```
> $L $TAPEUT1
LOADING $TAPEUT1 19P,08:30:45, LP= 7A00,...
TAPE01 DUAL NL 1600 ONLINE
DEVICE ADDRESS 4C

COMMAND (?): _
```

4. Enter RT to restore from tape.

- If you varied the tape online manually in step 2, continue with step 5.
- If not, continue with step 4(a).

```
COMMAND (?): RT
*****
* WARNING: TO ENSURE PROPER *
* DISK CONTENTS, THE SYSTEM *
* SHOULD BE INACTIVE WHILE *
* RUNNING THIS UTILITY *
*****

RESTORE OPTIONS:
1 = DEVICE
2 = VOLUME
3 = DATA SET
4 = EXIT
ENTER RESTORE OPTION: _
```

4(a). Reply Y to auto varyon mode. Enter the tape identification.

```
COMMAND (?): RT
AUTO VARYON MODE? (Y,N): Y
ENTER TAPEID (1-6 CHARACTERS): TAPE01
*****
* WARNING: TO ENSURE PROPER *
* DISK CONTENTS, THE SYSTEM *
* SHOULD BE INACTIVE WHILE *
* RUNNING THIS UTILITY *
*****

RESTORE OPTIONS:
1 = DEVICE
2 = VOLUME
3 = DATA SET
4 = EXIT
ENTER RESTORE OPTION: _
```

Restoring the System (*continued*)

5. Enter the number of the type of restore you want:

- Enter a **1** and continue with step 6(a) to restore the entire disk.
- Enter a **2** and continue with step 6(b) to restore an individual volume.
- Enter a **3** and continue with step 6(c) to restore an individual data set.

6(a). Enter the hexadecimal device address of the disk.

- If you are using the auto varyon function, continue with step 7(a).
- If not, continue with step 7(b).

```
ENTER RESTORE OPTION: 1
ENTER DISK IODA IN HEX(XX): 48
```

6(b). Enter the name of the disk volume to which the tape is being copied.

- If you are using the auto varyon function, continue with step 7(a).
- If not, continue with step 7(b).

```
ENTER RESTORE OPTION: 2
ENTER TARGET DISK VOLUME NAME: EDX003
```

6(c). Enter the data set and volume name of the data set to which the tape is being copied.

- If you are using the auto varyon function, continue with step 7(a).
- If not, continue with step 7(b).

```
ENTER RESTORE OPTION: 3
ENTER TARGET (NAME,VOLUME): DATASET1,EDX003
SOURCE (NAME,VOLUME): _
```

Keeping Records and Backing-up the System

Restoring the System *(continued)*

7(a). Enter the name of source tape data set and the volume label of the tape.

```
SOURCE (NAME,VOLUME): BACKUP,MYVOL
```

```
TAPE BACKUP, MYVOL ONLINE
```

```
xxx RESTORE OF xxx FROM TAPE BACKUP,MYVOL OK? (Y/N): _
```

7(b). If you had sufficient storage the system automatically varies the tape online. (You receive the message "TAPEXXXX,XXXX ONLINE" as shown in the screen for step 7(a).

```
ENTER TARGET (NAME,VOLUME)DATASET1,EDX003
```

```
INSUFFICIENT MEMORY TO LOAD DOUBLE BUFFERED UTILITY  
AUTOMATIC VARYON NOT AVAILABLE WITH SINGLE BUFFER  
RESTORE UTILITY
```

If you receive the insufficient storage message, vary the tape online manually as shown in step 2 of this procedure.

8. Did you enter the correct information?

- If so, enter a **Y**. If the source is on a single tape, continue with step 11.
- If not, enter an **N**. Continue with step 5(a), 5(b), or 5(c).

```
xxx RESTORE OF xxx FROM TAPE BACKUP,TAPE01 OK?  
(Y/N): y
```

```
USE ATTN/CA TO CANCEL THE RESTORE
```

Note: If you want to cancel the copy operation while it is in progress, press the attention key and enter **CA**.

Restoring the System (*continued*)

9. If you are restoring an entire disk device, the contents may span more than one tape. If this is the case, \$TAPEUT1 asks you to mount the next tape, vary it online, and respond with a Y as shown in step 9.

Note: If you have the automatic varyon function, mount the tape and skip to step 10.

10. After you mount the tape:

- Press the attention key.
- Enter \$VARYON *ioda* where *ioda* is the hexadecimal device address of the tape device.
- Press the enter key.

```
MOUNT BACKUP2,TAPE02
REPLY Y WHEN TAPE MOUNTED AND VARIED ONLINE?
```

```
REPLY Y WHEN TAPE MOUNTED AND VARIED ONLINE? _
> $VARYON 4C
TAPE02 ONLINE
? _
```

11. Enter a Y to continue.

If you receive a message to mount another tape, continue with step 8. If not, continue with step 11.

```
? Y
DISK RESTORED

RESTORE OPTIONS:
1 = DEVICE
2 = VOLUME
3 = DATA SET
4 = EXIT
ENTER RESTORE OPTION: _
```

Keeping Records and Backing-up the System

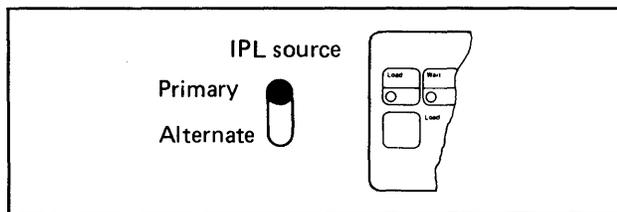
Restoring the System *(continued)*

12. Are you finished?

- If so, enter a **4** to end \$TAPEUT1.
Continue with step 13.
- If not, continue with step 5.

ENTER RESTORE OPTION: 4
\$TAPEUT1 ENDED AT 09:45:00

13. Initialize your IPL text. (Refer to “Compressing Volumes and Devices” on page OP-142 and follow steps 10(b) through 10(d).) When the nucleus is restored, set the IPL source switch to IPL the system from the disk and IPL the system.



Appendix A. Reference Material

This chapter contains reference material that helps you use your EDX system. It contains a cross reference chart that shows you the operator command or system utility to use for various jobs. This chapter also contains syntax descriptions for the operator commands and \$JOBUTIL commands, explanations of the printer spooling display formats, and formulas for estimating the required size of your spool data set.

The information in this chapter is organized as follows:

- “Job Cross Reference Chart” on page OP-346
- “Operator Command Syntax” on page OP-357
- “\$JOBUTIL Command Syntax” on page OP-376
- “Spool Status Display Formats” on page OP-391
- “Estimating Spool Data Set Size” on page OP-395

Reference Material

Job Cross Reference Chart

The following chart directs you to the operator command or utility that will help you do a particular job. If instructions for that job are included in the *Operation Guide (OP)*, *Event Driven Executive Language Programming Guide (PG)*, *Problem Determination Guide (PD)*, or *Installation and System Generation Guide (IS)*, the page prefix of the guide is listed in the fourth column - "Guide Information".

Job	Operator Command/ Utility	Session Manager Option	Guide Information
Allocate a data set	\$DISKUT1 \$TAPEUT1	3.1 3.10	OP, IS
Allocate an H-exchange data set	\$HXUT1	3.11	OP
Allocate a member in a graphics data base	\$DICOMP	5.2	-
Allocate a volume	\$INITDSK	3.7	OP, IS
Allocate a volume under a fixed head	\$INITDSK	3.7	OP, IS
Analyze GPIB errors	\$GPIBUT1	4.9	CO, PD
Analyze hardware errors	\$DUMP	9.1	PD
Analyze program checks	\$DUMP	9.1	PD
Analyze tape surface for defects	\$TAPEUT1	3.10	—
Browse a source data set	\$FSEDIT	1	—
Cancel a program	\$C	—	OP
Change hard-copy device	\$TERMUT1	4.1	OP
Change print screen PF key	\$TERMUT1	4.1	OP
Change spool job attributes	\$\$SPLUT1	4.7	OP
Change a tape label	\$TAPEUT1	3.10	OP
Change a terminal address	\$TERMUT1	4.1	OP
Close job created by \$DEBUG	\$DEBUG	—	—
Compile an EDL program	\$EDXASM	2.1	PG, OP
Compile and link edit an EDL program	\$EDXASM \$EDXLINK	2.2	PG, OP
Compile a program on a S/370 and execute it on a Series/1	\$UPDATEH	2.10	—

Figure 24 (Part 1 of 7). Job Cross-reference Chart

Job Cross Reference Chart (continued)

Job	Operator Command/Utility	Session Manager Option	Guide Information
Compress one volume or all volumes on a device	\$COMPRES	3.4	OP
Controlling the job queue processor	\$JOBQUT	10.1	OP
Control printer spooling	\$SPLUT1	4.7	OP
Convert object modules to executing code	\$EDXLINK \$UPDATE	2.7 2.9	PG
Copy all data sets	\$COPYUT1	3.3	OP
Copy a data set with automatic allocation	\$COPYUT1	3.3	IS, OP
Copy all data sets with the same prefix	\$COPYUT1	3.3	OP
Copy a basic exchange data set	\$COPY	3.5	OP
Copy a disk/diskette data set to tape	\$TAPEUT1	3.10	OP
Copy an H-exchange data set	\$HXUT1	3.11	OP
Copy a tape data set to diskette	\$TAPEUT1	3.10	OP
Copy a tape data set to tape	\$TAPEUT1	3.10	OP
Copy a \$TRAP data set to two diskettes	\$COPY	3.5	OP
Copy a volume or data set to an allocated volume or data set	\$COPY	3.5	OP
Create an upper/lower case data set	\$FSEDIT \$TERMUT2	1 4.2	OP
Create your own terminal keyboard characters	\$FONT	4.5	—
Date and time, display	\$W	—	OP
Date and time, set	\$T	—	OP
Delete job queue	\$JOBQUT	10.1	OP
Delete a data set	\$DISKUT1	3.1	OP
Delete all data sets with the same prefix	\$DISKUT1	3.1	OP
Delete a volume	\$INITDSK	3.7	OP
Determine address of a data set	\$DISKUT1	3.1	—

Figure 24 (Part 2 of 7). Job Cross-reference Chart

Reference Material

Job Cross Reference Chart (continued)

Job	Operator Command/ Utility	Session Manager Option	Guide Information
Determine how many records a data set contains	\$DISKUT1	3.1	OP
Determine terminal name, address, type and partition assignment	\$TERMUT1	4.1	OP
Determine how much free space is on a volume	\$DISKUT1	3.1	OP
Determine the hardware supported by supervisor	\$IOTEST	9.3	OP, IS
Device, set offline	\$VARYOFF	—	OP
Device, set online	\$VARYON	—	OP
Display data/volume/storage contents in decimal, EBCDIC, or hexadecimal	\$DEBUG \$DISKUT2	— 3.2,9.2	PD, OP
Dump storage to a data set on error condition	\$TRAP	—	PD, OP
Dump unmapped storage	\$TRAP \$DUMP	—	PD, OP
Estimate progress of compress	\$COMPRES	3.4	OP
Eject a page	\$E	—	OP
End job queue processing	\$JOBQUT	10.1	PD
Enter source statements	\$FSEDIT	1	PG, OP
Erase contents of a data set	\$DISKUT2	3.2,9.2	—
Erase display screen	\$B	—	OP
Format and print BSC trace data sets	\$BSCTRCE \$BSCUT1	8.1 8.2	CO, PD
Format and print error log information	\$DISKUT2	3.2,9.2	OP, PD
Generate a supervisor	\$XPSLINK	2.8 2.13	IS
Hold a job in the job queue processor	\$JOBQUT	10.1	OP
Identify the program function keys on a 4978 and 4980 display station	\$PFMAP	4.6	—
Increase the size of a data set	\$DISKUT1 \$COPY	3.1 3.5	—
Initialize a disk	\$DASDI \$MEMDISK	3.6 —	IS, OP
Initialize unmapped storage as a disk	\$MEMDISK	—	OP

Figure 24 (Part 3 of 7). Job Cross-reference Chart

Job Cross Reference Chart (continued)

Job	Operator Command/ Utility	Session Manager Option	Guide Information
Initialize a diskette to EDX format	\$DASDI	3.6	OP
Initialize a diskette to exchange format	\$DASDI	3.6	OP
Initialize a diskette for a stand-alone/\$TRAP dump	\$DASDI	3.6	OP
Initialize a tape	\$TAPEUT1	3.10	OP
Invoke your own operator command	\$U	—	OP, CU
IPL a remote Series/1	\$S1S1UT1	4.8	CO
Job queue processing controller	\$JOBQUT	10.1	OP
Job queue job submission utility	\$SUBMIT	10.2	OP
Link edit programs	\$EDXLINK	2.7	PG, OP
List all data sets	\$DISKUT1	3.1	OP
List data sets by type (program or data)	\$DISKUT1	3.1	OP
List all data sets with same prefix	\$DISKUT1	3.1	OP
List a directory	\$INITDSK	3.7	OP
List all volumes on all disk/diskette devices	\$DISKUT1 \$INITDSK	3.1	OP
List all volumes on a particular device	\$INITDSK	3.7	OP
List information about one data set	\$DISKUT1	3.1	OP
List messages in source message data set	\$MSGUT1	2.14	OP
Load and execute a program	\$L	—	OP
Load 4980 terminal	\$TERMUT2	4.2	OP
Loading control image	\$TERMUT2	4.2	OP
Loading control store	\$TERMUT2	4.2	OP
Locate required data sets/overlays before program execution	\$PREFIND	2.11	—
Message-source processing	\$MSGUT1	2.14	OP

Figure 24 (Part 4 of 7). Job Cross-reference Chart

Reference Material

Job Cross Reference Chart (continued)

Job	Operator Command/Utility	Session Manager Option	Guide Information
Modify a source data set	\$FSEDIT	1	OP,PG
Partition assigned to terminal, change	\$CP	—	OP
Prepare Version 1 and 2 data sets for use with Version 5	\$MIGRATE \$MIGRID \$MIGCOPY	—	IS
Print a formatted \$TRAP dump	\$DUMP	9.1	OP, PD
Print a stand-alone dump	\$DUMP	9.1	OP, PD
Programs, display all active	\$A ALL	—	OP
Programs, display all active in terminal's partition	\$A	—	OP
Recognizing intermittent program errors	\$TRAP	—	OP, PD
Recording hardware errors	\$LOG	—	PD
Recover a backed-up data set	\$MOVEVOL	3.8	OP
Redefine terminal scrolling	\$TERMUT1	4.1	—
Rename a disk/diskette volume	\$INITDSK	3.7	OP
Rename a terminal	\$TERMUT1	4.1	OP
Restart job queue processing	\$JOBQUT	10.1	OP
Retrieve a data set from a host	\$EDIT1N	—	—
Running batch job streams	\$JOBUTIL \$SUBMIT	7 10.2	OP
Running multiple jobs	\$JOBUTIL \$SUBMIT	7 10.2	OP
Send a data set to a host	\$HCFUT1	8.8	CO
Send a message to a terminal	\$TERMUT3	4.3	OP
Setting PF keys	\$TERMUT2	4.2	OP
Setting a terminal offline or online	\$TERMUT1	4.1	OP
Sort a disk/diskette volume directory	\$DIRECT	—	—

Figure 24 (Part 5 of 7). Job Cross-reference Chart

Job Cross Reference Chart (continued)

Job	Operator Command/ Utility	Session Manager Option	Guide Information
Spooling, change forms type	\$\$ ALT	—	OP
Spooling, change heading id	\$\$ ALT	—	OP
Spooling, change number of copies	\$\$ ALT	—	OP
Spooling, change printer	\$\$ ALT	—	OP
Spooling, delete a job	\$\$ DEL	—	OP
Spooling, delete a set of jobs	\$\$ DG	—	OP
Spooling, display status of jobs	\$\$ DISP	—	OP
Spooling, display status of resources	\$\$ DISP	—	OP
Spooling, display status of writers	\$\$ DISP	—	OP
Spooling, hold jobs	\$\$ HOLD	—	OP
Spooling, keep a job after printing	\$\$ KEEP	—	OP
Spooling, release held jobs	\$\$ REL	—	OP
Spooling, restart a writer	\$\$ WRES	—	OP
Spooling, start	\$L \$SPOOL	—	OP
Spooling, start a writer	\$\$ WSTR	—	OP
Spooling, stop	\$\$ STOP	—	OP
Spooling, stop a writer	\$\$ WSTP	—	OP
Storage, change contents (patch)	\$P	—	—
Storage, display contents	\$D	—	OP
Storage map, display for all partitions	\$A ALL	—	OP
Storage map, display for terminal's partition	\$A	—	OP
Submit a job into a host batch job stream	\$EDIT1N	—	—
Suspend job queue processing	\$JOBQUT	10.1	OP

Figure 24 (Part 6 of 7). Job Cross-reference Chart

Reference Material

Job Cross Reference Chart (continued)

Job	Operator Command/Utility	Session Manager Option	Guide Information
Test the binary synchronous access method (BSCAM)	\$BSCUT2	8.3	CO, PD
Test tape label type to ensure I/O commands executing correctly	\$TAPEUT1	3.10	—
Test the operation of sensor I/O features	\$IOTEST	9.3	—
Trace I/O activities on a BSC line	\$BSCTRCE	8.1	CO, PD
Transport data sets across a bisync line	\$TRANS	—	OP
Trap storage image on error condition	\$TRAP	—	PD
Trap to two diskettes	\$TRAP	—	PD, OP
Verify BSC hardware and software assignments	\$BSCUT2	8.3	CO, PD
Verify Series/1-Series/1 attachment performing correctly	\$S1S1UT1	4.8	CO, PD
Write a tape label	\$TAPEUT1	3.10	OP

Figure 24 (Part 7 of 7). Job Cross-reference Chart

\$FSEDIT PF Keys and Edit Commands

\$FSEDIT uses six program function keys and a set of special editing commands. Figure 25 lists the functions of the PF keys, and Figure 26 on page OP-354 lists the editing commands.

\$FSEDIT PF Keys

The following chart lists the functions that can be done with the PF keys under \$FSEDIT. The chart shows both the standard set of PF keys and the starter system set.

Function	Standard	Starter System
Attention function on 3101.	PF8	-
Enter function on a 3101 during insert mode. ⁸	PF7	-
Erase all new entries and re-display screen image. (Edit mode only)	PF1	PF1
End \$FSEDIT except in browse and edit modes.	PF3	PF5
Print the current screen image on the system printer (\$SYSPRTR). (Browse, and edit modes only. The print can be stopped by pressing the attention key and entering CA.)	PF6	PF6
Repeat the previous CHANGE primary command. (Edit mode only.)	PF5	PF4
Repeat the previous FIND primary command. (Browse and edit modes only.)	PF4	PF2
Scroll down to display the next group of lines toward the end of the data set. The SCROLL==> field in the upper-right of the display screen determines the number of lines scrolled. (Directory list, browse, and edit modes only.)	PF3	PF5
Scroll up to display the next group of lines toward the beginning of the data set. The SCROLL==> field in the upper-right of the display screen determines the number of lines scrolled. (Directory list, browse, and edit modes only.)	PF2	PF3

Figure 25. \$FSEDIT Program Function Keys

⁸ To insert characters on a 3101 display terminal while using \$FSEDIT, you must delete characters at the end of the line before attempting to insert any in the middle of the line.

Reference Material

\$FSEDIT PF Keys and Edit Commands (*continued*)

\$FSEDIT Edit Commands

The following chart lists the browse and edit operations for \$FSEDIT and shows the option or command that does the operation.

Some of the commands have two forms; an abbreviation and the full command. The full command is shown in parenthesis next to the abbreviation. You can enter either the abbreviation or the full command.

