RSX–11M/M–PLUS
RMS–11 Utilities

Order No. AA-L670A-TC

digital software

RSX-11M/M-PLUS
RMS-11 Utilities

Order No. AA-L670A-TC

April 1983

This document describes the RMS-11 utility programs: RMSDES, RMSIFL, RMSCNV, RMSDSP, RMSBCK, and RMSRST. RMSDEF, which has been superseded by RMSDES, is also described in an appendix.

SUPERSESSION/UPDATE INFORMATION:	This is a new manual for this release. It replaces Chapter 9 of the RMS-11 User's Guide, Version 1.8.
OPERATING SYSTEM AND VERSION:	RSX-11M Version 4.1 and RSX-11M-PLUS Version 2.1
SOFTWARE VERSION:	RMS-11 Version 2.0

digital equipment corporation · maynard, massachusetts

First Printing, April 1983

The information in this document is subject to change without notice and should not be construed as a commitment by Digital Equipment Corporation. Digital Equipment Corporation assumes no responsibility for any errors that may appear in this document.

The software described in this document is furnished under a license and may be used or copied only in accordance with the terms of such license.

No responsibility is assumed for the use or reliability of software on equipment that is not supplied by Digital Equipment Corporation or its affiliated companies.

Copyright © 1983 by Digital Equipment Corporation
All Rights Reserved.

Printed in U.S.A.

The postpaid READER'S COMMENTS form on the last page of this document requests the user's critical evaluation to assist in preparing future documentation.

The following are trademarks of Digital Equipment Corporation:

DEC	DIBOL	RSX
DEC/CMS	Edusystem	UNIBUS
DEC/MMS	IAS	VAX
DECnet	MASSBUS	VMS
DECsystem-10	PDP	VT
DECSYSTEM-20	PDT	digital
DECUS	RSTS	
DECwriter		

ZK2169

HOW TO ORDER ADDITIONAL DOCUMENTATION

In Continental USA and Puerto Rico call 800-258-1710

In New Hampshire, Alaska, and Hawaii call 603-884-6660

In Canada call 613-234-7726 (Ottawa-Hull)
800-267-6146 (all other Canadian)

DIRECT MAIL ORDERS (USA & PUERTO RICO)*

Digital Equipment Corporation
P.O. Box CS2008
Nashua, New Hampshire 03061

*Any prepaid order from Puerto Rico must be placed with the local Digital subsidiary (809-754-7575)

DIRECT MAIL ORDERS (CANADA)

Digital Equipment of Canada Ltd.
940 Belfast Road
Ottawa, Ontario K1G 4C2
Attn: A&SG Business Manager

DIRECT MAIL ORDERS (INTERNATIONAL)

Digital Equipment Corporation
A&SG Business Manager
c/o Digital's local subsidiary or approved distributor

Internal orders should be placed through the Software Distribution Center (SDC), Digital Equipment Corporation, Northboro, Massachusetts 01532

CONTENTS

		Page
PREFACE		vii
SUMMARY OF TECHNICAL CHANGES		ix
CHAPTER 1	INTRODUCTION	
CHAPTER 2	RMS-11 FILE DESIGN UTILITY: RMSDES	
2.1	INVOKING AND TERMINATING RMSDES	2-2
2.1.1	Invoking RMSDES to Create a Data File from an Existing File	2-3
2.1.2	Invoking RMSDES Using Indirect Command Files	2-3
2.2	RMSDES COMMANDS	2-4
2.2.1	CLEAR Command	2-6
2.2.2	CREATE Command	2-7
2.2.3	CTRL/Z Command	2-9
2.2.4	ESC Command	2-9
2.2.5	EXIT Command	2-9
2.2.6	GET Command	2-10
2.2.7	HELP Command	2-11
2.2.8	QUIT Command	2-12
2.2.9	SAVE Command	2-13
2.2.10	SET Command	2-13
2.2.11	SHOW Command	2-15
2.2.12	SHOW ID Command	2-16
2.3	USING THE RMSDES BUFFER TO DESIGN FILES	2-16
2.3.1	Setting Attribute Values	2-20
2.3.1.1	Setting Individual Attribute Values	2-21
2.3.1.2	Setting Attribute Values in Sections	2-21
2.3.2	Setting Attribute Values Using External Files	2-24
2.3.2.1	Setting Attribute Values from a Description File	2-24
2.3.2.2	Setting Attribute Values from a Data File	2-25
2.3.3	Clearing Attribute Values	2-25
2.3.3.1	Clearing Individual Attribute Values	2-26
2.3.3.2	Clearing Attribute Values in Sections	2-26
2.3.4	Showing Attribute Values	2-26
2.3.4.1	Showing Individual Attribute Values	2-26
2.3.4.2	Showing Attribute Values in Sections	2-27
2.4	SAVING THE FILE DESIGN	2-27
2.5	CREATING AN EMPTY DATA FILE	2-28
2.6	SECTIONS AND ATTRIBUTES	2-29
2.6.1	System Section	2-30
2.6.2	File Section	2-31
2.6.3	Record Section	2-36
2.6.4	Key Section	2-38
2.6.4.1	Key Segments	2-41
2.6.5	Area Section	2-42