Operation	Command	Parameters	Where used
After line - copy/move destination	A	none	Edit line command
Before line - copy/move destination	B	none	Edit line
Browse selected data set. (Data set is first read into work data set.)	B	none	Directory line
Browse contents of the work data set	BR(BROWSE)	none	Directory list
Cancel session and return to previous screen	CANC(CANCEL)	none	Browse/edit primary command
Change character strings	C(CHANGE)	/text1/text2/option text1 = the change from text text2 = the change to text / = any delimiter not in the text string. Three delimiters, all the same, must be used. option = NEXT, FIRST, or ALL	Edit primary
Change volume	CV	volname	directory list
Clear line commands, line numbers, mask, and cols.	RESET	none	Edit primary
Clear work data set	CL(CLEAR)	none	Edit primary
COBOL line numbers	COBOL	none	Edit primary
Copy lines of text	C or CC	none C - on one line CC - on first and last lines of a block	Edit line
Delete text	D or DD	none D - on one line DD - on first and last lines of a block	Edit line
Display insert mask	MASK	none	Edit line
Display columns	COLS	none	Edit line

Figure 26 (Part 1 of 3). \$FSEDIT Command Summary

\$FSEDIT PF Keys and Edit Commands (*continued*)

Operation	Command	Parameters	Where used
Display contents of work data set	BROWSE	none	Primary option 1
Display data in uppercase or lowercase	CAPS	ON OFF ALL ON = uppercase only OFF = uppercase and lowercase ALL = edit mode only, converts data set to all uppercase	Browse/edit primary command
Edit selected data set. (Data set is first read into work data set.)	E	none	Directory line
Edit contents of the work data set	ED(EDIT)	none	Directory list
Edit work data set	EDIT	none	Primary option 2
End \$FSEDIT	END	none	Primary option 8
Find a data set in list.	L(LOCATE)	dsname dsname = name of the data set.	Directory list
Find a line number	L(LOCATE)	line-number line-number = the number of the line you want	Browse/edit primary
Find a text string	F(FIND)	/text/option text = the string to be found / = any delimiter not in text Two delimiters, both the same, must be used. option = NEXT, or FIRST	Browse/edit primary
Insert text.	I or II	none I - on one line II - on first and last lines of a block	Edit line
Merge data from a source data set into work data set	MERGE	none	Primary option 7
Move line of text	M or MM	none M - on one line MM - on first and last lines of block	Edit line
Print current screen	PS	devname devname = the name of the printer. The default is \$SYSPRTR.	Edit primary
Print work data set on \$SYSPRTR	PRINT	none	Primary option 6
Read data set from native/host into work data set	READ	none	Primary option 3

Figure 26 (Part 2 of 3). \$FSEDIT Command Summary

Reference Material

\$FSEDIT PF Keys and Edit Commands *(continued)*

Operation	Command	Parameters	Where used
ReNUMBER data set	R(RENUM)	first increment first = the number assigned to the first line increment = the amount used in generating the line numbers	Edit primary
Return to previous menu	E(END)	none	Directory list, browse/edit primary
Return to primary option menu	M(MENU)	none	Directory list, browse/edit primary
Save contents of work data set	WRITE	dataset,volume dataset = the name of the data set where you want the work data set saved volume = the name of the volume containing the data set	Edit primary
Scroll to bottom of directory list	B(BOTTOM)	none	Directory list
Scroll to top of directory list	T(TOP)	none	Directory list
Set tab stops	TABS	loc1,loc2,...loc10 loc = the column number where you want a tab. Up to 10 tabs can be set.	Edit primary
Shift line two spaces to the left	(none	Edit line
Shift line two spaces to the right)	none	Edit line
After the first blank, shift the first block of data two spaces to the right. (A block of data is a group of characters separated by no more than one blank.)	>	none	Edit line
After the first blank, shift the first block of data two spaces to the left. (A block of data is a group of characters separated by no more than one blank.)	<	none	Edit line
Sort directory data set list.	S(SORT)	option D = date of last update N = data set name (default) R = reset sort option to NO SORT U = session manager user ID	Directory list
Submit batch job to host system	SUBMIT	none	Primary option 5
Tutorial text for using \$FSEDIT	HELP	none	Primary option 9
Write work data set to native/host	WRITE	none	Primary option 4

Figure 26 (Part 3 of 3). \$FSEDIT Command Summary

Operator Command Syntax

This section contains the syntax for each operator command. See “Entering Commands and Answering Prompts” on page OP-41 for an explanation of how to enter them.

Syntax Conventions

The following conventions are used in the presentation of the operator and \$JOBUTIL command syntax:

- Uppercase** If a parameter is shown in all uppercase letters, enter it exactly as shown.
- Lowercase** If a parameter is shown in all lowercase letters, substitute a variable value.
- []** If two or more parameters are enclosed in brackets, choose one of them. The parameters within the brackets are separated by the *or* (|) character.
- |** If parameters are separated by the *or* (|) character, choose one of them.

\$A - List Partitions and Active Programs

Use the \$A operator command to list the storage partitions defined for your EDX operating system. (A partition is a portion of storage in which programs run.) When you enter the \$A command, EDX tells you the starting address and size of each partition. It also tells you the names and address of each program active in each partition.

You can use \$A to list *all* partitions or only the one your terminal is assigned to. The \$A operator command is used in the instructions for “Listing Partitions and Active Programs” on page OP-227.

Syntax:

\$A	ALL blank
Required:	none
Default:	none

Operands	Description
ALL	Displays the starting addresses and sizes of all partitions and the active programs in each partition.
blank	If you do not enter ALL, EDX displays the starting addresses of the programs active in the partition where your terminal is running.

Reference Material

Operator Command Syntax (*continued*)

\$B - Blank Display Screen

Use the \$B operator command to erase all information on your display terminal screen. \$B is used in the instructions for "Erasing the Screen" on page OP-39.

Syntax:

\$B
Required: none
Default: none

Operands Description

None None

\$C - Cancel Program

Use the \$C operator command to cancel a program running in the same partition as your terminal. If there is more than one program of the same name, EDX asks (prompts) you for the load address of the program. The load address is the storage address where the program starts. You can find this address using the \$A operator command as described under "Listing Partitions and Active Programs" on page OP-227.

\$C should not be used as the normal means of stopping a program. It is used in the instructions for "Cancelling Programs" on page OP-236.

Note: \$C should not be used to cancel some of the system utilities. If \$C should not be used, the utility warns you on the first screen it displays.

Syntax:

\$C	program
Required:	program
Default:	none

Operands Description

program The name of the program to cancel. (See "Listing Partitions and Active Programs" on page OP-227, to determine the names of programs running in the system.)

Operator Command Syntax *(continued)*

\$CP - Change Display Terminal's Partition Assignment

Some jobs require that the display terminal be running in the same partition as the object of the job, such as \$C to cancel a program. Use the \$CP operator command to change the partition number for the terminal you are using.

\$CP is used in the instructions for "Changing a Terminal's Partition Assignment" on page OP-50.

Syntax:

\$CP	n
Required:	n
Default:	none

Operands Description

n The partition to which the terminal is to be assigned. (See "Listing System Terminal Assignments" on page OP-44 to determine the partition number.)

Reference Material

Operator Command Syntax (*continued*)

\$D - Dump Storage

Use the \$D operator command to display, or *dump*, the contents of an area of storage on the screen of your display terminal. When you enter the \$D command, EDX displays the hexadecimal contents of the specified storage locations.

\$D is used in the instructions for "Determining Version Level with \$D" on page OP-306.

Syntax:

\$D	origin,address,count
Required:	origin,address,count
Default:	none

Operands Description

origin	The hexadecimal origin address (the program load point). (See "Listing Partitions and Active Programs" on page OP-227 to determine this address.)
address	The hexadecimal address in the program at which the dump is to start.
count	The decimal number of words to dump (maximum value of 16).

\$E - Eject Printer Page

Use the \$E operator command to advance (eject) one or more pages on the system printer (\$SYSPRTR). Entering a number with \$E advances the paper that number of pages.

\$E is used in the instructions for "Advancing Printer Forms" on page OP-252.

Syntax:

\$E	n
Required:	none
Default:	ejects one page

Operands Description

n	The number of pages to eject
----------	------------------------------

Operator Command Syntax (*continued*)

\$L - Load a Program or Utility

Use the \$L operator command to load a program into storage and start it running. \$L is used in the instructions for “Using the \$L Operator Command” on page OP-217.

Syntax:

\$L	program,volume,storage data sets
Required:	program
Default:	volume defaults to IPL volume; storage defaults to the amount specified on the PROGRAM statement of the program to be loaded

<i>Operands</i>	<i>Description</i>
program	The name of the program being loaded. This is the same as the name of the data set where the program is stored.
volume.	The name of the volume containing the program data set. (See “List All Data Sets on All Volumes” on page OP-185 to determine this name.)
storage	The total additional storage (in bytes) to be added to the end of the loaded program (overrides the STORAGE= parameter specified in the PROGRAM statement). The number of bytes that must be specified, if any, is determined by local procedures.
data sets	The data set and volume names of one to nine data sets being passed to the program. This parameter is required if DS=?? is coded on the PROGRAM statement. They must be specified in the order used by the program and are entered in the format: <i>name,volume name,...</i> If data set names are required and you do not enter them as part of a single-line command, EDX prompts you for them.

Reference Material

Operator Command Syntax (*continued*)

\$P - Patch Storage

Use the \$P operator command to change (patch) one or more words of storage. (Refer to the *Problem Determination Guide* for instructions on using \$P.)

Note: The patch is only good for the current session. When the system is reinitialized (IPLed) or the executing program is reloaded, the patched data reverts to its original value.

Syntax:

\$P	origin,address,count
Required:	origin,address,count
Default:	none

Operands Description

origin	The hexadecimal origin address (program load point). (See "Listing Partitions and Active Programs" on page OP-227 to determine this address.)
address	The hexadecimal address in the program where the patch starts.
count	The decimal number of words being patched. A maximum of 16 words can be patched.

\$S - Control Spooled Program Output

Use the \$\$ operator command to control the operation of printer spooling from your display terminal. \$\$ has several subcommands that do these control functions. The syntax for these subcommands is described on the following pages under:

- "\$\$ ALT - Alter Spool Job Printing" on page OP-363
- "\$\$ DALL - Delete All Spool Jobs" on page OP-364
- "\$\$ DE - Delete a Spool Job" on page OP-365
- "\$\$ DG - Delete Generic Spool Jobs" on page OP-365
- "\$\$ DISP - Display Spool Status Information" on page OP-366
- "\$\$ HOLD - Hold Spool Job(s)" on page OP-366
- "\$\$ KEEP - Keep or Release a Spool Job" on page OP-367
- "\$\$ REL - Release Spool Jobs" on page OP-368
- "\$\$ STOP - Stop Spooling Facility" on page OP-368
- "\$\$ WRES - Restart a Spool Writer" on page OP-369
- "\$\$ WSTP - Stop a Spool Writer" on page OP-370
- "\$\$ WSTR - Start a Spool Writer" on page OP-371

Instructions for using \$\$ are found under "Spooling Program Output" on page OP-258.

Operator Command Syntax *(continued)*

\$\$ ALT - Alter Spool Job Printing

Use the \$\$ ALT command to change the parameters that control the way a spool job is printed. You can:

- Change the number of copies printed.
- Change the forms code for the job.
- Change the job name used on the spool job separator page.
- Direct one spool job to a different printer.
- Direct all spool jobs from one printer to another printer.
- Specify that forms alignment be verified before a job is printed.

\$\$ ALT is used in the instructions for “Altering Spool Job Printing Parameters” on page OP-276.

Syntax:

```
$$ ALT id [COPY n|FORM code|NAME heading|WRIT name|ALIGN Y/N]
or
$$ ALT WRIT cwriter nwriter

Required: id and either COPY FORM NAME WRIT or ALIGN
or
WRIT cwriter nwriter
Default: none
```

Operands Description

- | | |
|-------------|--|
| id | The one to three digit identification assigned to a spool job by the spool facility. This identification is included on the spool status report generated by the \$\$ DISP ALL operator command. (See “Displaying Spool Status Information” on page OP-275 for instructions for generating this report.) |
| COPY | n - The number of copies to be printed (must be from 1 to 127). |
| FORM | code - The four-character code identifying the forms required to print the spool job. |
| NAME | heading - A one to eight character heading printed on the spool job separator page. It defaults to the name of the program which created the spool job. |

Reference Material

Operator Command Syntax (*continued*)

WRIT The name of the spoolable printer or display terminal. The “WRIT name” form of this parameter is used to assign a printer or display terminal to a particular spool job. The “WRIT cwriter nwriter” form is used to direct spool jobs from one spool device to another.

name The name of the spool device for this spool job. (See “Specify Spool Devices” on page OP-271 to verify that a device can be used for spooling.)

cwriter The name of the current spool device whose spool jobs are to be redirected.

nwriter The name of the new spool device. (See “Specify Spool Devices” on page OP-271 to verify that a device can be used for spooling.)

ALIGN Y/N - Specifies whether forms alignment is to be verified before the spool job is printed (Y=yes, N=no). Alignment is verified for the next complete copy of the job. Alignment is not verified for a job that is printing when this command is entered, or for a job that has been stopped with the \$\$ WSTP command.

\$\$ DALL - Delete All Spool Jobs

Use \$\$ DALL to delete all ready or printing spool jobs. \$\$ DALL is used in the instructions for “Deleting Spool Jobs” on page OP-285.

Syntax:

\$\$ DALL

Required: None

Default: None

Operands Description

None None

Operator Command Syntax (*continued*)

\$\$ DE - Delete a Spool Job

Use \$\$ DE to delete one spool job that is either ready or printing. \$\$ DE is used in the instructions for “Deleting Spool Jobs” on page OP-285.

Syntax:

\$\$ DE id

Required: id

Default: None

Operands Description

id The one to three digit identification assigned to a spool job by the spool facility. This identification is included on the spool status report generated by the \$\$ DISP ALL operator command. (See “Displaying Spool Status Information” on page OP-275 for instructions for generating this report.)

\$\$ DG - Delete Generic Spool Jobs

Use the \$\$ DG command to delete all ready or printing spool jobs that have a name starting with a specified prefix. \$\$ DG is used in the instructions for “Deleting Spool Jobs” on page OP-285.

Syntax:

\$\$ DG string

Required: string

Default: None

Operands Description

string A one to eight character prefix that specifies the spool jobs to be deleted. All spool jobs with the prefix are deleted. (See “Displaying Spool Status Information” on page OP-275 to determine the names of the spool jobs.)

Reference Material

Operator Command Syntax (*continued*)

\$\$ DISP - Display Spool Status Information

Use \$\$ DISP to display information about spool jobs, spool resources, and spool writers. \$\$ DISP is used in the instructions for “Displaying Spool Status Information” on page OP-275.

Syntax:

```
$$ DISP id|ALL|STAT
```

Required: None

Default: ALL

<i>Operands</i>	<i>Description</i>
-----------------	--------------------

id	The internal one to three character identification assigned to a spool job by the spool facility. This identification is obtained by using the \$\$ DISP ALL command.
-----------	---

ALL	Displays the status of all spool jobs, all spool writers, and all spool resources.
------------	--

STAT	Displays the status of the spool resources.
-------------	---

\$\$ HOLD - Hold Spool Job(s)

Use the \$\$ HOLD command to hold a specific spool job, or all spool jobs, from being printed. Only active and ready spool jobs can be held. \$\$ HOLD is used in the instructions for “Holding Spool Jobs” on page OP-282.

Syntax:

```
$$ HOLD id|ALL
```

Required: None

Default: ALL

<i>Operands</i>	<i>Description</i>
-----------------	--------------------

id	The one to three character identification assigned to a spool job by the spool facility. This identification is included on the spool status report generated by the \$\$ DISP ALL operator command. (See “Displaying Spool Status Information” on page OP-275 for instructions for generating this report.)
-----------	--

ALL	Holds all active and ready spool jobs, and all future spool jobs.
------------	---

Operator Command Syntax (*continued*)

\$\$ KEEP - Keep or Release a Spool Job

Use \$\$ KEEP to keep a specific spool job from being deleted or to delete a job that has been kept. When a kept job is released, \$SPOOL prints one additional copy before deleting the job. \$\$ KEEP is used in the instructions for "Keeping and Releasing Kept Jobs" on page OP-284.

Syntax:

```
$$ KEEP      id Y|N
```

Required: id and either Y or N

Default: None

Operands Description

id	The one to three character identification assigned to a spool job by the spool facility. This identification is included on the spool status report generated by the \$\$ DISP ALL operator command. (See "Displaying Spool Status Information" on page OP-275 for instructions for generating this report.)
Y	Keeps the spool job available after it is printed. The spool job is both held and kept after printing, with the number of copies set to one. Thus, when released by using the \$\$REL command, it is printed (even if printed once already).
N	Deletes the spool job from the system after it is printed. When released by use of this operand, the number of copies of the spool job to be printed is set to one, even if more than one copy was requested before the job was kept.

Reference Material

Operator Command Syntax *(continued)*

\$\$ REL - Release Spool Jobs

Use \$\$ REL to release one, or all, held jobs for printing. A released job resumes its place in the ready queue; that is, its print order is still determined by the order in which it originally became ready. \$\$ REL is used in the instructions for "Releasing Held Jobs" on page OP-283.

Syntax:

```
$$ REL id|ALL
```

Required: None

Default: ALL

Operands Description

id The one to three character identification assigned to a spool job by the spool facility. This identification is included on the spool status report generated by the \$\$ DISP ALL operator command. (See "Displaying Spool Status Information" on page OP-275 for instructions for generating this report.)

ALL Releases all currently held spool jobs. This resets the effect of the \$\$ HOLD ALL command.

\$\$ STOP - Stop Spooling Facility

Use \$\$ STOP to stop the spooling facility. Spooling stops when any jobs in active or printing status finish.

\$\$ STOP is used in the instructions for "Stopping the Spooling Facility" on page OP-274.

Syntax:

```
$$ STOP
```

Required: None

Default: None

Operands Description

None None

Operator Command Syntax (*continued*)

\$\$ WRES - Restart a Spool Writer

Use \$\$ WRES to restart a temporarily stopped spool writer. You can restart a writer:

- At the beginning of the interrupted job.
- At the line following the last line printed.
- At a specified number of lines or pages before or after the last line printed.

\$\$ WRES is used in the instructions for “Restarting a Spool Writer” on page OP-280.

Syntax:

```
$$ WRES name [IMM|JOB|B nnn L/P|F nnn L/P] code
```

Required: name
 nnn and L or P if B or F specified

Default: IMM
 code defaults to blanks

Operands Description

name	The name of the writer which is to be restarted. (See “Displaying Spool Status Information” on page OP-275 to determine the writer name.)
IMM	Resume printing at the <i>next line</i> of the interrupted spool job.
JOB	Resume printing at the <i>start</i> of the interrupted spool job.
B	Resume printing <i>nnn</i> lines (L) or pages (P) <i>before</i> the line where the job was stopped.
F	Resume printing <i>nnn</i> lines (L) or pages (P) <i>after</i> the line where the job was stopped.
nnn	The number of lines or pages backward (B) or forward (F) from the point of interruption. Specify this parameter if you specified B or F. If the writer scrolls to the start of the spool job, the complete job is printed. If the writer scrolls to the end of the spool job, the job is not printed.
L	Specifies that <i>nnn</i> is in lines.
P	Specifies that <i>nnn</i> is in pages.

Reference Material

Operator Command Syntax (*continued*)

code The four-character forms code for the forms to be used for jobs printed by this writer. A new forms code specification takes effect when the writer processes the next job or the next copy of the current job. Defaults to blanks.