CONTENTS

		Page
2.7	RMSDES EXAMPLES	2-47
2.8	RMSDES MESSAGES	2-52
CHAPTER 3	RMS-11 INDEXED FILE LOAD UTILITY: RMSIFL	
3.1	INVOKING AND TERMINATING RMSIFL	3-2
3.2	RMSIFL COMMAND STRING	3-3
3.3	RMSIFL SWITCHES	3-5
3.3.1	Global Switch	3-5
3.3.2	Output File Switches	3-5
3.3.3	Input File Switches	3-7
3.4	RMSIFL PROCESSING	3-9
3.4.1	Checking the Command String	3-9
3.4.2	Reading and Sorting the Input Records	3-9
3.4.3	Loading Records in the Output File	3-10
3.4.4	Sorting the Alternate Key Temporary File	3-11
3.4.5	Building the Alternate Indexes	3-12
3.4.6	Terminating Processing	3-12
3.5	CALCULATING TEMPORARY FILE WORK SPACE	3-12
3.6	RMSIFL EXAMPLES	3-14
CHAPTER 4	RMS-11 FILE CONVERSION UTILITY: RMSCNV	
4.1	INVOKING AND TERMINATING RMSCNV	4-3
4.2	RMSCNV COMMAND STRING	4-4
4.3	RMSCNV SWITCHES	4-7
4.4	RMSCNV PROCESSING	4-14
4.4.1	Output File Organization and RMSCNV Processing	4-15
4.4.1.1	Sequential Output File Organization	4-15
4.4.1.2	Relative Output File Organization	4-15
4.4.1.3	Indexed Output File Organization	4-15
4.4.2	Input File Organization and RMSCNV Processing	4-16
4.4.2.1	Sequential Input File Organization	4-16
4.4.2.2	Relative Input File Organization	4-16
4.4.2.3	Indexed Input File Organization	4-16
4.5	RMSCNV EXAMPLES	4-16
CHAPTER 5	RMS-11 FILE DISPLAY UTILITY: RMSDSP	
5.1	INVOKING AND TERMINATING RMSDSP	5-2
5.2	RMSDSP COMMAND STRING	5-3
5.3	RMSDSP SWITCHES	5-4
5.4	RMSDSP DISPLAYS	5-5
CHAPTER 6	THE RMS-11 FILE BACK-UP UTILITY: RMSBCK	
6.1	INVOKING AND TERMINATING RMSBCK	6-2
6.2	RMSBCK COMMAND STRING	6-3
6.3	RMSBCK SWITCHES	6-5
6.3.1	Global Switches	6-5
6.3.2	Output File Switches	6-6
6.3.3	Input File Switches	6-8
6.4	RMSBCK PROCESSING	6-8
6.4.1	Explicit and Implicit Input File Specification	6-9
6.4.2	Data Integrity Checking	6-9
6.4.3	Extended Diagnostic Messages	6-10
6.4.4	Summary Listing	6-10
6.5	RMSBCK EXAMPLES	6-10