\$\$ WSTP - Stop a Spool Writer

Use \$\$ WSTP to stop a spool writer. You can stop a spool writer:

- Immediately at the start of the next line of the spool job or at the end of the current job.
- Temporarily, to be restarted with the \$\$ WRES command, or permanently. A spool job that is permanently stopped is restarted with the \$\$ WSTR command.

\$\$ WSTP is used in the instructions for “Stopping a Spool Writer” on page OP-279.

Syntax:

```
$$ WSTP name IMM|JOB TERM|NOTERM
```

Required: name

Default: IMM NOTERM

Operands Description

name The name of the writer which is being stopped. (See “Displaying Spool Status Information” on page OP-275 to determine the writer name.)

IMM Stop printing at the next line of the spool job.

JOB Stop printing at the end of the current job.

TERM Stop permanently. (The dedicated printer is released and writer task is ended.)

Note: A permanently stopped writer task can only be restarted by the \$\$ WSTR command.

NOTERM Stop temporarily. The writer can be restarted by the \$\$ WRES command. (The writer maintains control of the printer device.)

Operator Command Syntax (*continued*)

\$S WSTR - Start a Spool Writer

Use \$S WSTR to start a spool writer and specify a forms code.

\$S WSTR is used in the instructions for "Starting a Spool Writer" on page OP-278.

Syntax:

\$S WSTR	name code
Required:	name
Default:	code - defaults to blanks

Operands ***Description***

name The name of the printer for which the writer is to be started. This is also the name of the writer. (See "Displaying Spool Status Information" on page OP-275 to determine the printer name.)

code The four-character forms code for the printer forms used with the jobs printed by this writer. Defaults to blanks.

Reference Material

Operator Command Syntax (*continued*)

\$T - Set Date and Time

Use the \$T operator command to set the date and time in the EDX operating system.

You can only enter \$T from a display terminal named \$SYSLOG or \$SYSLOGA (the system logging display terminal or its alternate). If you enter it from a terminal with another name, the system displays the date and time as if you had entered a \$W command.

\$T is used in the instructions for "Setting the Time and Date" on page OP-26.

Notes:

1. \$T must be entered in prompt-reply format.
2. Make sure your time and date entry is correct as the system does not verify this data.

Syntax:

\$T	date,time
Required:	date,time
Default:	date defaults to 00/00/00 time defaults to 00:00:00

Operands *Description*

date	The current date. Can be entered as <i>mm/dd/yy</i> , <i>mm.dd.yy.</i> , or <i>mm dd yy</i> .
time	The current time. Can be entered as <i>hh:mm:ss</i> , <i>hh.mm</i> , or <i>hh mm</i> .

\$U - Invoke Your Own Operator Command

Use the \$U operator command to enter an operator command function that is unique to your system. Refer to *Customization Guide* for information on how to write a program for the \$U operator command.

If \$U is entered and your system does not have a program to support it, EDX displays the message "FUNCTION NOT DEFINED".

Operator Command Syntax (*continued*)

\$VARYOFF - Set Device Offline

Use the \$VARYOFF command to set a diskette or tape drive offline. When a diskette or tape drive is offline, it is not under computer control. When you vary a tape drive offline, the tape is rewound to the beginning.

If you remove a diskette from a 4966 diskette unit, you should vary the 4966 offline or EDX will think that the diskette is still available for use.

Note: To vary the 4966 offline, code the slot parameter.

See "Removing Diskette Magazines" on page OP-20 for instructions using the \$VARYOFF operator command.

Syntax:

```
$VARYOFF ioda slot
```

```
Required: ioda
```

```
Default: none
```

Operands Description

ioda	The hexadecimal device address of the diskette or tape device being varied offline. (See "Listing the Devices on Your System" on page OP-318 to determine the device address.)
slot	The number of the 4966 slot containing the diskette being varied offline. The valid slot numbers for the 4966 magazine unit are: 0 All diskettes (1,2,3,A,B) 1 Slot 1 2 Slot 2 3 Slot 3 A Magazine 1 B Magazine 2

Reference Material

Operator Command Syntax (*continued*)

\$VARYON - Set Device Online

Use the \$VARYON operator command to set a diskette or tape drive online. You must set a 4966 diskette unit or tape unit online so that EDX knows it is there.

Note: To vary the 4966 offline, code the slot parameter.

You do not have to enter \$VARYON when you put a new diskette into a diskette unit. EDX automatically varies it online when you shut the door of the diskette unit.

See "Varying the 4966 Diskette Magazine Unit Online" on page OP-18 for the instructions using the \$VARYON operator command.

Syntax:

\$VARYON	ioda slot file EX
Required:	ioda
Default:	file defaults to 1 maximum value of file is 255

Operands Description

ioda	The hexadecimal device address of the diskette or tape device being varied online. (See "Listing the Devices on Your System" on page OP-318 to determine the device address.)
slot	The number of the 4966 slot containing the diskette being varied online. The valid slot numbers for the 4966 magazine unit are: 0 All diskettes (1, 2, 3, A, B) 1 Slot 1 2 Slot 2 3 Slot 3 A Magazine 1 B Magazine 2
file	The decimal file number on the tape being accessed.
EX	Override the tape expiration date. If a tape data set is initialized with an expiration date, EX must be used to be able to write to that tape data set. The file number must also be specified.

Operator Command Syntax (*continued*)

\$W - Display Date and Time

Use the \$W operator command to display the date and time, according to your EDX system, on your display terminal.

See "Displaying the Date and Time" on page OP-27 for instructions using \$W.

Syntax:

\$W

Required: none

Default: none

Operands Description

None

Reference Material

\$JOBUTIL Command Syntax

This section lists the syntax for the \$JOBUTIL commands used in \$JOBUTIL procedures. These commands are used in the instructions for “Coding a \$JOBUTIL Procedure” on page OP-241.

The conventions described under “Syntax Conventions” on page OP-357 are also used for these commands.

AL — Allocate Data Set

Use the AL command to allocate a data set. AL returns a \$DISKUT3 return code that can be used by the JUMP command.

Note: Do not code an AL command between a PROGRAM and an EXEC command.

Syntax:

AL	name,volume size type
Required:	name,volume size
Default:	type - D

Operands Description

name	The name of the data set.
volume	The volume on which the data set is being allocated.
size	The number of 256 byte records in the data set
type	The data type of the data set as follows: D Specifies data (default) P Specifies program

Example:

This example allocates a data set named TEMPDS, on volume EDX003. TEMPDS has 25 256-byte records and has a type of *data*.

AL	TEMPDS,EDX003 25 D
----	--------------------

\$JOBUTIL Command Syntax (*continued*)

DE — Delete Data Set

Use the DE command to delete a data set.

Note: Do not code a DE command between a PROGRAM and an EXEC command.

Syntax:

```
DE          name,volume
```

```
Required:  name,volume
```

```
Default:   none
```

Operands *Description*

name The name of the data set being deleted.

volume The name of the volume which contains the data set. (See “Listing Volume and Data Set Directory Information” on page OP-181 to determine the data set and volume names.)

Example:

This example deletes a data set named TEMPDS from volume EDX003.

```
DE          TEMPDS ,EDX003
```

Reference Material

\$JOBUTIL Command Syntax (*continued*)

DS — Identify Data Set

Use the DS command to pass the name of a data set to the program specified on the previous PROGRAM command. The DS command is required if DS=?? is coded on the PROGRAM statement.

DS commands must be coded between PROGRAM and EXEC commands. Up to nine (9) DS commands can be specified for one program. They must be specified in the order used by the program.

Syntax:

DS	name,volume
Required:	name
Default:	volume is IPL volume

Operands *Description*

name The name of the data set.

volume The name of the volume which contains the data set. (See "Listing Volume and Data Set Directory Information" on page OP-181 to determine the data set and volume names.)

Example:

This example passes data set WORK1 on volume EDX001.

DS	WORK1 ,EDX001
----	---------------

\$JOBUTIL Command Syntax (*continued*)

EJECT — Start New Log Page

Use the EJECT command to print the next command in the log listing at the top of a new page. If no logging is being done, EJECT commands are ignored.

Syntax:

EJECT
Required: none
Default: none

Operands Description

None None

Example:

EJECT

EOJ — End of Job

Use the EOJ command to specify the end of the primary procedure. EOJ must be the last command in the procedure data set.

Syntax:

EOJ
Required: none

Operands Description

None

Example:

EOJ

Reference Material

\$JOBUTIL Command Syntax (*continued*)

EOP — End of Procedure

Use the EOP command to specify the end of a nested procedure. EOP must be the last command in the called procedure data set.

Syntax:

EOP
Required: none

Operands *Description*

None

Example:

EOP

EXEC — Execute Program

Use the EXEC command to load and execute the program identified in the preceding PROGRAM command. EXEC must be preceded by a PROGRAM command.

Syntax:

EXEC	STORAGE=size
Required:	none
Default:	STORAGE=0

Operands *Description*

STORAGE The number of bytes of additional storage needed by the program. This value overrides the value specified on the PROGRAM statement in the program. STORAGE=0 (the default) specifies that the dynamic storage specified on the PROGRAM statement (if any) should be allocated.

\$JOBUTIL Command Syntax (*continued*)

Example:

```
EXEC  
EXEC          STORAGE=256
```

JOB — Identify Job

Use the JOB command to identify the procedure. \$JOBUTIL prints the job identification on the terminal that loads the job. The date and time are also displayed if timer support is included in your system.

Syntax:

```
JOB          job-name  
Required:   job-name
```

Operands Description

job-name The job identification. It can be from 1 to 62 characters in length.

Example:

```
JOB          THIS PROCEDURE SETS UP AND RUNS PAYROLL
```

Reference Material

\$JOBUTIL Command Syntax (*continued*)

JUMP — Jump to Label

Use the JUMP command to skip one or more commands in a procedure. You can jump to a specified LABEL command unconditionally, or by testing a program completion code. If you test a completion code, \$JOBUTIL jumps to the specified LABEL command if the condition is true.

Jumps are forward only and are limited to the current procedure only. A jump cannot be made to a LABEL command in another procedure.

If you do not code the name of a LABEL command, \$JOBUTIL jumps to the next unlabeled LABEL command or to an EOJ or EOP command, whichever occurs first.

Note: The JUMP command cannot be entered in the PAUSE mode.

Syntax:

```
JUMP      label-name,op,cc
```

```
Required: none
```

Operands

Description

label-name

The name associated with a LABEL command

op

LT, GT, EQ, GE, LE, or NE.

cc

Program completion code.

Example:

The first JUMP in this example causes \$JOBUTIL to skip to the LABEL command LABEL1 if the completion code is equal to 20. The second skips to LABEL command LABEL1 without checking the completion code. The third command skips to the first unlabeled LABEL command, or an EOJ or EOP command, whichever occurs first.

```
JUMP      LABEL1 , EQ , 20
JUMP      LABEL1
JUMP
```

\$JOBUTIL Command Syntax (*continued*)

LABEL — Identify Continuation Point

Use the LABEL command to identify the next \$JOBUTIL command to be processed after a related JUMP command.

Note: The LABEL command cannot be entered in the PAUSE mode.

Syntax:

LABEL	label-name
Required: none	

<i>Operands</i>	<i>Description</i>
label-name	The unique name of this LABEL command.

Example:

LABEL	LABEL1
LABEL	

Reference Material

\$JOBUTIL Command Syntax (*continued*)

LOG — Log Control

Use the LOG command to specify whether the \$JOBUTIL commands are to be printed as they are read. You can also specify the terminal on which they are to be printed.

LOG commands can be placed anywhere in the procedure data set. If a terminal device is not specified, the commands are displayed on the display terminal from which \$JOBUTIL was started.

Syntax:

```
LOG      ON|OFF|terminal-name
```

Required: none

Default: ON

<i>Operands</i>	<i>Description</i>
ON	\$JOBUTIL commands are printed on the display terminal.
OFF	\$JOBUTIL commands are not to be printed.
terminal-name	The name of the display terminal or printer where the \$JOBUTIL commands are printed. (See "Listing System Terminal Assignments" on page OP-44 to determine the name of the display and printer terminals attached to your system.)

Example:

```
LOG      ON
LOG      OFF
LOG      $SYSPRTR
LOG
```

\$JOBUTIL Command Syntax *(continued)*

NOMSG — No Message Logging

Use the NOMSG command to specify that you do not want the “LOADING PROGRAM...” message for the program specified on the previous PROGRAM command.

The NOMSG command must be coded between a PROGRAM and an EXEC command.

Syntax:

NOMSG
Required: none

Operands Description

None

Example:

PROGRAM CALCDEMO
NOMSG

Reference Material

\$JOBUTIL Command Syntax (*continued*)

PARM — Pass Parameter

Use the PARM command to pass parameters to the program specified in the previous PROGRAM command. A maximum of 62 characters can be passed.

The parameters are passed as an EBCDIC character string at 2 characters per word for a maximum of 31 words. They can be referenced in the loaded program as beginning at label \$PARM1. The number of characters read by the program is determined by the PARM= operand on the PROGRAM statement in the program. For example, PARM=31 would cause the maximum 62 characters from positions 10-71 of the PARM command to be transferred to the specified program starting at the label \$PARM1. (Refer to the PROGRAM statement in the *Language Reference* for additional information on the PROGRAM statement.)

PARM must be coded between a PROGRAM and an EXEC command.

Syntax:

PARM	parameters
------	------------

Required:	none
-----------	------

Operands Description

parameters	1-62 characters being passed to the program.
------------	--

Example:

This example passes the parameter string "NOLIST,XREF".

PARM	NOLIST,XREF
------	-------------

\$JOBUTIL Command Syntax (*continued*)

PAUSE — Await Manual Intervention

Use the PAUSE command to allow \$JOBUTIL commands to be entered from a terminal keyboard. (All \$JOBUTIL commands except JUMP and LABEL can be entered.)

When \$JOBUTIL reads a PAUSE command, it prompts you for input. You communicate with \$JOBUTIL by pressing the attention key and entering one of the following three PAUSE subcommands.

- GO - Ends the PAUSE mode. \$JOBUTIL continues with the next command in the procedure data set.
- ENTER - Allows you to enter \$JOBUTIL commands from the display terminal.
- ABORT - Ends the current \$JOBUTIL procedure.

Syntax:

PAUSE

Required: none

Operands Description

None

Examples:

The following is an example of a procedure that uses the PAUSE command and the display terminal screen during PAUSE mode:

The procedure with PAUSE:

```
JOB          PAYROLL
LOG          ON
PROGRAM      WAGES
EXEC
REMARK       LAST DAY OF MONTH?
REMARK       RUN MONTHEND
PAUSE
EOJ
```

Reference Material

\$JOBUTIL Command Syntax (*continued*)

Display terminal screen during pause mode:

```
PAUSE - * - ATTN: GO/ENTER/ABORT

PAUSE
>ENTER

ENTER COMMAND PROGRAM
ENTER OPERAND MONTHEND
ENTER COMMAND EXEC
```

PROC — Execute Procedure

Use the PROC command to pass control to another procedure (nested procedure). PROC cannot be placed between a PROGRAM and an EXEC command. The completion code of the last program executed in the nested procedure is available for testing by a JUMP command in the main procedure.

You cannot pass control to another procedure from a nested procedure.

Syntax:

```
PROC      procedure-name,volume

Required: procedure-name
Default:  volume - to IPL volume
```

Operands	Description
procedure-name	The name of the procedure data set to which control is passed.
volume	The volume containing the procedure data set. (See "Listing Volume and Data Set Directory Information" on page OP-181 to determine the volume name.)

Example:

This example calls a nested procedure named PAYROLL on volume EDX002 and a procedure PAYCHKS on the IPL volume.

```
PROC      PAYROLL,EDX002
PROC      PAYCHKS
```

\$JOBUTIL Command Syntax (*continued*)

PROGRAM — Identify Program

Use the PROGRAM command to identify a program to be executed (loaded).

Syntax:

```
PROGRAM program-name,volume
```

```
Required: program-name
```

```
Default: volume is IPL volume
```

Operands

Description

program-name The name of the data set containing the program to be run. The program name and the data set name are the same.

volume The name of the volume containing the data set. (See “Listing Volume and Data Set Directory Information” on page OP-181 to determine the data set and volume names.)

Example:

This example identifies a program named \$DEBUG on volume EDX002, and a program named CHECKS on the IPL volume. (Two PROGRAM commands must be separated by an EXEC command.)

```
PROGRAM $DEBUG,EDX002
EXEC
PROGRAM CHECKS
```

Reference Material

\$JOBUTIL Command Syntax (*continued*)

REMARK — Display Remark

Use the REMARK command to display a message to the operator. The message is displayed whether LOG is ON or OFF. The message can be a maximum of 62 characters long.

Syntax:

```
REMARK    comment
```

```
Required: none
```

Operands **Description**

comment 1 to 62 character message.

Example:

```
REMARK    INSERT DISKETTE EDX005
```

* — Comment

Use an asterisk (*) in column 1 to specify internal comments in the procedure data set. Comments can start in column 2. The internal comment is not printed.

Syntax:

```
*internal comment
```

```
Required: none
```

Operands **Description**

None

Example:

```
* THIS PROCEDURE IS A TEST CASE
```

Spool Status Display Formats

The following sections explain the formats of the status displays generated by the \$SPLUT1 utility and the \$\$ DISP command. (See “Displaying Spool Status Information” on page OP-275 for instructions on generating the \$\$ DISP reports.)

\$SPLUT1 Status Display

\$SPLUT1 displays the following status information when loaded and any time it is used to change one of the spool parameters:

```
*****
**                                     **
** CHANGES EFFECTIVE THE NEXT TIME $SPOOL IS LOADED **
**                                     **
*****
 1   2   3   4   5   6   7   8   9   10  11
DSNAME-VOLUME-MAXJOBS-MAXACTV-RESTRT-GRPS-GRPSZ-SEP-DEVICES-AUTO-FORM
.
.
.
```

- 1** DSNAME - The name of the spool data set. (See “Specify the Spool Data Set” on page OP-264 to change the name of the data set.)
- 2** VOLUME - The name of the volume containing the spool data set. (See “Specify the Spool Data Set” on page OP-264 to change the name of the volume.)
- 3** MAXJOBS (maximum jobs) - the maximum number of spool jobs that can be in the system at one time. (See “Set the Maximum Jobs Option” on page OP-266 to change this number.)
- 4** MAXACTV (maximum active) - The maximum number of spool jobs that can be active at one time. (See “Set the Maximum Active Jobs Option” on page OP-267 to change this number.)
- 5** RESTRT (restart) - A Y means that restart mode is in effect. An N means that restart mode is not in effect. (See “Set the Restart Option” on page OP-262 to change the restart mode.)
- 6** GRPS (groups) - The number of spool groups that can be contained in the spool data set. (See “Set Spool Groups and Group Size Options” on page OP-268 to change this number.)
- 7** GRPSZ (group size) - The number of records in one spool group. (See “Set Spool Groups and Group Size Options” on page OP-268 to change this number.)
- 8** SEP (separator) - A Y means that a separator page is printed between spool jobs. An N means that a separator page is not printed. (See “Set the Separator Page Option” on page OP-270 to change this option.)

Reference Material

Spool Status Display Formats *(continued)*

- 9 DEVICES - The system names of the devices that are specified as being spool devices. This is also the name of the spool writer for the device. (See "Specify Spool Devices" on page OP-271 to change these devices.)
- 10 AUTO (auto start) - A Y means that the writer for this device starts automatically when spool (\$SPOOL) is loaded. (See "Specify Spool Devices" on page OP-271 to change this parameter.)
- 11 FORMS - The forms code for special forms that must be used for jobs printed by this writer. (This code is set by the \$\$ ALT FORM, \$\$ WRES, or \$\$ WSTR commands, the printed spool control record. See "Altering Spool Job Printing Parameters" on page OP-276 to change the forms code.)