CONTENTS

Page

CHAPTER 7 RMS-11 FILE RESTORATION UTILITY: RMSRST

7.1 INVOKING AND TERMINATING RMSRST 7-2
7.2 RMSRST COMMAND STRING 7-3
7.3 RMSRST SWITCHES 7-5
7.3.1 Global Switches 7-5
7.3.2 Output File Switches 7-7
7.3.3 Input File Switches 7-8
7.4 RMSRST PROCESSING 7-9
7.4.1 Explicit and Implicit Input File Specification 7-10
7.4.2 Choice of Restoration Account or Volume . . . 7-10
7.4.3 Intersystem File Transfer 7-10
7.4.4 Data Integrity Checking 7-11
7.4.5 Extended Diagnostic Messages 7-11
7.4.6 Summary Listing 7-12
7.5 RMSRST EXAMPLES 7-12

APPENDIX A RMS-11 FILE DEFINITION UTILITY: RMSDEF

A.1 PURPOSE A-1
A.2 EFFECT A-1
A.3 CALL AND TERMINATION A-5
A.3.1 Calling the Utility A-5
A.3.2 Terminating the Utility A-5
A.4 PROCESS A-5
A.4.1 Command File A-5
A.4.2 File Specification A-7
A.4.3 Data Structure A-8
A.4.4 Key Definition A-10
A.4.5 File Structure A-13
A.4.6 Data Allocation A-16
A.4.7 Protection A-18
A.4.8 File Creation A-19
A.5 RMSDEF MESSAGES A-20

APPENDIX B RMSDES AREA DEFINITION

APPENDIX C RMS-11 COMMAND UTILITIES MESSAGES

C.1 RMSCNV, RMSDSP, RMSBCK, AND RMSRST ERROR MESSAGES C-2
C.2 RMSIFL ERROR MESSAGES C-17

APPENDIX D SORT ROUTINE ERROR CODES

INDEX

EXAMPLES

EXAMPLE 5-1 RMSDSP Display Format for Disk Sequential Files . 5-5
5-2 RMSDSP Display Format for Relative Files 5-6
5-3 RMSDSP Display Format for Indexed Files 5-6
5-4 RMSDSP Display Format for Indexed Files (with /FU Specified) 5-6
5-5 RMSDSP Display Format for Magnetic Tape Files . . 5-7
5-6 RMSDSP Display for Back-Up Files 5-7
5-7 RMSDSP Display Format for Back-Up Files (with /FU Specified) 5-8

CONTENTS

Page

5-8 RMSDSP Display Format Using Wildcard Characters . 5-8

FIGURES

FIGURE 4-1 Using RMSCNV with DECnet 4-2
A-1 RMSDEF Processing Flowchart A-4

TABLES

TABLE 2-1 RMSDES Commands 2-4
2-2 RMSDES Section and Attribute Keywords 2-18
3-1 RMSIFL Switches 3-4
4-1 RMSCNV Switches 4-6
5-1 RMSDSP Switches 5-4
6-1 RMSBCK Switches 6-4
7-1 RMSRST Switches 7-4

PREFACE

MANUAL OBJECTIVES

This reference manual describes the utility programs supported by Record Management Services for PDP-11 operating systems (RMS-11).

INTENDED AUDIENCE

This manual is intended for users who either do not require or do not have access to full RMS-11 capabilities: high-level language programmers, and nonprogramming users such as system managers and operators.

STRUCTURE OF THIS DOCUMENT

This document is organized into seven chapters and four appendixes:

- Chapter 1, Introduction
- Chapter 2, RMS-11 File Design Utility: RMSDES
- Chapter 3, RMS-11 Indexed File Load Utility: RMSIFL
- Chapter 4, RMS-11 File Conversion Utility: RMSCNV
- Chapter 5, RMS-11 File Display Utility: RMSDSP
- Chapter 6, RMS-11 File Back-Up Utility: RMSBCK
- Chapter 7, RMS-11 File Restoration Utility: RMSRST
- Appendix A, RMS-11 File Definition Utility: RMSDEF
- Appendix B, RMSDES Area Definition
- Appendix C, RMS-11 Command Utilities Messages
- Appendix D, Sort Routine Error Codes

ASSOCIATED DOCUMENTS

To use the RMS-11 utilities, you should be familiar with the concepts and user information in RSX-11M/M-PLUS RMS-11: An Introduction and the RSX-11M/M-PLUS RMS-11 User's Guide. The introduction presents the major concepts of RMS-11, and the user's guide provides detailed user information for programmers who want to use RMS-11, from either high-level language programs or MACRO-11 programs.