Spool Job Display Format

A \$\$ DISP JOBID command displays the following information about a spool job.

1	2	3	4	5	6	7	8	9	10	11	12	13	14					
JOB	---	NAME	---	TERMINAL	--	WRITER	--	FORM	ACT	RDY	HLD	PTG	INC	KEP	ALN	LINE	--	CPY
.																		
.																		

- 1 JOB - The spool job identification assigned by the spool facility.
- 2 NAME - An optional separator page heading supplied by the \$\$ ALT NAME command or the printed spool control record. It defaults to the name of the program that generated the output. (See "Altering Spool Job Printing Parameters" on page OP-276 to change this name.)
- 3 TERMINAL - The name of the terminal in use when \$SPLUT1 was loaded. This name appears on the separator page with the heading. (See "Specify Spool Devices" on page OP-271 to change the terminal name.)
- 4 WRITER - The name of the spool writer. This name is the same as the name of the spool output device. (See "Altering Spool Job Printing Parameters" on page OP-276 or "Specify Spool Devices" on page OP-271 to change the writer name.)
- 5 FORM - The forms code for special forms that must be used for this job. (This code is set by the \$\$ ALT FORM, \$\$ WRES, \$\$ WSTR commands, or the printed spool control record. See "Altering Spool Job Printing Parameters" on page OP-276 to change the forms code.)
- 6 ACT (active) - An A in this column means that the spool job is being created by the application program.
- 7 RDY (ready) - An R in this column means that the spool job has been created and is ready for printing by the spool writer.

Spool Status Display Formats *(continued)*

- 8 HLD (held) - An H in this column means that the spool job is being held by the operator and is not available for printing. (See “Releasing Held Jobs” on page OP-283 to release the job.)
- 9 PTG (printing) - A P in this column means that a spool writer is currently printing the job. (See “Holding Spool Jobs” on page OP-282 to stop the job.)
- 10 INC (incomplete) - An I in this column means that the spool job is incomplete because of abnormal system termination.
- 11 KEP (keep) - A K in this column means that the spool job is not to be deleted after it is printed but is to be kept until specifically deleted by the operator. (See “Keeping and Releasing Kept Jobs” on page OP-284 to delete the job.)
- 12 ALN (alignment) - An A in this column means that the forms alignment option applies to this job. (See “Altering Spool Job Printing Parameters” on page OP-276 to change this option.)
- 13 LINE - The number of lines in the spool job. This number may differ from the actual number of print lines in the job because it includes all I/O instructions directed to the printer (ENQT, DEQT, TERMCTL, etc.).
- 14 CPY (copy) - The number of copies to be printed. (See “Altering Spool Job Printing Parameters” on page OP-276 to change the number of copies.)

Spool Writer Display Format

A `$$ DISP ALL` command displays the following information about all spool writers, jobs, and resources:

```
1 2 3 4 5 6 7 8 9 10
WRITER----JOBID--FORMS--WTG-STP-JOB-INA-INV-LINE--PAGE
.
.
.
```

- 1 WRITER - The name of the spool writer. This is also the spool device name. (See “Altering Spool Job Printing Parameters” on page OP-276 or “Specify Spool Devices” on page OP-271 to change the writer name.)
- 2 JOBID - The spool job identification assigned by the spool facility; if not blank, the writer is processing the job.
- 3 FORMS - The forms code for special forms that must be used for jobs printed by this writer. (This code is set by the `$$ ALT FORM`, `$$ WRES`, `$$ WSTR` commands, or the printed spool control record. See “Altering Spool Job Printing Parameters” on page OP-276 to change this code.)

Reference Material

Spool Status Display Formats (*continued*)

- 4 WTG (waiting) - A w in this column means that the writer is waiting because there are no ready spool jobs.
- 5 STP (stopped) - An s in this column means that the writer is temporarily stopped. (See “Restarting a Spool Writer” on page OP-280 to restart this writer.)
- 6 JOB - A J in this column means that the writer is stopping or restarting at end or start of the job. (See “Stopping a Spool Writer” on page OP-279 or “Restarting a Spool Writer” on page OP-280 to change this parameter.)
- 7 INA (inactive) - An N in this column means that the writer is either ended or not started (See “Starting a Spool Writer” on page OP-278 to start this writer.)
- 8 INV (invalid) - A v in this column means that the device corresponding to the writer name is not a spool device or that the writer is not in a status indicated by numbers 4, 5, 6, or 7.
- 9 LINE - The number of print commands processed by the spool writer. This is the number of print lines processed, not the total number of lines in the spool job.
- 10 PAGE - The number of the page currently being printed by the writer.

Spool Resources Display Format

A \$S DISP ALL OR STAT command displays the following information about the spool resources:

```
JOBS: 1 a(b) ACTV: 2 a(b) GRPS: 3 a(b) DS=SP00L 4 ,EDX003
```

The first number following each category (*a*) is the current value. The number in parentheses (*b*) is the maximum value.

- 1 JOBS - The maximum number of spool jobs that can be in the system. (See “Set the Maximum Jobs Option” on page OP-266 to change this number.)
- 2 ACTV (active) - The maximum number of spool jobs that can be active. (See “Set the Maximum Active Jobs Option” on page OP-267 to change this number.)
- 3 GRPS (groups) - The number of data set groups allocated to all spool jobs. (See “Set Spool Groups and Group Size Options” on page OP-268 to change this number.)
- 4 DS= (data set) - The data set and volume names of the spool data set being used. (See “Specify the Spool Data Set” on page OP-264 to change the data set and volume name.)

Estimating Spool Job Print Time

The print time represents the interval between the time the job starts printing and the time the job is finished printing. To calculate the print time for any spool job, use the following formula:

$$L/S = \text{time in minutes (add 10 for overhead)}$$

L is the number of lines and S is the printer speed in lines per minute.

Use the procedure “Displaying Spool Status Information” on page OP-275 to get the number of lines in a spool job. Refer to the description manual for your printer for the printer speed.

Estimating Spool Data Set Size

You can start with a spool data set size of 8510 records and decrease or increase the size according to need. Use the \$S DISP STAT operator command to monitor space usage and the \$S DISP JOBID command to determine the number of lines in the current spool jobs. (See “Displaying Spool Status Information” on page OP-275 for instructions on using \$S DISP.)

Some of the factors which affect the amount of space needed are:

- The size of the largest completed job.
- The number of concurrent programs which spool output.
- The amount of data printed by those programs.
- Whether spool jobs are held or kept and for how long.
- The number and speed of your printers.
- The group size used.

The following formulas help estimate the size needed if you want a more precise allocation of the spool data set.

Estimating \$EDXASM and \$S1ASM Output Listing Sizes

To calculate space requirements for the output listings of the \$EDXASM compiler and \$S1ASM assembler, use the following formula: (*n* is the number of print lines expected.)

$$(n * 80) / 256 = \text{number of records needed}$$

Reference Material

Estimating Spool Data Set Size *(continued)*

Estimating \$EDXLINK Output Listing Size

To calculate space requirements for the output listing of the linkage editor (\$EDXLINK), use the following formula: (n is the number of print lines expected.)

$$(n * 65) / 256 = \text{number of records needed}$$

Estimating Application Program Output Listing Size

Spooled data is stored in a compressed format; this must be considered in estimating space requirements. For example, multiple repetitions of the same character (such as blanks) within a print line occupy only two bytes. (Refer to the \$PACK/\$UNPACK description in the *Communications Guide* for a description of the compression algorithm).

To calculate space requirements for the compiler output listing of any application program, use the following formula: (n is the number of print lines expected and l is the average length of compressed spooled lines.)

$$(n * l) / 256 = \text{number of records needed}$$

Estimating COBOL, FORTRAN, and PLI Output Listing Sizes

To calculate space requirements for the output listings of the COBOL, FORTRAN and PLI compilers, use the following formula: (n is the number of print lines expected)

$$(N * 60) / 256 = \text{number of records needed}$$

Appendix B. Blank Forms

Blank copies of the following forms are supplied for you convenience:

- “\$JOBUTIL Procedure Planning Guide” on page OP-399
- “Backup Log” on page OP-407
- “Diskette Log” on page OP-411
- “Job Instructions” on page OP-415
- “Operations Log” on page OP-419
- “Problem Recording Form” on page OP-423
- “\$JOBUTIL Procedures Log” on page OP-403
- “Tape Log” on page OP-431

Blank Forms



\$JOBUTIL Procedure Planning Guide

FOR THE PRIMARY PROCEDURE:

Job identification (JOB): _____

Stored in: _____, _____ on disk _____ diskette # _____

Command logging: (LOG) ON _____ OFF _____ Terminal name _____

Name and volume of nested procedures called by this procedure:

_____, _____, _____, _____, _____

FOR EACH PROGRAM:

Name of the program to be run (PROGRAM & EXEC): _____

Additional storage required by the program: EXEC STORAGE= _____

Turn off the log message for this program (NOMSG): Yes _____ No _____

Data sets that must be allocated before the program can be run (AL):

name	,volume	size	type	name	,volume	size	type
------	---------	------	------	------	---------	------	------

_____, _____, _____, _____, _____, _____, _____, _____

_____, _____, _____, _____, _____, _____, _____, _____

JUMP to LABEL _____ if allocate is not equal (NE) to -1

Data sets that must be deleted before the program can be run (DE):

_____, _____, _____, _____, _____, _____, _____, _____

Data sets that must be deleted after the program is run (DE):

_____, _____, _____, _____, _____, _____, _____, _____

Data sets whose names must be passed when program is loaded:

_____, _____, _____, _____, _____, _____, _____, _____

_____, _____, _____, _____, _____, _____, _____, _____

JUMP to LABEL _____ if condition code from _____ is: _____

LT _____ GT _____ EQ _____ GE _____ LE _____ NE _____

Parameters required by the program (PARM): _____

Operator intervention required before/after the program

is run (PAUSE): Yes _____ No _____

If yes, commands to be entered:

_____, _____, _____, _____, _____, _____, _____, _____

Special processing:

_____, _____, _____, _____, _____, _____, _____, _____

_____, _____, _____, _____, _____, _____, _____, _____

_____, _____, _____, _____, _____, _____, _____, _____

_____, _____, _____, _____, _____, _____, _____, _____

IBM Series/1 Event Driven Executive

\$JOBUTIL Procedure Planning Guide

FOR THE PRIMARY PROCEDURE:

Job identification (JOB): _____

Stored in: _____, _____ on disk _____ diskette # _____

Command logging: (LOG) ON _____ OFF _____ Terminal name _____

Name and volume of nested procedures called by this procedure:

FOR EACH PROGRAM:

Name of the program to be run (PROGRAM & EXEC): _____, _____

Additional storage required by the program: EXEC STORAGE= _____

Turn off the log message for this program (NOMSG): Yes _____ No _____

Data sets that must be allocated before the program can be run (AL):

name	,volume	size	type	name	,volume	size	type
------	---------	------	------	------	---------	------	------

_____	_____	_____	_____	_____	_____	_____	_____
_____	_____	_____	_____	_____	_____	_____	_____

JUMP to LABEL _____ if allocate is not equal (NE) to -1

Data sets that must be deleted before the program can be run (DE):

Data sets that must be deleted after the program is run (DE):

Data sets whose names must be passed when program is loaded:

JUMP to LABEL _____ if condition code from _____ is: _____

LT _____ GT _____ EQ _____ GE _____ LE _____ NE _____

Parameters required by the program (PARM): _____

Operator intervention required before/after the program

is run (PAUSE): Yes _____ No _____

If yes, commands to be entered:

\$JOBUTIL Procedure Planning Guide

FOR THE PRIMARY PROCEDURE:

Job identification (JOB): _____

Stored in: _____, _____ on disk _____ diskette # _____

Command logging: (LOG) ON _____ OFF _____ Terminal name _____

Name and volume of nested procedures called by this procedure:

FOR EACH PROGRAM:

Name of the program to be run (PROGRAM & EXEC): _____

Additional storage required by the program: EXEC STORAGE= _____

Turn off the log message for this program (NOMSG): Yes _____ No _____

Data sets that must be allocated before the program can be run (AL):

name	,volume	size	type	name	,volume	size	type
------	---------	------	------	------	---------	------	------

_____	,	_____	_____	_____	,	_____	_____
_____	,	_____	_____	_____	,	_____	_____

JUMP to LABEL _____ if allocate is not equal (NE) to -1

Data sets that must be deleted before the program can be run (DE):

Data sets that must be deleted after the program is run (DE):

Data sets whose names must be passed when program is loaded:

JUMP to LABEL _____ if condition code from _____ is: _____

LT _____ GT _____ EQ _____ GE _____ LE _____ NE _____

Parameters required by the program (PARM): _____

Operator intervention required before/after the program

is run (PAUSE): Yes _____ No _____

If yes, commands to be entered:

Special processing:

\$JOBUTIL Procedure Planning Guide

FOR THE PRIMARY PROCEDURE:

Job identification (JOB): _____

Stored in: _____, _____ on disk _____ diskette # _____

Command logging: (LOG) ON _____ OFF _____ Terminal name _____

Name and volume of nested procedures called by this procedure:
_____, _____, _____, _____, _____

FOR EACH PROGRAM:

Name of the program to be run (PROGRAM & EXEC): _____, _____

Additional storage required by the program: EXEC STORAGE= _____

Turn off the log message for this program (NOMSG): Yes _____ No _____

Data sets that must be allocated before the program can be run (AL):

name	,volume	size	type	name	,volume	size	type
------	---------	------	------	------	---------	------	------

_____	_____	_____	_____	_____	_____	_____	_____
_____	_____	_____	_____	_____	_____	_____	_____

JUMP to LABEL _____ if allocate is not equal (NE) to -1

Data sets that must be deleted before the program can be run (DE):

Data sets that must be deleted after the program is run (DE):

Data sets whose names must be passed when program is loaded:

JUMP to LABEL _____ if condition code from _____ is: _____

LT _____ GT _____ EQ _____ GE _____ LE _____ NE _____

Parameters required by the program (PARM): _____

Operator intervention required before/after the program

is run (PAUSE): Yes _____ No _____

If yes, commands to be entered:

Special processing:

IBM Series/1 Event Driven Executive

Job Instructions

Job name: _____

Program/procedure name: _____ On volume: _____

Stored on: Disk _____ Diskette # _____

Other data sets required:

Data set name	Volume name	Diskette #	Tape #
_____	_____	_____	_____
_____	_____	_____	_____
_____	_____	_____	_____

Input data required: (Attached Y/N Obtain from)

Jobs that must be run first:

Prompts that can be expected:

Special printer forms:

Notes/Special Instructions:

Date run: _____ Time: _____ Operator: _____

Comments:

Job Instructions

Job name: _____

Program/procedure name: _____ On volume: _____

Stored on: Disk _____ Diskette # _____

Other data sets required:

Data set name	Volume name	Diskette #	Tape #
_____	_____	_____	_____
_____	_____	_____	_____
_____	_____	_____	_____

Input data required: (Attached Y/N Obtain from)

Jobs that must be run first:

Prompts that can be expected:

Special printer forms:

Notes/Special Instructions:

Date run: _____ Time: _____ Operator: _____

Comments:

IBM Series/1 Event Driven Executive

Job Instructions

Job name: _____

Program/procedure name: _____ On volume: _____

Stored on: Disk _____ Diskette # _____

Other data sets required:

Data set name	Volume name	Diskette #	Tape #
_____	_____	_____	_____
_____	_____	_____	_____
_____	_____	_____	_____

Input data required: (Attached Y/N Obtain from)

Jobs that must be run first:

Prompts that can be expected:

Special printer forms:

Notes/Special Instructions:

Date run: _____ Time: _____ Operator: _____

Comments:

IBM Series/1 Event Driven Executive

Job Instructions

Job name: _____

Program/procedure name: _____ On volume: _____

Stored on: Disk _____ Diskette # _____

Other data sets required:

Data set name	Volume name	Diskette #	Tape #
_____	_____	_____	_____
_____	_____	_____	_____
_____	_____	_____	_____

Input data required: (Attached Y/N Obtain from)

Jobs that must be run first:

Prompts that can be expected:

Special printer forms:

Notes/Special Instructions:

Date run: _____ Time: _____ Operator: _____

Comments:

IBM Series/1 Event Driven Executive

Problem Recording Form

Date _____ Time _____ the problem occurred.

Program name _____ has been running for _____.

Record the console switch positions and mark any indicators that are on:

Load <input type="checkbox"/>	Wait <input type="checkbox"/>	Run <input type="checkbox"/>	IPL Source Primary Alternate Auto IPL Normal Diagnostic	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Check <input type="checkbox"/>	Reset ***	Store		Data Buffer	Console Interrupt ***	Start	Stop <input type="checkbox"/>													
Level 0 <input type="checkbox"/>	Level 1 <input type="checkbox"/>	Level 2 <input type="checkbox"/>		Level 3 <input type="checkbox"/>	Stop On Address <input type="checkbox"/>	Instruct Step <input type="checkbox"/>	Check Restart <input type="checkbox"/>	Stop On Error <input type="checkbox"/>												

List unexpected results that occurred:

List expected results before the problem occurred:

Was any display, or printed output being generated just before the problem? Yes _____ No _____

If display or printed output was being generated, is it correct? Yes _____ No _____ If no, specify what is wrong:

Attach any output from the program such as printer output, magnetic tape or diskette output. attached: Yes _____ No _____

List, or attach a copy of, any system messages.

List the I/O devices that were in operation:

Note any unusual conditions of I/O devices, such as whether any indicators are on that should be off or off that should be on.

Take a stand-alone dump.

Dump is on diskette number _____

Dump diskette is attached: Yes _____ No _____

If not attached, it is _____

Problem Recording Form (continued)

If the system is in a loop and you have a programmer console:



1. Press the Stop key



2. Press the Instruct Step key to place the processor in instruction-step mode.



3. Press the Start key and record the indicators that are on. Each time the Start key is pressed, one instruction is executed and the indicators show a new address in the instruction loop.

4. Repeat pressing the Start key and recording the indicators until the first pattern of indicators reappears. This may be as few as 2 patterns, or it may be many. This information will be of help in analyzing the loop.

1.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○

2.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○

3.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○

4.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○

5.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○

6.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○

7.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○

8.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○

9.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○

10.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○

11.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○

12.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○

Additional notes:

IBM Series/1 Event Driven Executive

Problem Recording Form

Date _____ Time _____ the problem occurred.

Program name _____ has been running for _____.