In addition, RSX-11M/M-PLUS RMS-11 Macro Programmer's Guide provides programmer information on RMS-11 operations for MACRO-11 programmers; and the RSX-11M/M-PLUS RMS-11 Mini-Reference Insert summarizes, for easy reference, the RMS-11 utilities and error codes.

CONVENTIONS USED IN THIS DOCUMENT

Convention	Meaning
UPPERCASE WORDS AND LETTERS	Uppercase words and letters used in examples indicate that you should type the word or letter exactly as shown.
lowercase words and letters	Lowercase words and letters used in examples indicate that you are to substitute a word or value of your choice.
quotation marks (") apostrophes (')	The term "quotation marks" refers to double quotation marks ("). The term "apostrophe" refers to a single quotation mark (').
[ ]	Square brackets indicate that the enclosed item is optional.
{ }	Braces are used to enclose lists from which one element is to be chosen.
...	A horizontal ellipsis indicates that the preceding item(s) can be repeated one or more times. For example: file-spec[,file-spec...]
. . .	A vertical ellipsis indicates that not all of the statements in an example or figure are shown.
DES: SHOW FILE ATTRIBUTE: NAME	In examples of commands you enter and system responses, all output lines and prompting characters that the system prints or displays are shown in black letters. All the lines that you type are shown in red letters.
RET	A symbol with a 1- to 3-character abbreviation indicates that you press a key on the terminal.
CTRL/X	The symbol CTRL/x indicates that you must press the key labeled CTRL while you simultaneously press another key (for example, <CTRL/C>, <CTRL/Y>, <CTRL/O>). In examples, this control key sequence is shown as ^x (for example, ^C, ^Y, ^O) because that is how the system echoes control key sequences.

Unless otherwise noted, all numeric values are represented in decimal notation.

Unless otherwise specified, you terminate commands by pressing the RETURN key.

SUMMARY OF TECHNICAL CHANGES

The RMS-11 File Design Utility (RMSDES) is a new utility that allows you to interactively design and create files. RMSDES supersedes the RMS-11 File Definition Utility (RMSDEF).

The RMS-11 File Display Utility (RMSDSP) provides more information about magnetic tape files and about back-up files on disk and on magnetic tape (container files), and provides an optional detailed display for back-up files. The displays also appear in a more readable format, and the size field shows both the allocated and used sizes of the file.

The /DE switch to the RMS-11 Indexed File Load Utility (RMSIFL) has been enhanced to allow specification of a different device for RMSIFL's temporary files.

The /SL switch to the RMS-11 File Conversion Utility (RMSCNV), the File Back-Up Utility (RMSBCK), and the File Restoration Utility (RMSRST) has been modified to append summary listings of processing to a single file if that file exists.

Five switches have been added to the RMS-11 File Conversion Utility (RMSCNV): the /CA switch, the /EO switch, the /IM switch, the /KN switch, and the /ML switch. The /CA switch creates an output file with the attributes of an existing file; the /EO switch changes ASCII stream files that end with a CTRL/Z character to end with null characters; the /IM switch reads and writes records in block mode; the /KN switch reads an indexed file using a key name; and the /ML switch explicitly sets the limit of memory allocation. The /PD switch no longer requires a number sign (#) before an octal padding character or quotation marks around an ASCII padding character unless it is lowercase, a space, or a tab.

In addition, RMSCNV file conversion operations can now specify files on remote nodes. On systems with DECnet network capabilities and on which RMSCNV has Data Access Protocol (DAP) support, RMSCNV file conversions can take place not only locally, but over the network as well. You can convert files between a local and a remote node, or between two remote nodes.

One switch has been added to the RMS-11 File Restoration Utility (RMSRST): /CV. The /CV switch converts file version numbers from octal to decimal for files created and backed up on RSX-11M/M-PLUS systems and restored to VAX/VMS systems; and from decimal to octal for files created and backed up on VAX/VMS systems and restored to RSX-11M/M-PLUS systems.

All utilities now display information in lowercase letters, and the error message format for all utilities (except RMSDEF) has been changed (see Appendix C).

CHAPTER 1

INTRODUCTION

RMS-11 utilities allow you to define, manipulate, and maintain RMS-11 files. The utilities are provided as independent tasks and are designed for users -- for example, high-level language programmers, system managers and operators -- who either do not require or do not have access to the full RMS-11 capabilities.

The RMS-11 utilities are:

- RMS-11 File Design Utility (RMSDES)

  RMSDES allows you to design and create indexed, sequential, and relative files. You can either specify the file attributes interactively or read in the attributes of an existing, external data file that you want to re-create with little or no modification.

- RMS-11 Indexed File Load Utility (RMSIFL)

  RMSIFL reads records from any type of RMS-11 file and loads them into an indexed file.

- RMS-11 File Conversion Utility (RMSCNV)

  RMSCNV reads records from an RMS-11 file of any type and loads them into another RMS-11 file of any type.

- RMS-11 File Display Utility (RMSDSP)

  RMSDSP displays a concise description of any RMS-11 file, including RMS-11 back-up files on disk or on magnetic tape (container files).

- RMS-11 File Back-Up Utility (RMSBCK)

  RMSBCK transfers the contents of any RMS-11 disk file to another disk or to magnetic tape to maintain the file should the original file be lost or damaged.

- RMS-11 File Restoration Utility (RMSRST)

  RMSRST transfers, to disk, files that were backed up on magnetic tape or disk using RMSBCK.

- RMS-11 File Definition Utility (RMSDEF)

  RMSDEF allows you to define the attributes of a file. Although this utility has been superseded by the RMSDES utility, it is available to users and is documented in Appendix A.

Help Text and Error Messages. The interactive utilities (RMSDES and RMSDEF) provide error messages and help text for each prompt for input. The command utilities (RMSIFL, RMSCNV, RMSDSP, RMSBCK, and RMSRST) each have a single help text and share a set of common error messages. The command utilities' error messages are documented in Appendix C. RMSDES and RMSDEF error messages are documented in Chapter 2 and Appendix A, respectively.

Intersystem File Transfer. On systems with DECnet network capabilities and on which RMSCNV has Data Access Protocol (DAP) support, RMSCNV file conversions can take place not only locally, but over the network as well. You can convert files between a local and a remote node, or between two remote nodes.

For example, working on an RSX-11M/M-PLUS system, you can convert files to which you have access between your system and another RSX-11M/M-PLUS system. Or, you can convert files from your RSX-11M/M-PLUS system to either a RSTS/E or VAX/VMS system, or between a RSTS/E system and a VAX/VMS system.

See the RSX-11M/M-PLUS RMS-11 User's Guide for more information on RMS-11 remote access capabilities. See Chapter 4 for more information on RMSCNV.

You can also use RMSBCK and RMSRST to transport files between different systems. RMSBCK preserves the RMS-11 file attributes when you back up files to disk or magnetic tape; the file can thus be transported and restored, using RMSRST, to another operating system. See Chapter 7 for more information on restoring files to different operating systems.

Note that you can use RMSDSP to display an RMS-11 back-up file's attributes and determine which operating system the file was created and backed up on. See Chapter 5 for more information on RMSDSP.

CHAPTER 2

RMS-11 FILE DESIGN UTILITY: RMSDES

The RMS-11 File Design Utility (RMSDES) is an interactive utility that allows you to design and create RMS-11 files to contain data records.

You design a file by defining the file's attributes. When you create a file, a file is built that has those attributes and can store data records as required by your program(s).