Record the console switch positions and mark any indicators that are on:

<table border="1"> <tr> <td>Load</td> <td>Wait</td> <td>Run</td> </tr> <tr> <td><input type="checkbox"/></td> <td><input type="checkbox"/></td> <td><input type="checkbox"/></td> </tr> </table>			Load	Wait	Run	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<p>IPL Source</p> <p>Primary <input type="checkbox"/></p> <p>Alternate <input type="checkbox"/></p> <p>Auto IPL <input type="checkbox"/></p> <p>Normal <input type="checkbox"/></p> <p>Diagnostic <input type="checkbox"/></p>			<table border="1"> <tr> <td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td> </tr> <tr> <td><input type="checkbox"/></td><td><input type="checkbox"/></td> </tr> </table>																0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	<input type="checkbox"/>															
Load	Wait	Run																																																									
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>																																																									
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15																																												
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>																																												
<table border="1"> <tr> <td>Check</td> <td>Reset</td> </tr> <tr> <td><input type="checkbox"/></td> <td>...</td> </tr> </table>		Check	Reset	<input type="checkbox"/>	...	<table border="1"> <tr> <td>Store</td> <td>Data Buffer</td> </tr> <tr> <td><input type="checkbox"/></td> <td><input type="checkbox"/></td> </tr> </table>		Store	Data Buffer	<input type="checkbox"/>	<input type="checkbox"/>	<table border="1"> <tr> <td>Console Interrupt</td> <td>Start</td> </tr> <tr> <td>...</td> <td><input type="checkbox"/></td> </tr> </table>		Console Interrupt	Start	...	<input type="checkbox"/>	<table border="1"> <tr> <td>Stop</td> <td></td> </tr> <tr> <td><input type="checkbox"/></td> <td></td> </tr> </table>		Stop		<input type="checkbox"/>																																					
Check	Reset																																																										
<input type="checkbox"/>	...																																																										
Store	Data Buffer																																																										
<input type="checkbox"/>	<input type="checkbox"/>																																																										
Console Interrupt	Start																																																										
...	<input type="checkbox"/>																																																										
Stop																																																											
<input type="checkbox"/>																																																											
<table border="1"> <tr> <td>Level 0</td> <td>Level 1</td> </tr> <tr> <td><input type="checkbox"/></td> <td><input type="checkbox"/></td> </tr> </table>		Level 0	Level 1	<input type="checkbox"/>	<input type="checkbox"/>	<table border="1"> <tr> <td>Level 2</td> <td>Level 3</td> </tr> <tr> <td><input type="checkbox"/></td> <td><input type="checkbox"/></td> </tr> </table>		Level 2	Level 3	<input type="checkbox"/>	<input type="checkbox"/>	<table border="1"> <tr> <td>Stop On Address</td> <td>Instruct Step</td> </tr> <tr> <td><input type="checkbox"/></td> <td><input type="checkbox"/></td> </tr> </table>		Stop On Address	Instruct Step	<input type="checkbox"/>	<input type="checkbox"/>	<table border="1"> <tr> <td>Check Restart</td> <td>Stop On Error</td> </tr> <tr> <td><input type="checkbox"/></td> <td><input type="checkbox"/></td> </tr> </table>		Check Restart	Stop On Error	<input type="checkbox"/>	<input type="checkbox"/>																																				
Level 0	Level 1																																																										
<input type="checkbox"/>	<input type="checkbox"/>																																																										
Level 2	Level 3																																																										
<input type="checkbox"/>	<input type="checkbox"/>																																																										
Stop On Address	Instruct Step																																																										
<input type="checkbox"/>	<input type="checkbox"/>																																																										
Check Restart	Stop On Error																																																										
<input type="checkbox"/>	<input type="checkbox"/>																																																										

List unexpected results that occurred:

List expected results before the problem occurred:

Was any display, or printed output being generated just before the problem? Yes _____ No _____

If display or printed output was being generated, is it correct? Yes _____ No ____ If no, specify what is wrong:

Attach any output from the program such as printer output, magnetic tape or diskette output. attached: Yes ____ No ____

List, or attach a copy of, any system messages.

List the I/O devices that were in operation:

Note any unusual conditions of I/O devices, such as whether any indicators are on that should be off or off that should be on.

Take a stand-alone dump.

Dump is on diskette number _____

Dump diskette is attached: Yes _____ No _____

If not attached, it is _____

IBM Series/1 Event Driven Executive

Problem Recording Form

Date _____ Time _____ the problem occurred.

Program name _____ has been running for _____.

Record the console switch positions and mark any indicators that are on:

Load	Wait	Run	IPL Source		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Primary	<input type="checkbox"/>																	
			Alternate	<input type="checkbox"/>	Check		Reset		Store		Data Buffer		Console Interrupt		Start		Stop				
			Auto IPL	<input type="checkbox"/>																	
			Normal	<input type="checkbox"/>	Level 0	Level 1	Level 2	Level 3	Stop On Address	Instruct Step	Check Restart	Stop On Error									
			Diagnostic	<input type="checkbox"/>																	

List unexpected results that occurred:

List expected results before the problem occurred:

Was any display, or printed output being generated just before the problem? Yes _____ No _____

If display or printed output was being generated, is it correct? Yes _____ No ____ If no, specify what is wrong:

Attach any output from the program such as printer output, magnetic tape or diskette output. attached: Yes ____ No ____

List, or attach a copy of, any system messages.

List the I/O devices that were in operation:

Note any unusual conditions of I/O devices, such as whether any indicators are on that should be off or off that should be on.

Take a stand-alone dump.

Dump is on diskette number _____

Dump diskette is attached: Yes _____ No _____

If not attached, it is _____

Problem Recording Form

Date _____ Time _____ the problem occurred.

Program name _____ has been running for _____.

Record the console switch positions and mark any indicators that are on:

Load	Wait	Run	IPL Source		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Primary	<input type="checkbox"/>																
			Alternate	<input type="checkbox"/>	Check	Reset		Store	Data Buffer	Console Interrupt	Start	Stop								
			Auto IPL	<input type="checkbox"/>	<input type="checkbox"/>										
			Normal	<input type="checkbox"/>	Level 0	Level 1	Level 2	Level 3	Stop On Address	Instruct Steu	Check Restart	Stop On Error								
			Diagnostic	<input type="checkbox"/>																

List unexpected results that occurred:

List expected results before the problem occurred:

Was any display, or printed output being generated just before the problem? Yes _____ No _____

If display or printed output was being generated, is it correct? Yes _____ No _____ If no, specify what is wrong:

Attach any output from the program such as printer output, magnetic tape or diskette output. attached: Yes _____ No _____

List, or attach a copy of, any system messages.

List the I/O devices that were in operation:

Note any unusual conditions of I/O devices, such as whether any indicators are on that should be off or off that should be on.

Take a stand-alone dump.

Dump is on diskette number _____

Dump diskette is attached: Yes _____ No _____

If not attached, it is _____

Glossary of Terms and Abbreviations

This glossary defines terms and abbreviations used in the Series/1 Event Driven Executive software publications. All software and hardware terms pertain to EDX. This glossary also serves as a supplement to the *IBM Data Processing Glossary*, GC20-1699.

\$\$SYSLOGA, \$\$SYSLOGB. The name of the alternate system logging device. This device is optional but, if defined, should be a terminal with keyboard capability, not just a printer.

\$\$SYSLOG. The name of the system logging device or operator station; must be defined for every system. It should be a terminal with keyboard capability, not just a printer.

\$\$SYSPRTR. The name of the system printer.

abend. Abnormal end-of-task. Termination of a task prior to its completion because of an error condition that cannot be resolved by recovery facilities while the task is executing.

ACCA. See asynchronous communications control adapter.

address key. Identifies a set of Series/1 segmentation registers and represents an address space. It is one less than the partition number.

address space. The logical storage identified by an address key. An address space is the storage for a partition.

application program manager. The component of the Multiple Terminal Manager that provides the program management facilities required to process user requests. It controls the contents of a program area and the execution of programs within the area.

application program stub. A collection of subroutines that are appended to a program by the linkage editor to provide the link from the application program to the Multiple Terminal Manager facilities.

asynchronous communications control adapter. An ASCII terminal attached via #1610, #2091 with #2092, or #2095 with #2096 adapters.

attention key. The key on the display terminal keyboard that, if pressed, tells the operating system that you are entering a command.

attention list. A series of pairs of 1 to 8 byte EBCDIC strings and addresses pointing to EDL instructions. When the attention key is pressed on the terminal, the operator can enter one of the strings to cause the associated EDL instructions to be executed.

backup. A copy of data to be used in the event the original data is lost or damaged.

base record slots. Space in an indexed file that is reserved for based records to be placed.

base records. Records are placed into an indexed file while in load mode or inserted in process mode with a new high key.

basic exchange format. A standard format for exchanging data on diskettes between systems or devices.

binary synchronous device data block (BSCDDB). A control block that provides the information to control one Series/1 Binary Synchronous Adapter. It determines the line characteristics and provides dedicated storage for that line.

Glossary of Terms and Abbreviations

block. (1) See data block or index block. (2) In the Indexed Method, the unit of space used by the access method to contain indexes and data.

block mode. The transmission mode in which the 3101 Display Station transmits a data stream, which has been edited and stored, when the SEND key is pressed.

BSCAM. See binary synchronous communications access method.

binary synchronous communications access method. A form of binary synchronous I/O control used by the Series/1 to perform data communications between local or remote stations.

BSCDDB. See binary synchronous device data block.

buffer. An area of storage that is temporarily reserved for use in performing an input/output operation, into which data is read or from which data is written. See input buffer and output buffer.

bypass label processing. Access of a tape without any label processing support.

CCB. See terminal control block.

central buffer. The buffer used by the Indexed Access Method for all transfers of information between main storage and indexed files.

character image. An alphabetic, numeric, or special character defined for an IBM 4978 Display Station. Each character image is defined by a dot matrix that is coded into eight bytes.

character image table. An area containing the 256 character images that can be defined for an IBM 4978 Display Station. Each character image is coded into eight bytes, the entire table of codes requiring 2048 bytes of storage.

character mode. The transmission mode in which the 3101 Display Station immediately sends a character when a keyboard key is pressed.

cluster. In an indexed file, a group of data blocks that is pointed to from the same primary-level index block, and includes the primary-level index block. The data records and blocks contained in a cluster are logically contiguous, but are not necessarily physically contiguous.

COD (change of direction). A character used with ACCA terminal to indicate a reverse in the direction of data movement.

cold start. Starting the spool facility by erasing any spooled jobs remaining in the spool data set from any previous spool session.

command. A character string from a source external to the system that represents a request for action by the system.

common area. A user-defined data area that is mapped into the partitions specified on the SYSTEM definition statement. It can

be used to contain control blocks or data that will be accessed by more than one program.

completion code. An indicator that reflects the status of the execution of a program. The completion code is displayed or printed on the program's output device.

constant. A value or address that remains unchanged throughout program execution.

controller. A device that has the capability of configuring the GPIB bus by designating which devices are active, which devices are listeners, and which device is the talker. In Series/1 GPIB implementation, the Series/1 is always the controller.

conversion. See update.

control station. In BSCAM communications, the station that supervises a multipoint connection, and performs polling and selection of its tributary stations. The status of control station is assigned to a BSC line during system generation.

cross-partition service. A function that accesses data in two partitions.

cross-partition supervisor. A supervisor in which one or more supervisor modules reside outside of partition 1 (address space 0).

data block. In an indexed file, an area that contains control information and data records. These blocks are a multiple of 256 bytes.

data record. In an indexed file, the records containing customer data.

data set. A group of records within a volume pointed to by a directory member entry in the directory for the volume.

data set control block (DSCB). A control block that provides the information required to access a data set, volume or directory using READ and WRITE.

data set shut down. An indexed data set that has been marked (in main storage only) as unusable due to an error.

DCE. See directory control entry.

device data block (DDB). A control block that describes a disk or diskette volume.

direct access. (1) The access method used to READ or WRITE records on a disk or diskette device by specifying their location relative the beginning of the data set or volume. (2) In the Indexed Access Method, locating any record via its key without respect to the previous operation. (3) A condition in terminal I/O where a READTEXT or a PRINTTEXT is directed to a buffer which was previously enqueued upon by an IOCB.

directory. (1) A series of contiguous records in a volume that describe the contents in terms of allocated data sets and free space. (2) A series of contiguous records on a device that describe the contents in terms of allocated volumes and free space. (3) For the Indexed Access Method Version 2, a data set that defines the relationship between primary and secondary indexed files (secondary index support).

directory control entry (DCE). The first 32 bytes of the first record of a directory in which a description of the directory is stored.

directory member entry (DME). A 32-byte directory entry describing an allocated data set or volume.

display station. An IBM 4978, 4979, or 3101 display terminal or similar terminal with a keyboard and a video display.

DME. See directory member entry.

DSCB. See data set control block.

dynamic storage. An increment of storage that is appended to a program when it is loaded.

end-of-data indicator. A code that signals that the last record of a data set has been read or written. End-of-data is determined by an end-of-data pointer in the DME or by the physical end of the data set.

ECB. See event control block.

EDL. See Event Driven Language.

emulator. The portion of the Event Driven Executive supervisor that interprets EDL instructions and performs the function specified by each EDL statement.

end-of-tape (EOT). A reflective marker placed near the end of a tape and sensed during output. The marker signals that the tape is nearly full.

enter key. The key on the display terminal keyboard that, if pressed, tells the operating system to read the information you entered.

event control block (ECB). A control block used to record the status (occurred or not occurred) of an event; often used to synchronize the execution of tasks. ECBs are used in conjunction with the WAIT and POST instructions.

Event Driven Language (EDL). The language for input to the Event Driven Executive compiler (\$EDXASM), or the Macro and Host assemblers in conjunction with the Event Driven Executive macro libraries. The output is interpreted by the Event Driven Executive emulator.

EXIO (execute input or output). An EDL facility that provides user controlled access to Series/1 input/output devices.

external label. A label attached to the outside of a tape that identifies the tape visually. It usually contains items of identification such as file name and number, creation data, number of volumes, department number, and so on.

external name (EXTRN). The 1- to 8-character symbolic EBCDIC name for an entry point or data field that is not defined within the module that references the name.

FCA. See file control area.

FCB. See file control block.

file. A set of related records treated as a logical unit. Although file is often used interchangeably with data set, it usually refers to an indexed or a sequential data set.

file control area (FCA). A Multiple Terminal Manager data area that describes a file access request.

file control block (FCB). The first block of an indexed file. It contains descriptive information about the data contained in the file.

file control block extension. The second block of an indexed file. It contains the file definition parameters used to define the file.

file manager. A collection of subroutines contained within the program manager of the Multiple Terminal Manager that provides common support for all disk data transfer operations as needed for transaction-oriented application programs. It supports indexed and direct files under the control of a single callable function.

floating point. A positive or negative number that can have a decimal point.

formatted screen image. A collection of display elements or display groups (such as operator prompts and field input names and areas) that are presented together at one time on a display device.

free pool. In an indexed data set, a group of blocks that can be used for either data blocks or index blocks. These differ from other free blocks in that these are not initially assigned to specific logical positions in the file.

free space. In an indexed file, records blocks that do not currently contain data, and are available for use.

free space entry (FSE). An 8-byte directory entry defining an area of free space within a volume or a device.

FSE. See free space entry.

general purpose interface bus. The IEEE Standard 488-1975 that allows various interconnected devices to be attached to the GPIB adapter (RPQ D02118).

Glossary of Terms and Abbreviations

GPIB. See general purpose interface bus.

group. A unit of 100 records in the spool data set allocated to a spool job.

H exchange format. A standard format for exchanging data on diskettes between systems or devices.

host assembler. The assembler licensed program that executes in a 370 (host) system and produces object output for the Series/1. The source input to the host assembler is coded in Event Driven Language or Series/1 assembler language. The host assembler refers to the System/370 Program Preparation Facility (5798-NNQ).

host system. Any system whose resources are used to perform services such as program preparation for a Series/1. It can be connected to a Series/1 by a communications link.

IACB. See indexed access control block.

IAR. See instruction address register.

ICB. See indexed access control block.

IIB. See interrupt information byte.

image store. The area in a 4978 that contains the character image table.

immediate data. A self-defining term used as the operand of an instruction. It consists of numbers, messages or values which are processed directly by the computer and which do not serve as addresses or pointers to other data in storage.

index. In an indexed file, an ordered collection of pairs of keys and pointers, used to sequence and locate records.

index block. In an indexed file, an area that contains control information and index entries. These blocks are a multiple of 256 bytes.

indexed access control block (IACB/ICB). The control block that relates an application program to an indexed file.

indexed access method. An access method for direct or sequential processing of fixed-length records by use of a record's key.

indexed data set. Synonym for indexed file.

indexed file. A file specifically created, formatted and used by the Indexed Access Method. An indexed file is sometimes called an indexed data set.

index entry. In an indexed file, a key-pointer pair, where the pointer is used to locate a lower-level index block or a data block.

index register (#1, #2). Two words defined in EDL and contained in the task control block for each task. They are used to contain data or for address computation.

input buffer. (1) See buffer. (2) In the Multiple Terminal Manager, an area for terminal input and output.

input output control block (IOCB). A control block containing information about a terminal such as the symbolic name, size and shape of screen, the size of the forms in a printer, or an optional reference to a user provided buffer.

instruction address register (IAR). The pointer that identifies the machine instruction currently being executed. The Series/1 maintains a hardware IAR to determine the Series/1 assembler instruction being executed. It is located in the level status block (LSB).

integer. A positive or negative number that has no decimal point.

interactive. The mode in which a program conducts a continuous dialogue between the user and the system.

internal label. An area on tape used to record identifying information (similar to the identifying information placed on an external label). Internal labels are checked by the system to ensure that the correct volume is mounted.

interrupt information byte (IIB). In the Multiple Terminal Manager, a word containing the status of a previous input/output request to or from a terminal.

invoke. To load and activate a program, utility, procedure, or subroutine into storage so it can run.

job. A collection of related program execution requests presented in the form of job control statements, identified to the jobstream processor by a JOB statement.

job control statement. A statement in a job that specifies requests for program execution, program parameters, data set definitions, sequence of execution, and, in general, describes the environment required to execute the program.

job stream processor. The job processing facility that reads job control statements and processes the requests made by these statements. The Event Driven Executive job stream processor is \$JOBUTIL.

jumper. (1) A wire or pair of wires which are used for the arbitrary connection between two circuits or pins in an attachment card. (2) To connect wire(s) to an attachment card or to connect two circuits.

key. In the Indexed Access Method, one or more consecutive characters used to identify a record and establish its order with respect to other records. See also key field.

key field. A field, located in the same position in each record of an indexed file, whose content is used for the key of a record.

level status block (LSB). A Series/1 hardware data area that contains processor status. This area is eleven words in length.

library. A set of contiguous records within a volume. It contains a directory, data sets and/or available space.

line. A string of characters accepted by the system as a single input from a terminal; for example, all characters entered before the carriage return on the teletypewriter or the ENTER key on the display station is pressed.

link edit. The process of resolving external symbols in one or more object modules. A link edit is performed with \$EDXLINK whose output is a loadable program.

listener. A controller or active device on a GPIB bus that is configured to accept information from the bus.

load mode. In the Indexed Access Method, the mode in which records are loaded into base record slots in an indexed file.

load module. A single module having cross references resolved and prepared for loading into storage for execution. The module is the output of the \$UPDATE or \$UPDATEH utility.

load point. (1) Address in the partition where a program is loaded. (2) A reflective marker placed near the beginning of a tape to indicate where the first record is written.

lock. In the Indexed Access Method, a method of indicating that a record or block is in use and is not available for another request.

logical screen. A screen defined by margin settings, such as the TOPM, BOTM, LEFTM and RIGHTM parameters of the TERMINAL or IOCB statement.