To design a file using RMSDES, you issue RMSDES commands to set, reset, clear, and display file attribute values in the RMSDES design buffer. This is called a design session. The attributes consist of the standard RMS-11 attributes, such as file organization and record format, plus some additional, informational attributes required by RMSDES, such as target system.

The design buffer is an RMSDES work space that you use to design your files. It is divided into five sections: system, file, record, key, and area. Each section defines a group of file attributes. For example, in the file section, you define attributes such as file name, organization, and owner; in the record section, you define attributes such as record size, format, and carriage control.

The functional grouping of attributes in the design buffer makes it easy to define, change, or display any attribute or group of attributes. See Section 2.6 for a complete description of each buffer section and the file attributes defined in each section.

You gain access to the design buffer and initiate a design session when you invoke RMSDES. At the beginning of a design session, the file attributes are initialized to RMS-11 default values. You can set file attributes either interactively from your terminal by using the SET command, or from an external file by using the GET command.

If you are designing your file interactively, you can either explicitly set the attributes to values that you specify, or you can accept the default values. If you are designing an indexed file, you can either explicitly define areas or allow RMSDES to define them by default.

If you use the SET command to design a file interactively, you will be prompted for attribute values. RMSDES prompting is conditional; you are prompted only for attributes that pertain to your file design. For example, you will be prompted for cluster size only if you have set the target system to RSTS/E.

If you are designing a file using an external file, RMSDES sets the attribute values in the design buffer to match those in the external file. This is useful if you want to duplicate an existing file's design.

You can display file attributes by using the SHOW command. You can change attribute values at any time during the design session by using either the SET command (to set a new value) or the CLEAR command (to restore the default value).

You can save the file design in a description file by using the SAVE command, or you can create an empty data file by using the CREATE command.

Throughout a design session, you can get online assistance by typing a question mark (?) or the HELP command.

Sections 2.1 and 2.2 describe how to invoke and terminate RMSDES and how to use the RMSDES commands. Section 2.3 describes how to use the design buffer to design a file. Sections 2.4 and 2.5 describe saving designs and creating empty data files. Section 2.6 describes the buffer sections and file attributes.

This chapter also provides examples (Section 2.7) and documents the RMSDES messages (Section 2.8).

2.1 INVOKING AND TERMINATING RMSDES

RMSDES can be either installed or uninstalled. If RMSDES is installed, invoke it with the command:

```
DES
```

If RMSDES is uninstalled on RSX-11M/M-PLUS systems, invoke it with the MCR command:

```
RUN $RMSDES
```

RMSDES will respond with the prompt:

```
DES:
```

You can now type any of the RMSDES commands.

You can install RSMDES with another name as follows:

```
INS $RMSDES/TASK=...xxx
```

where xxx is a 3-character name that you chose for the utility. The utility will then prompt:

```
xxx>
```

To terminate RMSDES, type one of the following commands in response to the DES: prompt:

```
QUIT

<CTRL/Z>

EXIT file-spec
```

The QUIT and CTRL/Z commands terminate RMSDES, and the system prompt is returned. Unless you have saved the file design or created an empty data file, the information in your design buffer will be lost.

The EXIT command saves the file design in the specified file, terminates RMSDES, and the system prompt is returned.

Section 2.2 describes the RMSDES commands and their formats.

2.1.1 Invoking RMSDES to Create a Data File from an Existing File

You can also invoke RMSDES to create an empty data file from an existing description or data file by typing the following in response to the system prompt:

```
DES filename[.typ] [kind]
```

The default file type is .DES. The file kind is either DES (for a description file) or DAT (for a data file). The default kind is DES and need not be specified. After this command is executed, the system prompt will be returned.

The file that is created will have the same attributes as the external file. If the external file is a description file, the newly created file will have the same file name and type that are set in the description file. If no file name and type are specified in the description file, the file will be created as FILE.DAT.

If the external file is a data file, the new data file will have the same file name and type as the existing file and a version number 1 higher than the existing data file.

See also Sections 2.3.2 and 2.5 for more information on using external files and on creating empty data files.

2.1.2 Invoking RMSDES Using Indirect Command Files

You can execute series of RMSDES commands by means of indirect command files. The commands in an indirect command file are executed when an at sign (@) followed by the file's name is specified to the utility. The default file type of .CMD is assumed if none is provided.