LSB. See level status block.

mapped storage. The processor storage that you defined on the SYSTEM statement during system generation.

member. A term used to identify a named portion of a partitioned data set (PDS). Sometimes member is also used as a synonym for a data set. See data set.

menu. A formatted screen image containing a list of options. The user selects an option to invoke a program.

menu-driven. The mode of processing in which input consists of the responses to prompting from an option menu.

message. In data communications, the data sent from one station to another in a single transmission. Stations communication with a series of exchanged messages.

multifile volume. A unit of recording media, such as tape reel or disk pack, that contains more than one data file.

multiple terminal manager. An Event Driven Executive licensed program that provides support for transaction-oriented applications on a Series/1. It provides the capability to define transactions and manage the programs that support those transactions. It also manages multiple terminals as needed to support these transactions.

multivolume file. A data file that, due to its size, requires more than one unit of recording media (such as tape reel or disk pack) to contain the entire file.

new high key. A key higher than any other key in an indexed file.

nonlabeled tapes. Tapes that do not contain identifying labels (as in standard labeled tapes) and contain only files separated by tapemarks.

null character. A user-defined character used to define the unprotected fields of a formatted screen.

option selection menu. A full screen display used by the Session Manager to point to other menus or system functions, one of which is to be selected by the operator. (See primary option menu and secondary option menu.)

output buffer. (1) See buffer. (2) In the Multiple Terminal Manager, an area used for screen output and to pass data to subsequent transaction programs.

overlay. The technique of reusing a single storage area allocated to a program during execution. The storage area can be reused by loading it with overlay programs that have been specified in the PROGRAM statement of the program or by calling overlay segments that have been specified in the OVERLAY statement of \$EDXLINK.

overlay area. A storage area within a program reserved for overlay programs specified in the PROGRAM statement or overlay segments specified in the OVERLAY statement in \$EDXLINK.

overlay program. A program in which certain control sections can use the same storage location at different times during execution. An overlay program can execute concurrently as an asynchronous task with other programs and is specified in the EDL PROGRAM statement in the main program.

overlay segment. A self-contained portion of a program that is called and sequentially executes as a synchronous task. The entire program that calls the overlay segment need not be maintained in storage while the overlay segment is executing. An overlay segment is specified in the OVERLAY statement of \$EDXLINK or \$XPSLINK (for initialization modules).

overlay segment area. A storage area within a program or supervisor reserved for overlay segments. An overlay segment area is specified with the OVLAREA statement of \$EDXLINK.

Glossary of Terms and Abbreviations

parameter selection menu. A full screen display used by the Session Manager to indicate the parameters to be passed to a program.

partition. A contiguous fixed-sized area of storage. Each partition is a separate address space.

performance volume. A volume whose name is specified on the DISK definition statement so that its address is found during IPL, increasing system performance when a program accesses the volume.

physical timer. Synonym for timer (hardware).

polling. In data communications, the process by which a multipoint control station asks a tributary if it can receive messages.

precision. The number of words in storage needed to contain a value in an operation.

prefind. To locate the data sets or overlay programs to be used by a program and to store the necessary information so that the time required to load the prefound items is reduced.

primary file. An indexed file containing the data records and primary index.

primary file entry. For the Indexed Access Method Version 2, an entry in the directory describing a primary file.

primary index. The index portion of a primary file. This is used to access data records when the primary key is specified.

primary key. In an indexed file, the key used to uniquely identify a data record.

primary-level index block. In an indexed file, the lowest level index block. It contains the relative block numbers (RBNs) and high keys of several data blocks. See cluster.

primary menu. The program selection screen displayed by the Multiple Terminal Manager.

primary option menu. The first full screen display provided by the Session Manager.

primary station. In a Series/1-to-Series/1 Attachment, the processor that controls communication between the two computers. Contrast with secondary station.

primary task. The first task executed by the supervisor when a program is loaded into storage. It is identified by the PROGRAM statement.

priority. A combination of hardware interrupt level priority and a software ranking within a level. Both primary and secondary tasks will execute asynchronously within the system according to the priority assigned to them.

process mode. In the Indexed Access Method, the mode in which records can be retrieved, updated, inserted, or deleted.

processor status word (PSW). A 16-bit register used to (1) record error or exception conditions that may prevent further processing and (2) hold certain flags that aid in error recovery.

program. A disk- or diskette-resident collection of one or more tasks defined by a PROGRAM statement; the unit that is loaded into storage. (See primary task and secondary task.)

program header. The control block found at the beginning of a program that identifies the primary task, data sets, storage requirements and other resources required by a program.

program/storage manager. A component of the Multiple Terminal Manager that controls the execution and flow of application programs within a single program area and contains the support needed to allow multiple operations and sharing of the program area.

protected field. A field in which the operator cannot use the keyboard to enter, modify, or erase data.

PSW. See processor status word.

QCB. See queue control block.

QD. See queue descriptor.

QE. See queue element.

queue control block (QCB). A data area used to serialize access to resources that cannot be shared. See serially reusable resource.

queue descriptor (QD). A control block describing a queue built by the DEFINEQ instruction.

queue element (QE). An entry in the queue defined by the queue descriptor.

quiesce. To bring a device or a system to a halt by rejection of new requests for work.

quiesce protocol. A method of communication in one direction at a time. When sending node wants to receive, it releases the other node from its quiesced state.

record. (1) The smallest unit of direct access storage that can be accessed by an application program on a disk or diskette using READ and WRITE. Records are 256 bytes in length. (2) In the Indexed Access Method, the logical unit that is transferred between \$IAM and the user's buffer. The length of the buffer is defined by the user. (3) In BSCAM communications, the portions of data transmitted in a message. Record length (and, therefore, message length) can be variable.

recovery. The use of backup data to re-create data that has been lost or damaged.

reflective marker. A small adhesive marker attached to the reverse (nonrecording) surface of a reel of magnetic tape. Normally, two reflective markers are used on each reel of tape. One indicates the beginning of the recording area on the tape (load point), and the other indicates the proximity to the end of the recording area (EOT) on the reel.

relative block address (RBA). The location of a block of data on a 4967 disk relative to the start of the device.

relative record number. An integer value identifying the position of a record in a data set relative to the beginning of the data set. The first record of a data set is record one, the second is record two, the third is record three.

relocation dictionary (RLD). The part of an object module or load module that is used to identify address and name constants that must be adjusted by the relocating loader.

remote management utility control block (RCB). A control block that provides information for the execution of remote management utility functions.

reorganize. The process of copying the data in an indexed file to another indexed file in a manner that rearranges the data for more optimum processing and free space distribution.

restart. Starting the spool facility w the spool data set contains jobs from a previous session. The jobs in the spool data set can be either deleted or printed when the spool facility is restarted.

return code. An indicator that reflects the results of the execution of an instruction or subroutine. The return code is usually placed in the task code word (at the beginning of the task control block).

roll screen. A display screen which is logically segmented into an optional history area and a work area. Output directed to the screen starts display at the beginning of the work area and continues on down in a line-by-line sequence. When the work area gets full, the operator presses ENTER/SEND and its contents are shifted into the optional history area and the work area itself is erased. Output now starts again at the beginning of the work area.

SBIOCB. See sensor based I/O control block.

second-level index block. In an indexed data set, the second-lowest level index block. It contains the addresses and high keys of several primary-level index blocks.

secondary file. See secondary index.

secondary index. For the Indexed Access Method Version 2, an indexed file used to access data records by their secondary keys. Sometimes called a secondary file.

secondary index entry. For the Indexed Access Method Version 2, this an an entry in the directory describing a secondary index.

secondary key. For the Indexed Access Method Version 2, the key used to uniquely identify a data record.

secondary option menu. In the Session Manager, the second in a series of predefined procedures grouped together in a hierarchical structure of menus. Secondary option menus provide a breakdown of the functions available under the session manager as specified on the primary option menu.

secondary task. Any task other than the primary task. A secondary task must be attached by a primary task or another secondary task.

secondary station. In a Series/1-to-Series/1 Attachment, the processor that is under the control of the primary station.

sector. The smallest addressable unit of storage on a disk or diskette. A sector on a 4962 or 4963 disk is equivalent to an Event Driven Executive record. On a 4964 or 4966 diskette, two sectors are equivalent to an Event Driven Executive record.

selection. In data communications, the process by which the multipoint control station asks a tributary station if it is ready to send messages.

self-defining term. A decimal, integer, or character that the computer treats as a decimal, integer, or character and not as an address or pointer to data in storage.

sensor based I/O control block (SBIOCB). A control block containing information related to sensor I/O operations.

sequential access. The processing of a data set in order of occurrence of the records in the data set. (1) In the Indexed Access Method, the processing of records in ascending collating sequence order of the keys. (2) When using READ/WRITE, the processing of records in ascending relative record number sequence.

serially reusable resource (SRR). A resource that can only be accessed by one task at a time. Serially reusable resources are usually managed via (1) a QCB and ENQ/DEQ statements or (2) an ECB and WAIT/POST statements.

service request. A device generated signal used to inform the GPIB controller that service is required by the issuing device.

session manager. A series of predefined procedures grouped together as a hierarchical structure of menus from which you select the utility functions, program preparation facilities, and language processors needed to prepare and execute application programs. The menus consist of a primary option menu that displays functional groupings and secondary option menus that display a breakdown of these functional groupings.

shared resource. A resource that can be used by more than one task at the same time.

Glossary of Terms and Abbreviations

shut down. See data set shut down.

source module/program. A collection of instructions and statements that constitute the input to a compiler or assembler. Statements may be created or modified using one of the text editing facilities.

spool job. The set of print records generated by a program (including any overlays) while enqueued to a printer designated as a spool device.

spool session. An invocation and termination of the spool facility.

spooling. The reading of input data streams and the writing of output data streams on storage devices, concurrently with job execution, in a format convenient for later processing or output operations.

SRQ. See service request.

stand-alone dump. An image of processor storage written to a diskette.

stand-alone dump diskette. A diskette supplied by IBM or created by the \$DASDI utility.

standard labels. Fixed length 80-character records on tape containing specific fields of information (a volume label identifying the tape volume, a header label preceding the data records, and a trailer label following the data records).

static screen. A display screen formatted with predetermined protected and unprotected areas. Areas defined as operator prompts or input field names are protected to prevent accidental overlay by input data. Areas defined as input areas are not protected and are usually filled in by an operator. The entire screen is treated as a page of information.

station. In BSCAM communications, a BSC line attached to the Series/1 and functioning in a point-to-point or multipoint connection. Also, any other terminal or processor with which the Series/1 communicates.

subroutine. A sequence of instructions that may be accessed from one or more points in a program.

supervisor. The component of the Event Driven Executive capable of controlling execution of both system and application programs.

system configuration. The process of defining devices and features attached to the Series/1.

SYSGEN. See system generation.

system generation. The processing of defining I/O devices and selecting software options to create a supervisor tailored to the needs of a specific Series/1 hardware configuration and application.

system partition. The partition that contains the root segment of the supervisor (partition number 1, address space 0).

talker. A controller or active device on a GPIB bus that is configured to be the source of information (the sender) on the bus.

tape device data block (TDB). A resident supervisor control block which describes a tape volume.

tapemark. A control character recorded on tape used to separate files.

task. The basic executable unit of work for the supervisor. Each task is assigned its own priority and processor time is allocated according to this priority. Tasks run independently of each other and compete for the system resources. The first task of a program is the primary task. All tasks attached by the primary task are secondary tasks.

task code word. The first two words (32 bits) of a task's TCB; used by the emulator to pass information from system to task regarding the outcome of various operations, such as event completion or arithmetic operations.

task control block (TCB). A control block that contains information for a task. The information consists of pointers, save areas, work areas, and indicators required by the supervisor for controlling execution of a task.

task supervisor. The portion of the Event Driven Executive that manages the dispatching and switching of tasks.

TCB. See task control block.

terminal. A physical device defined to the EDX system using the TERMINAL configuration statement. EDX terminals include directly attached IBM displays, printers and devices that communicate with the Series/1 in an asynchronous manner.

terminal control block (CCB). A control block that defines the device characteristics, provides temporary storage, and contains links to other system control blocks for a particular terminal.

terminal environment block (TEB). A control block that contains information on a terminal's attributes and the program manager operating under the Multiple Terminal Manager. It is used for processing requests between the terminal servers and the program manager.

terminal screen manager. The component of the Multiple Terminal Manager that controls the presentation of screens and communications between terminals and transaction programs.

terminal server. A group of programs that perform all the input/output and interrupt handling functions for terminal devices under control of the Multiple Terminal Manager.

terminal support. The support provided by EDX to manage and control terminals. See terminal.

timer. The timer features available with the Series/1 processors. Specifically, the 7840 Timer Feature card (4955 only) or the native timer (4952, 4954, and 4956). Only one or the other is supported by the Event Driven Executive.

trace range. A specified number of instruction addresses within which the flow of execution can be traced.

transaction oriented applications. Program execution driven by operator actions, such as responses to prompts from the system. Specifically, applications executed under control of the Multiple Terminal Manager.

transaction program. See transaction-oriented applications.

transaction selection menu. A Multiple Terminal Manager display screen (menu) offering the user a choice of functions, such as reading from a data file, displaying data on a terminal, or waiting for a response. Based upon the choice of option, the application program performs the requested processing operation.

tributary station. In BSCAM communications, the stations under the supervision of a control station in a multipoint connection. They respond to the control station's polling and selection.

unmapped storage. The processor storage in your processor that you did not define on the SYSTEM statement during system generation.

unprotected field. A field in which the operator can use the keyboard to enter, modify or erase data. Also called non-protected field.

update. (1) To alter the contents of storage or a data set. (2) To convert object modules, produced as the output of an assembly or compilation, or the output of the linkage editor, into a form that can be loaded into storage for program execution and to update the directory of the volume on which the loadable program is stored.

user exit. (1) Assembly language instructions included as part of an EDL program and invoked via the USER instruction. (2) A point in an IBM-supplied program where a user written routine can be given control.

variable. An area in storage, referred to by a label, that can contain any value during program execution.

vary offline. (1) To change the status of a device from online to offline. When a device is offline, no data set can be accessed on that device. (2) To place a disk or diskette in a state where it is unknown by the system.

vary online. To place a device in a state where it is available for use by the system.

vector. An ordered set or string of numbers.

volume. A disk, diskette, or tape subdivision defined using \$INITDSK or \$TAPEUT1.

volume descriptor entry (VDE). A resident supervisor control block that describes a volume on a disk or diskette.

volume label. A label that uniquely identifies a single unit of storage media.



Index

The following index contains entries for this book only. See the *Library Guide and Common Index* for a Common Index to all Event Driven Executive books.

Special Characters

- \$A - list partitions and programs
 - procedure OP-227
 - syntax OP-357
- \$B - blank display screen
 - procedure OP-43
 - syntax OP-358
- \$C - cancel program
 - procedure OP-236
 - syntax OP-358
- \$COMPRES utility
 - compress examples
 - a volume OP-142
 - IPL volume OP-142
 - compressing devices OP-142
- \$COPY utility
 - copy
 - data set OP-164
 - dump to diskette OP-299
 - from basic exchange OP-171
 - to basic exchange OP-173
 - volume OP-162
- \$COPYUT1 utility
 - copy multiple data sets OP-155
 - copy multiple data sets by prefix OP-159
 - use in backup procedure OP-322, OP-327
- \$CP - change terminal partition
 - procedure OP-50
 - syntax OP-359
- \$D - dump storage
 - procedure OP-306
- syntax OP-360
- \$DASDI utility
 - creating stand-alone dump diskette OP-207
 - initialize
 - bytes-per-sector format OP-102
 - EDX format OP-93
 - format types OP-90
 - standard for information interchange OP-97
- \$DISKUT1 utility
 - allocate data set OP-124
 - delete
 - data sets OP-137
 - determine free space on a volume OP-205
 - determining volume a data set is on OP-185
 - in backup procedure OP-322
 - list directory
 - all data sets in CTS/RBA mode OP-187
 - all data sets on a volume OP-187
 - data sets on all volumes OP-185
 - data sets starting with specified prefix OP-185
 - data-type data sets OP-191
 - data-type data sets in CTS/RBA mode OP-191
 - free space available in volume OP-205
 - one data set OP-187
 - program-type data sets OP-191
 - program-type data sets in CTS/RBA mode OP-191
 - volumes OP-181
 - renaming \$SMINIT OP-78
 - renaming data sets OP-132
- \$DISKUT2 utility
 - determining version level OP-308
 - list

Index

- all/portion of data set OP-195
- data-type data set OP-198
- program-type data set OP-202
- source data set OP-195
- \$DUMP** utility
 - print a dump OP-293
- \$E** - eject printer page
 - procedure OP-252
 - syntax OP-360
- \$EDXASM** utility
 - output listing size OP-395
- \$EDXLINK** utility
 - output listing size OP-396
- \$EDXNUC** supervisor data set
 - allocating OP-106
- \$FSEEDIT** utility
 - entering/editing data OP-210
 - PF keys OP-353
 - work data set OP-210
- \$HXUT1** utility
 - allocate H-exchange data set OP-126
 - copy
 - copy to EDX data set OP-175
 - copy to H-exchange data set OP-178
 - initialize volume OP-111
- \$INITDSK** utility
 - allocate
 - allocate volume OP-121
 - delete volumes OP-136
 - in backup procedure OP-321
 - initialize
 - IPL text OP-147
 - volume directories OP-106
 - IPL text OP-106
 - listing volume directories OP-183
 - rename
 - volumes OP-131
 - use in backup procedure OP-326
- \$INITIAL** programs
 - session manager OP-78
- \$IOTEST** utility
 - list
 - devices attached to Series/1 OP-318
 - devices supported by supervisor OP-318
- \$JOBQ**, job queue processor
 - See job queue processor
- \$JOBQUT** utility
 - change logging terminal OP-233
 - delete job queues OP-239
 - end
 - end job queue processing OP-240
 - resume job queue processing OP-231
 - starting job queue processor OP-247
 - suspend
 - job queue processing OP-231
- \$JOBUTIL** utility
 - coding a procedure OP-241
 - command syntax OP-376
 - entering a procedure OP-243
 - example procedure OP-245
 - planning a procedure OP-241
 - procedures log OP-317
 - responding to a PAUSE OP-235
 - starting
 - with \$L OP-222
 - with \$SUBMIT OP-225
 - with the session manager OP-223
- \$L** - load program
 - \$JOBUTIL procedure OP-222
 - individual programs OP-217
 - syntax OP-361
- \$MOVEVOL** utility
 - restoring a system OP-334
 - saving a system OP-319
- \$P** - patch storage
 - syntax OP-362
- \$S** - control printer spooling
 - ALT, alter printing
 - procedure OP-276
 - syntax OP-363
 - DALL, delete all jobs
 - procedure OP-285
 - syntax OP-364
 - DE, delete one job
 - procedure OP-285
 - syntax OP-365
 - DG, delete jobs by prefix
 - procedure OP-285
 - syntax OP-365
 - DISP, display status
 - procedure OP-275
 - syntax OP-366
 - HOLD, hold job
 - procedure OP-282
 - syntax OP-366
 - KEEP, keep/release job
 - procedure OP-284
 - syntax OP-367
 - REL, release jobs
 - procedure OP-283
 - syntax OP-368
 - STOP, stop spooling
 - procedure OP-274
 - syntax OP-368
 - WRES, restart writer
 - procedure OP-280
 - syntax OP-369
 - WSTP, stop writer
 - procedure OP-279
 - syntax OP-370
 - WSTR, start writer
 - procedure OP-278
 - syntax OP-371
- \$SMMAIN**
 - See session manager
- \$SMMLLOG** session manager logon menu
 - defined OP-72
 - example OP-79
- \$SMMPRIM** primary option menu
 - defined OP-72

example OP-79

\$SPLUT1 utility

- changing spool group size OP-268
- maximum active spool jobs, changing OP-267
- maximum spool jobs, changing OP-266
- separator page, changing OP-270
- setting spool start mode OP-262
- specifying spool devices OP-271
- spool data set, changing OP-264
- status display OP-391

\$SPOOL, program output spooling

See spooling

\$SUBMIT utility

- delete a job OP-237
- display
 - job status OP-228
- hold a batch job OP-230
- loading programs OP-225
- release a held job OP-229

\$S1ASM utility

- output listing size OP-395

\$T - set date, time

- procedure OP-26
- syntax OP-372

\$TAPEUT1 utility

- allocate a data set OP-127
- change
 - label processing attributes OP-118
 - tape drive attributes OP-118
- copy a data set OP-167
- dump tape records OP-302
- initialize a tape OP-112
- restore
 - disk or disk volume from tape OP-338
 - system OP-338
- save
 - a system OP-325
 - volume on tape OP-325

\$TERMUT1 utility

- change
 - a terminal offline OP-52
 - a terminal online OP-54
 - hard-copy device OP-49
 - page formatting parameters OP-255
- list terminal name/type/address OP-44
- reassign terminal address OP-48
- rename a terminal OP-47
- vary

\$TERMUT2 utility

- load
- save
- setting PF keys OP-55

\$TERMUT3 utility

- procedure OP-45

\$TRAP utility

- printing a dump OP-293

\$U operator command

\$VARYOFF - set device offline

- procedure OP-20
- syntax OP-373

\$VARYON - set device online

- diskette procedure OP-18
- syntax OP-374
- tape procedure OP-128

\$W - display date, time

- procedure OP-27
- syntax OP-375

A

active spool jobs, changing maximum number OP-267

addresses

- finding hardware OP-318
- re-assigning terminal OP-48

advancing printer forms

- procedure OP-252
- syntax OP-360

AL - allocate data set

- syntax OP-376
- using OP-244

aligning forms

- for spool jobs OP-363
- printer OP-252

allocate

- data set
 - \$EDXNUC** OP-106
 - basic exchange OP-97
 - disk/diskette OP-124
 - H-exchange OP-126
 - spool OP-260
 - tape OP-127
 - with **\$JOBUTIL** OP-376
- volume
 - on disk OP-121
 - on diskette OP-106, OP-121
 - on tape OP-112

ALT subcommand

- procedure OP-276
- syntax OP-363

alter

- spool job printing
 - \$S ALT** syntax OP-363
 - procedure OP-276

alternate session menu, session manager

- defined OP-72
- selecting OP-79

attached devices, listing OP-318

attention key OP-41

automatic initialization, using OP-325

Index

B

background, running programs OP-219
backup system
 backup log OP-317
 determining number of diskettes OP-320
 formatting diskettes for OP-320
 restarting OP-311
 restoring from diskette OP-334
 restoring from tape OP-338
 saving on diskettes OP-319
 saving on tape OP-325
basic console OP-22
basic exchange diskette
 copying from OP-171
 copying to OP-173
 format defined OP-90
 initializing OP-97
blank forms
 \$JOBUTIL planning OP-399
 \$JOBUTIL procedures log OP-403
 backup log OP-407
 diskette log OP-411
 job instructions OP-415
 operations log OP-419
 problem recording OP-423
 tape log OP-431
blanking display screen
 \$B syntax OP-358
 procedure OP-43
 session manager OP-83
block mode, defined OP-42
bottom margin, changing OP-253
branch
 \$JOBUTIL OP-244
 JUMP syntax OP-382
 LABEL syntax OP-383
bytes per sector diskette format
 defined OP-90
 initializing a diskette for OP-102

C

cancel
 job queue jobs OP-237
 programs
 \$C syntax OP-358
 procedure OP-236
 suspending session manager OP-82
 utilities OP-236
change
 control store OP-55
 diskettes OP-15, OP-19
 display screen format OP-51
 hard-copy device OP-49
 keyboard values OP-55
 page formatting OP-253
 PF key OP-49, OP-55
 source data set OP-210

 spool options OP-261
 tape attributes OP-118
 tape label OP-112
terminal
 address OP-48
 displaying job queue status OP-233
 name OP-47
 partition OP-50
 upper/lowercase characters OP-56
 volume
 volume or data set name OP-131
character mode, defined OP-42
character set, specifying OP-255
character/local function code OP-55
characters, terminals
 deleting OP-35
 entering new OP-34
 erasing OP-34
 erasing to end of field OP-37
 erasing to end of line OP-38
 erasing to end of screen OP-39
 inserting OP-36
 lowercase OP-56
 protected OP-34
 replacing OP-35
circuit breaker
 location OP-11
 resetting OP-11
clear
 \$B syntax OP-358
 procedure OP-43
 session manager OP-83
command
 \$JOBUTIL
 See \$JOBUTIL utility
 entering OP-41
 operator
 See operator commands
comment in procedure OP-390
compress
 disk, diskette, or volume OP-142
configuring
 terminal OP-253
control
control record, spooling OP-258
control store
 changing OP-55
 data table OP-55
 data table example OP-65
 defined OP-55
 loading OP-58
 saving OP-70
copy
 \$TRAP dump to diskette OP-299
 backup system
 from diskette OP-334
 from tape OP-338
 on diskette OP-319
 on tape OP-325
 basic exchange data set

from OP-171
to OP-173
data on EDX systems
 local to remote OP-148
 multiple data sets with allocation OP-155, OP-159
 one data set without allocation OP-164
 remote to local OP-148
 tape data sets OP-167
 using \$TRANS utility OP-148
 volume OP-162
data to/from non-EDX systems
 basic exchange data set OP-171, OP-173
 H exchange data set OP-175, OP-178
member
create
 See also allocate
 stand-alone dump diskette OP-207
cursor, moving OP-32
custom menus, session manager OP-73

D

DALL subcommand
 procedure OP-285
 syntax OP-364
data
 See also data set
 backing up on diskette OP-319
 backing up on tape OP-325
 copying data
 See copy
 editing OP-210
 entering OP-210
 entering on display terminal OP-41
 how stored OP-86
 listing OP-195
 preparing organization OP-91
 restoring from diskette OP-334
 restoring from tape OP-338
 types defined OP-86
data set
 allocate
 AL command OP-376
 basic exchange OP-97
 disk/diskette OP-124
 H-exchange OP-126
 tape OP-127
 basic exchange
 See basic exchange diskette
 copying data
 data type
 allocating OP-124
 defined OP-87
 defined OP-87
 delete
 DE subcommand OP-377
 generic OP-139
 multiple data sets OP-139
 one data set OP-137

determining volume a data set is on OP-185
directories
 See data set directory, listing
H-exchange
 See H-exchange diskette
list contents
 data-type OP-198
 program-type OP-202
 source data set OP-195
list directory
 all data sets in CTS/RBA mode OP-187
 all data sets on a volume OP-187
 data sets on all volumes OP-185
 data sets starting with specified prefix OP-185
 data-type data sets OP-191
 data-type data sets in CTS/RBA mode OP-191
 free space available in volume OP-205
 one data set OP-187
 program-type data sets OP-191
 program-type data sets in CTS/RBA mode OP-191
 volumes OP-181
organization
program type
 allocating OP-124
 defined OP-87
rename OP-132
restoring OP-334, OP-338
saving OP-319, OP-325
spool data set OP-258
data set directory
 defined OP-88
 determining volume a data set is on OP-185
 initializing OP-106, OP-121
 list
 listing
 all data sets on all volumes OP-185
 all data sets on one volume OP-187
 data-type data sets OP-191
 one data set OP-187
 program-type data sets OP-191
 number of
 data set group, spooling OP-259
data type data sets
 See data set
date
 displaying
 command syntax OP-375
 procedure OP-27
 setting
 command syntax OP-372
 procedure OP-26
DE subcommand
 syntax OP-377
 using OP-244
DE, \$\$ delete spool job
 procedure OP-285
 syntax OP-365
define
 data set
 See allocate

Index

- define key
 - clearing OP-68
 - defined OP-55
 - specifying OP-66
 - define mode OP-55
 - DEL key OP-35
 - delete
 - data set
 - data sets OP-136
 - DE command syntax OP-377
 - job queue OP-237
 - spool jobs
 - \$S DALL syntax OP-364
 - \$S DG syntax OP-365
 - command syntax OP-365
 - procedure OP-285
 - tape data set OP-127
 - volumes OP-136
 - determine
 - data set
 - number
 - terminal
 - device
 - compressing OP-142
 - determining those on system OP-318
 - hardware OP-8
 - specifying for spooling OP-271
 - spool device OP-259
 - vary offline
 - command syntax OP-373
 - procedure OP-20
 - vary online
 - command syntax OP-374
 - diskette procedure OP-18
 - tape procedure OP-128
 - DG subcommand
 - procedure OP-285
 - syntax OP-365
 - directory
 - data set
 - See data set directory
 - disk
 - compressing OP-142
 - copying data
 - See copy
 - data sets
 - See data set
 - data storage OP-86
 - free space, determining OP-205
 - preparing for use OP-91
 - restoring from diskette OP-334
 - restoring from tape OP-338
 - saving on tape OP-325
 - saving volumes OP-319
 - volume
 - See volume
 - diskette
 - backup log OP-317
 - basic exchange
 - See basic exchange diskette
 - bytes per sector format
 - defined OP-90
 - initializing OP-102
 - compressing OP-142
 - construction OP-12
 - copying, data
 - See copy
 - data sets
 - See data set
 - data storage OP-86
 - defined OP-90
 - diskette log OP-316
 - EDX format
 - defined OP-90
 - initializing OP-93
 - format combinations OP-92
 - free space, determining OP-205
 - H-exchange
 - See H-exchange diskette
 - handling OP-13
 - IBM standard for information interchange
 - initializing OP-97
 - initialize
 - basic exchange OP-97
 - bytes-per-sector format OP-102
 - data set directory OP-121
 - EDX format OP-93
 - H-exchange OP-97
 - H-exchange volume OP-111
 - IBM standard for information interchange
 - format OP-97
 - introduction OP-92
 - IPL text OP-106, OP-147
 - stand-alone dump OP-207
 - volume directory OP-106
 - inserting OP-15
 - listing directories
 - See data set directory, listing
 - preparing for use OP-91
 - record sizes OP-121, OP-162, OP-320
 - removing OP-19
 - restoring system OP-334
 - stand-alone dump, creating OP-207
 - system backup OP-319
 - types OP-12
 - vary offline
 - command syntax OP-373
 - procedure OP-20
 - vary online
 - command syntax OP-374
 - procedure OP-18
 - volume
 - See volume
 - volume label defined OP-88
 - 4966 magazines, removing OP-20
- DISP subcommand
 - procedure OP-275
 - syntax OP-366
 - display
 - job queue status OP-228

- partitions
 - \$A syntax OP-357
 - procedure OP-227
- programs
 - \$A syntax OP-357
- REMARK OP-390
- spool status
 - command syntax OP-366
 - procedure OP-275
- storage
 - command syntax OP-360
 - procedure OP-306
- display screen
 - See display terminal
- display stations
 - See display terminal
- display terminal
 - assignments, listing OP-44
 - attention key OP-41
 - change partition OP-50
 - changing addresses OP-48
 - clear screen OP-43
 - clearing the screen OP-43
 - control store
 - changing OP-55
 - loading OP-58
 - DEL key OP-35
 - deleting characters OP-35
 - entering commands OP-41
 - entering new characters OP-34
 - EOF key OP-37
 - EOL key OP-38
 - EOS key OP-39
 - erasing
 - characters OP-34
 - to end of field OP-37
 - to end of line OP-38
 - to end of screen OP-39
 - hard-copy device, changing OP-49
 - INS MOD key OP-36
 - inserting characters OP-36
 - listing names OP-44
 - lowercase characters OP-56
 - moving cursor OP-32
 - operation introduction OP-29
 - partition, changing OP-50
 - PF keys, setting OP-55
 - PF6 key OP-40
 - printer, changing OP-49
 - printing the screen OP-40
 - protected field, defined OP-34
 - renaming OP-47
 - replacing characters OP-35
 - screen
 - adjusting brightness OP-30
 - clearing OP-43
 - erasing OP-37
 - format OP-51
 - printing OP-40
 - protected characters/fields OP-34

- sending a message to another terminal OP-45
- sending commands to EDX OP-42
- static screen OP-34
- switching on OP-30
- unprotected field, defined OP-34
- vary offline OP-52
- vary online OP-54
- downshift mode OP-55
- DS, identify data set
 - syntax OP-378
 - using OP-244
- dump
 - \$D operator command
 - procedure OP-306
 - syntax OP-360
 - copying to diskette OP-299
 - data set
 - data-type OP-198
 - program-type OP-202
 - to tape OP-325
 - disk to tape OP-325
 - printing OP-293
 - stand-alone
 - creating diskette OP-207
 - printing OP-293
 - taking OP-290
 - tape records OP-302
 - to printer
 - to terminal
 - volume to diskette OP-319
 - volume to tape OP-325
- dump diskette, creating OP-207

E

- edit
 - data OP-210
- editor, using \$FSEDIT OP-210
- EDX diskette format OP-90
- EJECT command OP-379
- ejecting printer page
 - command syntax OP-360
 - EJECT OP-379
 - procedure OP-252
- electrical power
 - emergency off OP-10
 - switching off OP-10
 - switching on OP-9
- emergency power off
 - procedure OP-10
 - restoring OP-11
- emergency pull switch, resetting OP-11
- end
 - command syntax OP-379, OP-380
 - job queue processing OP-240
 - job, EOJ command (\$JOBUTIL)
 - nested procedure, EOP command (\$JOBUTIL)
- enter
 - a \$JOBUTIL procedure OP-243

Index

- data OP-210
- operator commands
 - See also operator commands
 - how to OP-41
 - when session manager is active OP-82
- ENTER key, operating OP-42
- EOF key OP-37
- EOJ statement
 - syntax OP-379
 - using OP-244
- EOL key OP-38
- EOP statement
 - syntax OP-380
 - using OP-244
- EOS key OP-39
- erase
 - entire screen OP-43
 - to end of field OP-37
 - to end of line OP-38
 - to end of screen OP-39
- errors
 - determining version level OP-306
 - dumping tapes to printer OP-302
 - gathering information OP-288
 - printing a dump OP-293
 - restarting the system OP-311
 - taking a stand-alone dump OP-290
- EXEC statement
 - syntax OP-380
 - using OP-244
- execute
 - program
 - with \$JOBUTIL OP-222, OP-241
 - with \$L OP-217
 - with \$SUBMIT OP-225
 - with session manager OP-219
 - utility
 - with \$JOBUTIL OP-222, OP-241
 - with \$L OP-217
 - with \$SUBMIT OP-225
 - with session manager OP-219

F

- filing data OP-86
- flexible disk OP-12
- foreground, running programs OP-219
- format
- formatting diskettes
 - See diskette
- forms
 - aligning printer OP-252
 - alignment prompt, spool OP-286
 - backup log OP-317, OP-407
 - diskette log OP-316, OP-411
 - job instructions OP-315, OP-415
 - operations log OP-314, OP-419
 - problem recording OP-288, OP-423
 - procedure planning OP-241, OP-399

- procedures log OP-317, OP-403
- tape log OP-316, OP-431
- forms code, spooling OP-258
- free space, determining OP-205
- full screen defined OP-42
- function code, character/local OP-55
- function ID code OP-56

H

- H-exchange diskette
 - allocate a data set OP-126
 - copying from data sets OP-175
 - copying to data sets OP-178
 - format defined OP-90
 - initializing OP-97
 - initializing volume OP-111
- hardware
 - determining what is on system OP-318
 - devices/units OP-8
 - switching on/off OP-9
- head slot, diskette OP-12
- history lines, changing OP-253
- hold
 - spool jobs
 - command syntax OP-366
 - procedure OP-282
- HOLD subcommand
 - procedure OP-282
 - syntax OP-366

I

- IBM standard for information interchange
 - defined OP-90
 - initializing a diskette for OP-97
- identify
 - batch job stream
 - batch job stream continuation point OP-383
 - batch job stream data set OP-378
- image store
 - defined OP-56
 - loading OP-58
- information storage OP-85
- initialize
 - diskettes
 - See diskette, initialize
 - tapes OP-112
- INS MOD key OP-36
- insert
 - characters OP-36
- interrupt code OP-56
- interrupt keys
 - attention key OP-41
 - enter key OP-42
 - setting PF keys OP-55
- invoke
 - session manager OP-77

text editor OP-211
your operator command (\$U) OP-372
IPL (initial program load)
 automatic IPL OP-25
 determining volume OP-183
 initialize text OP-106, OP-147
 IPL procedures OP-22
 loading session manager OP-78
 manual IPL OP-24
 setting IPL mode OP-24, OP-25
 volume, determining OP-183
IPL diskette for system backup OP-321, OP-326

J

JOB - identify job OP-381
job cross reference chart OP-346
job display format, spool OP-392
job instruction form OP-315
job printing, spool
 altering parameters OP-276
 command syntax OP-363
job procedures
 \$JOBUTIL command syntax OP-376
 coding OP-241
 example OP-245
 planning OP-241
 starting OP-222
job queue
job queue processor
 \$JOBUTIL command syntax OP-376
 changing logging terminal OP-233
 controlling jobs OP-228
 deleting jobs OP-237
 displaying job status OP-228
 ending job queue processing OP-240
 holding jobs OP-230
 loading programs OP-225
 releasing a held job OP-229
 resuming jobs OP-231
 starting OP-247
 suspending jobs OP-231
JOBQUT, job queue controller
 See \$JOBQUT utility
jobs, printer spooling
 See spooling
JUMP subcommand
 syntax OP-382
 using OP-244

K

KEEP subcommand
 procedure OP-284
 syntax OP-367
keeping records OP-314
keeping spool jobs
 command syntax OP-367
 procedure OP-284
kept spool jobs, releasing
 releasing held jobs OP-283, OP-368
 releasing kept jobs OP-284, OP-367
key position OP-56

L

LABEL command
 syntax OP-383
 using OP-244
labeling tapes OP-112, OP-118
left margin, changing OP-253
lines per inch, specifying OP-255
list
 data set contents
 data-type dump OP-198
 program-type OP-202
 source data set OP-195
 data set directory
 all data sets on all volumes OP-185
 all data sets on one volume OP-187
 data-type OP-191
 one data set OP-187
 program-type OP-191
 devices on a system OP-318
 partitions and programs OP-227
 volume directory
 on one device OP-183
load
 control store OP-58
 image store OP-58
 job queue processor OP-247
 program
 \$L syntax OP-361
 EXEC program OP-244, OP-380
 preparing OP-216
 with \$JOBUTIL OP-222
 with \$L OP-217
 with \$SUBMIT OP-225
 with the session manager OP-219
 session manager OP-77
 system (IPL) OP-22
 text editor OP-211
LOG command syntax OP-384
logical volume
 defined OP-87
 initialize OP-106
logoff menu, session manager
 defined OP-72
 procedure OP-84

Index

logon menu, session manager
defined OP-72
procedure OP-79

logs
\$JOBUTIL procedures OP-317
backup OP-317
diskettes OP-316
job instructions OP-315
tape OP-316

lowercase characters
\$FSEDIT CAPS OP-355
specifying OP-56

M

managing your system OP-313
manual IPL procedure OP-24
maximum spool jobs, changing OP-266
members

See data set

menus, session manager
custom defined OP-73
logon/logoff
defined OP-72
example OP-79
parameter input
defined OP-73
example OP-82
primary option
defined OP-72
example OP-79
secondary option
defined OP-73
example OP-81

message
sending to another terminal OP-45
modification level, determining OP-306
modify

See change

move
disk or volume from diskette OP-334
disk or volume from tape OP-338
disk or volume to diskette OP-319
disk, volume or data set to tape OP-325

N

names, finding hardware OP-318
new line key, operating OP-41
NOMSG, no load message
syntax OP-385
using OP-244
nonlabeled tapes
changing label OP-118
initializing OP-112

O

operations
cross reference chart OP-346
getting started OP-7
keeping records OP-314
reading procedures OP-5
operations log OP-314
operator commands
\$A - list partitions and programs
procedure OP-227
syntax OP-357
\$B - blank display screen
procedure OP-43
syntax OP-358
\$C - cancel program
procedure OP-236
syntax OP-358
\$CP - change display terminal partition
procedure OP-50
syntax OP-359
\$D - dump storage
procedure OP-306
syntax OP-360
\$E - eject printer page
procedure OP-252
syntax OP-360
\$L - load program or utility
procedure OP-217
syntax OP-361
\$P - patch storage
syntax OP-362
\$S - control printer spooling
ALT OP-276, OP-363
DALL OP-285, OP-364
DE OP-285, OP-365
DG OP-285, OP-365
DISP OP-275, OP-366
HOLD OP-282, OP-366
KEEP OP-284, OP-367
REL OP-283, OP-368
STOP OP-274, OP-368
WRES OP-280, OP-369
WSTP OP-279, OP-370
WSTR OP-278, OP-371
\$T - set date, time
procedure OP-26
syntax OP-372
\$U - user operator command
syntax OP-372
\$U, user operator command OP-372
\$VARYOFF - set device offline
procedure OP-20
syntax OP-373
\$VARYON - set device online
procedure OP-18
syntax OP-374
\$W - display date, time
procedure OP-27
syntax OP-375

entering OP-41
operator console OP-22
operator errors OP-287
option selection menus, session manager
 defined OP-72
 using OP-80
output listing size, estimating OP-395
output, program
 See spooling
overflow lines, changing OP-255

P

page eject, printer
 \$E syntax OP-360
 procedure OP-252
page size, changing OP-253
paper
 advancing OP-252
 aligning OP-252
parameter input menu
 defined OP-73
 example OP-82
PARM, parameter passing
 syntax OP-386
 using OP-244
partition
 assignment
 \$CP syntax OP-359
 procedure OP-50
 listing
 \$A syntax OP-357
 procedure OP-227
 listing terminal's OP-44
partition assignment
 \$CP syntax OP-359
passing parameters
 \$JOBUTIL OP-244, OP-386
PAUSE command
 responding to OP-235
 syntax OP-387
PF keys
 See program function (PF) keys
power off procedure OP-10
power on procedure OP-9
primary option menu, session manager
 defined OP-72
 example OP-79
print
 printing data set contents
 See data set, list contents
 printing spool jobs
 See spooling
 stand-alone or \$TRAP dump OP-293
 tapes OP-302
 terminal screen OP-40
print density, specifying OP-255
print mode, specifying OP-255
printer spooling
 See spooling
printers
 advancing forms OP-252
 aligning forms OP-252
 assignments, listing OP-44
 changing addresses OP-48
 changing page formatting OP-253
 controlling output OP-251
 ejecting pages OP-252, OP-360
 renaming OP-47
 specifying as spoolable OP-255
 varying online OP-52, OP-54
problem determination
 determining version level OP-306
 dumping tapes OP-302
 printing a dump OP-293
 recording form OP-288
 restarting the system after OP-311
 taking stand-alone dump OP-290
PROC command
 syntax OP-388
 using OP-244
procedure
 reading OP-5
 running job OP-215
 writing OP-241
procedures log OP-317
processor device OP-8
program
 cancelling
 \$C syntax OP-358
 procedure OP-236
 checking and controlling OP-227
 deleting job queues OP-237
 dump
 estimating output listing size OP-396
 identify \$JOBUTIL OP-389
 listing active
 \$A syntax OP-357
 procedure OP-227
 load
 \$L syntax OP-361
 with \$JOBUTIL OP-222
 with \$L OP-217
 with job queue processor OP-225
 with session manager OP-219
 preparing to run OP-216
 problems OP-287
 running background OP-219
 running foreground OP-219
 writing a procedure to run OP-241
PROGRAM command
 syntax OP-389
 using OP-244
program function (PF) keys
 printing display screen OP-40
 session manager
 See session manager, program function keys
 setting OP-55

Index

- setting 4978 OP-67
- with \$FSEEDIT OP-353
- program output
 - See spooling
- program-type data sets
 - See data set
- prompts, answering OP-41
- protected field
 - defined OP-34

R

- rack enclosure OP-8
- read/write head, diskette OP-12
- recording
 - problems OP-288
- records
 - defined OP-86
 - keeping OP-314
 - sizes, diskettes OP-121, OP-162, OP-320
- redirect spool jobs
 - command syntax OP-363
 - procedure OP-276
- REL subcommand
 - procedure OP-283
 - syntax OP-368
- release
 - \$S REL syntax OP-368
 - command syntax OP-367
 - job queue held job OP-229
 - spool jobs
 - held jobs OP-283
 - kept jobs OP-284
- REMARK syntax OP-390
- rename
 - a data set OP-132
 - a volume OP-131
- resources
 - display spool OP-394
- restart
 - EDX after errors OP-311
 - spool writer OP-280
- restart mode, spooling
 - defined OP-258
 - setting OP-262
- restore
 - control/image store OP-58
 - disk or volume
 - from diskette OP-334
 - from tape OP-338
 - using automatic varyon OP-338
 - system from tape OP-338
 - 4974 to standard character set OP-58
- right margin, changing OP-253
- RPS system, copying EDX data to/from OP-171
- running
 - running programs
 - background OP-219
 - foreground OP-219

- running programs and procedures OP-215

S

- save
 - control store OP-70
 - disk or volume
 - on diskettes OP-319
 - on tape OP-325
 - using automatic initialization OP-325
 - using automatic varyon OP-325
 - system on tape OP-325
 - work data set OP-84
- scan code OP-56
- screens, display terminal
 - See display terminal
- secondary option menu
 - defined OP-73
 - examples OP-81
- send
 - message to another terminal OP-45
- SEND key, operating OP-41
- Series/1 hardware
 - determining what is on system OP-318
 - devices/units OP-8
 - introduction OP-8
 - switching on/off OP-9
- session manager
 - alternate session menu
 - defined OP-72
 - selecting OP-79
 - clearing the screen OP-83
 - custom menus defined OP-73
 - ending OP-84
 - entering parameters OP-82
 - entering user ID OP-79
 - introduction OP-72
 - loading
 - during IPL OP-78
 - with \$L OP-77
 - loading programs
 - using background OP-219
 - using foreground OP-219
 - logging off OP-84
 - logging on OP-79
 - logon menu
 - defined OP-72
 - example OP-79
 - parameter input menu
 - defined OP-73
 - example OP-82
 - primary option menu
 - defined OP-72
 - example OP-79
 - program function keys
 - clearing screen OP-83
 - return to previous screen OP-83
 - return to primary option menu OP-83
 - suspending OP-82

- restarting OP-83
- return to previous screen OP-83
- return to primary option menu OP-83
- secondary option menu
 - defined OP-73
 - example OP-81
- selecting an option OP-80
- signing-on OP-79
- starting a \$JOBUTIL procedure OP-223
- suspending (PF1) OP-82
- user ID defined OP-72
- utilities supported OP-74
- set
 - device offline
 - command syntax OP-373
 - diskette procedure OP-20
 - device online
 - command syntax OP-374
 - diskette procedure OP-18
 - tape procedure OP-128
- shifted mode OP-56
- software errors OP-287
- spooling
 - \$S ALT OP-276, OP-363
 - \$S DALL OP-285, OP-364
 - \$S DE OP-285, OP-365
 - \$S DG OP-285, OP-365
 - \$S DISP OP-275, OP-366
 - \$S HOLD OP-282, OP-366
 - \$S KEEP OP-284, OP-367
 - \$S REL OP-283, OP-368
 - \$S STOP OP-274, OP-368
 - \$S WRES OP-280, OP-369
 - \$S WSTP OP-279, OP-370
 - \$S WSTR OP-278, OP-371
 - \$SPLUT1 display format OP-391
 - allocating spool dataset OP-260
 - alter job printing
 - procedure OP-276
 - syntax OP-363
 - autostart, setting OP-271
 - considerations OP-260
 - control record OP-258
 - copies, changing OP-276
 - data set
 - changing OP-264
 - defined OP-258
 - estimating size OP-395
 - group defined OP-259
 - delete a job
 - command syntax OP-365
 - procedure OP-285
 - delete all jobs
 - command syntax OP-364
 - procedure OP-285
 - delete generic jobs
 - command syntax OP-365
 - procedure OP-285
 - devices, specifying OP-271
 - display formats OP-391
 - display status
 - command syntax OP-366
 - procedure OP-275
 - ending
 - command syntax OP-368
 - procedure OP-274
 - estimating print time OP-395
 - estimating spool data set size OP-395
 - forms alignment
 - responding to prompt OP-286
 - specifying OP-276
 - forms code
 - changing OP-276
 - defined OP-258
 - specifying OP-271
 - forms, altering (\$S ALT) OP-363
 - group size, changing OP-268
 - hold jobs
 - command syntax OP-366
 - procedure OP-282
 - introduction OP-258
 - job display format OP-392
 - jobs, holding OP-282
 - keep jobs
 - command syntax OP-367
 - procedure OP-284
 - maximum active jobs, changing OP-267
 - maximum jobs, changing OP-266
 - operator functions
 - redirecting jobs OP-276
 - release jobs
 - command syntax OP-368
 - procedure OP-283
 - releasing kept jobs
 - command syntax OP-367
 - procedure OP-284
 - resources display format OP-394
 - restart mode
 - defined OP-258
 - setting OP-262
 - restart writer
 - command syntax OP-369
 - procedure OP-280
 - separator page
 - setting option OP-270
 - specifying heading OP-276
 - spool device defined OP-259
 - spool facility defined OP-259
 - spool job defined OP-259
 - spool session defined OP-259
 - spool writer defined OP-259
 - start a writer
 - command syntax OP-371
 - procedure OP-278
 - start mode, setting OP-262
 - starting spool facility OP-274
 - status, displaying OP-275
 - stop a writer
 - command syntax OP-370
 - procedure OP-279

Index

- stop spooling
 - command syntax OP-368
 - procedure OP-274
- terms OP-258
- writer
 - changing OP-276, OP-363
 - restarting OP-280, OP-369
 - starting OP-278, OP-371
 - status OP-275, OP-366
 - stopping OP-279, OP-370
 - writer display format OP-393
- stand-alone dump
 - create diskette OP-207
 - printing OP-293
 - taking OP-290
- standard labels, tape
 - changing OP-118
 - initializing OP-112
- start
 - \$JOBUTIL procedure
 - with \$L OP-222
 - with \$SUBMIT OP-225
 - with the session manager OP-223
 - job queue processor OP-247
 - programs
 - \$L syntax OP-361
 - with \$L OP-217
 - with \$SUBMIT OP-225
 - with the session manager OP-219
- static screen
 - defined OP-42
- status display formats, spool OP-391
- stop
 - job queue processing OP-231, OP-240
 - programs OP-236
 - session manager OP-84
 - spool writer OP-279, OP-370
 - spooling OP-274
- STOP subcommand
 - procedure OP-274
 - syntax OP-368
- storage
 - displaying
 - command syntax OP-360
 - procedure OP-306
 - patching OP-362
 - stand-alone dumps OP-290
- storage devices
 - free space, determining OP-205
 - storing data OP-85
- storing information
 - See data
- submit
 - job
- SUBMIT, job queue job submission
 - See \$SUBMIT utility
- supervisor
 - backing up on diskette OP-319
 - backing up on tape OP-325
 - restarting after error OP-311

- restoring from diskette OP-334
- restoring from tape OP-338
- suspend
 - job queue processing OP-231
- syntax
 - operator commands
 - See operator commands
- system
 - configuration, listing OP-318
 - operations log OP-314
 - records, keeping OP-313
 - restoring from diskette OP-334
 - restoring from tape OP-338
 - saving on diskette OP-319
 - saving on tape OP-325

T

- tab
 - moving the cursor OP-32
- tape
 - allocate a data set OP-127
 - allocating a volume OP-112
 - backup log OP-317
 - change
 - density OP-118
 - label OP-118
 - speed OP-118
 - data set organization OP-86
 - deleting data sets OP-127
 - dumping to printer OP-302
 - initializing OP-112
 - log OP-316
 - preparing for use OP-91
 - restoring system OP-339
 - with automatic varyon OP-338
 - with double buffer OP-338
 - saving system
 - using automatic initialization OP-325
 - with automatic varyon OP-325
 - with double buffer OP-325
 - vary online OP-168
 - volume organization OP-86
- tape drive
- terminal
 - See also display terminal
 - See also printers
 - addresses
 - changing OP-48
 - listing OP-44
 - names, listing OP-44
 - partitions, listing OP-44
 - renaming OP-47
 - screen format OP-51
 - varying offline OP-52
 - varying online OP-54
- text editing utilities
- time and date
 - display

command syntax OP-375
procedure OP-27
set
command syntax OP-372
procedure OP-26
top margin, changing OP-253
transmitting data, using \$TRANS OP-148

U

units, hardware OP-8
unmapped storage
creating stand-alone dump diskette OP-207
dumping OP-290
printing a dump OP-293
unprotected field
defined OP-34
unshifted mode OP-55
upper-case characters
\$FSEDIT CAPS OP-355
specifying OP-56
upshift mode OP-56
utilities, cancelling OP-236
utility
See \$JOBUTIL utility

V

vary
device offline
command syntax OP-373
display terminal OP-52
device online
command syntax OP-374
display terminal OP-54
tape OP-128
version/modification level
determining
with \$D OP-306
with a dump OP-308
of programs OP-308
volume
allocating OP-121
allocating data sets on OP-124
backing up OP-319
compressing OP-142
copying OP-162
data set directory, initializing OP-121
data sets on
See data set directory, listing
defined OP-87
deleting OP-136
determine volume a data set is on OP-185
directory
free space, determining OP-205
label
assigning OP-106
defined OP-88

rename OP-131
volume directory
defined OP-88
initializing OP-106

W

WRES subcommand
procedure OP-280
syntax OP-369
write
\$JOBUTIL procedure OP-241
writer, spooling
See spooling
WSTP subcommand
procedure OP-279
syntax OP-370
WSTR subcommand
procedure OP-278
syntax OP-371

3

3101 Display Terminal
See also display terminal
character/block mode defined OP-42
upper/lowercase characters OP-57

4

4964 Diskette Storage Unit
diskette initialization
See diskette, initialize
opening the door OP-15
4965 Diskette Storage Unit
diskette initialization
See diskette, initialize
opening the door OP-15
4966 Diskette Magazine Unit
diskette initialization
See diskette, initialize
opening the door OP-15
removing diskettes OP-20
using magazines OP-16
4974 Matrix Printer
image store defined OP-56
loading image store OP-58
4975 printer
page formatting OP-255
4978 Display Station
See also display terminal
changing keyboard values OP-56
character/local function code OP-55
control store
changing OP-55
data table defined OP-55

Index

- data table example OP-65
- defined OP-55
- loading OP-58
- saving OP-70
- define key OP-55
- define mode OP-55
- downshift OP-55
- function ID code OP-56
- interrupt code OP-56
- key position OP-56
- lowercase characters OP-56
- PF keys, changing OP-55
- scan code OP-56
- shifted OP-56
- unshift OP-55
- upshift OP-56
- 4979 Display Station
 - See display terminal
- 4980 Display Station
 - See also display terminal
 - control store
 - changing OP-55

- defined OP-55
- loading OP-58
- saving OP-70
- define key OP-55
- define mode OP-55
- downshift OP-55
- function ID code OP-56
- interrupt code OP-56
- key position OP-56
- PF keys, changing OP-55
- scan code OP-56
- shifted OP-56
- unshift OP-55
- upshift OP-56

5

- 5219 Printer
 - changing print density OP-254

IBM Series/1 Event Driven Executive

Publications Order Form

Instructions:

1. Complete the order form, supplying all of the requested information. (Please print or type.)
2. If you are placing the order by phone, dial **1-800-IBM-2468**.
3. If you are mailing your order, fold the order form as indicated, seal with tape, and mail. We pay the postage.

Ship to:

Name:

Address:

City:

State:

Zip:

Bill to:

Customer number:

Name:

Address:

City:

State:

Zip:

Your Purchase Order No.:

Phone: ()

Signature:

Date:

Order:

Description	Order number	Qty.
Reference books:		
Set of the following six books. To order individual copies, use the following order numbers.	SBOF-1627	_____
<i>Communications Guide</i>	SC34-0638	_____
<i>Extended Address Mode and Performance Analyzer User Guide</i>	SC34-0591	_____
<i>Installation and System Generation Guide</i>	SC34-0646	_____
<i>Language Reference</i>	SC34-0643	_____
<i>Library Guide and Common Index</i>	SC34-0645	_____
<i>Messages and Codes</i>	SC34-0636	_____
<i>Operator Commands and Utilities Reference</i>	SC34-0644	_____
Guides and reference cards:		
Set of the following four books and reference cards. To order individual copies, use the following order numbers.	SBOF-1628	_____
<i>Customization Guide</i>	SC34-0635	_____
<i>Event Driven Language Programming Guide</i>	SC34-0637	_____
<i>Operation Guide</i>	SC34-0642	_____
<i>Problem Determination Guide</i>	SC34-0639	_____
<i>Language Reference Card</i>	SX34-0165	_____
<i>Operator Commands and Utilities Reference Card</i>	SX34-0164	_____
<i>Conversion Charts Reference Card</i>	SX34-0163	_____
<i>Reference Card Envelope</i>	SX34-0166	_____
Set of three reference cards and storage envelope. (One set is included with order number SBOF-1627)	SBOF-1629	_____
Binders:		
3-ring easel binder with 1 inch rings	SR30-0324	_____
3-ring easel binder with 2 inch rings	SR30-0327	_____
Standard 3-ring binder with 1 inch rings	SR30-0329	_____
Standard 3-ring binder with 1 1/2 inch rings	SR30-0330	_____
Standard 3-ring binder with 2 inch rings	SR30-0331	_____
Diskette binder (Holds eight 8-inch diskettes.)	SB30-0479	_____

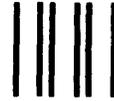
Publications Order Form

Cut or Fold Along Line

Fold and tape

Please Do Not Staple

Fold and tape



NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO. 40 ARMONK, N.Y.



POSTAGE WILL BE PAID BY ADDRESSEE:

IBM Corporation
1 Culver Road
Dayton, New Jersey 08810

Fold and tape

Please Do Not Staple

Fold and tape



International Business Machines Corporation

IBM Series/1 Event Driven Executive
Operation Guide

Order No. SC34-0642-0

READER'S
COMMENT
FORM

This manual is part of a library that serves as a reference source for systems analysts, programmers, and operators of IBM systems. You may use this form to communicate your comments about this publication, its organization, or subject matter, with the understanding that IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you. Your comments will be sent to the author's department for whatever review and action, if any, are deemed appropriate.

Note: *Copies of IBM publications are not stocked at the location to which this form is addressed. Please direct any requests for copies of publications, or for assistance in using your IBM system, to your IBM representative or to the IBM branch office serving your locality.*

Note: Staples can cause problems with automated mail sorting equipment.
Please use pressure sensitive or other gummed tape to seal this form.

Thank you for your cooperation. No postage stamp necessary if mailed in the U.S.A. (Elsewhere, an IBM office or representative will be happy to forward your comments or you may mail directly to the address in the Edition Notice on the back of the title page.)

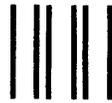
Reader's Comment Form

Cut or Fold Along Line

Fold and tape

Please Do Not Staple

Fold and tape



NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO. 40 ARMONK, N.Y.



POSTAGE WILL BE PAID BY ADDRESSEE:

International Business Machines Corporation
Information Development, Department 28B
P.O. Box 1328
Boca Raton, Florida 33432

Fold and tape

Please Do Not Staple

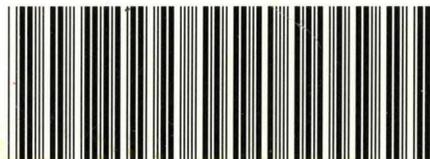
Fold and tape





International Business Machines Corporation

SC34-0642-0



SC34-0642-0
Program Number: 5719-XS5
File Number: S1-40
Printed in U.S.A.