Installed RMSDES will accept indirect command files either from the system prompt or from the utility prompt. For example, to execute the commands in the indirect command file DESCOM.CMD, you could type:

```
DES @DESCOM
   or
DES: @DESCOM
```

Uninstalled RMSDES will accept indirect command files only from the utility prompt. For example:

```
DES: @DESCOM
```

RMSDES will access the indirect command file and execute the commands. If you executed the indirect command file from the system prompt, the system prompt will be returned.

If you executed the indirect command file from within RMSDES, the DES: prompt will be returned, so you can continue the design session, if desired. Note, however, that any attributes that were set or reset as a result of executing the indirect command file from within RMSDES will remain set in the design buffer.

You can issue a SHOW ALL command after indirect command file execution to determine the attribute settings in the design buffer. You might also want to include a CLEAR ALL command at the end of the indirect command file so that all the attributes are reset to their initial default values at the end of indirect command file execution.

You cannot nest indirect command files run from RMSDES; that is, an indirect command file cannot contain a command calling another indirect command file.

You can include comments, using semicolons (;) or exclamation points (!), in indirect command files run from either the system prompt or the utility prompt.

2.2 RMSDES COMMANDS

Table 2-1 summarizes the RMSDES commands, their formats, and their functions. You can abbreviate commands as long as the characters typed uniquely identify the command.

The sections that follow Table 2-1 describe the commands and their formats, in alphabetical order.

Table 2-1: RMSDES Commands

Command	Format and Function
CLEAR	CLEAR ALL Restores all attribute values in all sections to their default values. CLEAR section ALL Restores all attribute values in the specified section to their default values. CLEAR section attribute Restores the specified attribute value in the specified section to its default value.
CREATE	CREATE [filename[.typ]] Creates an empty data file that has the attribute values specified in the design buffer. For indexed files in which areas are not defined, prompts for whether areas are to be defined by default. If you do not specify a file name and type, the file will have those specified in the design buffer. If you did not specify a file name and type in the design buffer, the file will be created as FILE.DAT.
CTRL/Z	<CTRL/Z> Terminates RMSDES without saving the design or creating an empty data file.

(continued on next page)

Table 2-1 (Cont.): RMSDES Commands

Command	Format and Function
ESC	<ESC> In response to any prompt, returns the DES: prompt and preserves all attribute values in the design buffer.
EXIT	EXIT filename[.typ] Stores the file design in the description file specified in the command line and terminates RMSDES. The default file type is .DES.
GET	GET filename[.typ] [kind] Reads the file design specified in a description file, and sets the appropriate attribute values in the design buffer. Reads the attribute values of a data file, and sets the appropriate attribute values in the design buffer. The default file kind is DES. If the file is a data file, kind must be specified: DAT.
HELP[1]	HELP Lists all available help topics and gives instructions for displaying the text. HELP COMMANDS Lists the valid commands for which help is available. HELP command Displays help text for the specified command. HELP SECTIONS Lists all available help topics for all sections and gives instructions for displaying the text. HELP section Lists all available help topics for all attributes in the specified section. HELP section attribute Displays help text for the specified attribute in the specified section. ? Displays help text for the section, attribute, or value for which you are being prompted.

(continued on next page)

1. For any form of the HELP command, you can type a question mark (?) instead of HELP. Note that you can type the HELP form only in response to the DES: prompt.

Table 2-1 (Cont.): RMSDES Commands

Command	Format and Function
QUIT	QUIT Terminates RMSDES, without storing the design or creating an empty data file.
SAVE	SAVE filename[.typ] Stores the file design in the description file specified in the command line. The default file type is .DES.
SET	SET ALL Prompts for setting all attribute values in all sections. For indexed files in which areas are not defined, prompts for whether areas are to be defined by default. SET section ALL Prompts for setting all attribute values in the specified section. SET section attribute value Sets the specified attribute value in the specified section.
SHOW	SHOW ALL Displays all attribute values in all sections. SHOW section ALL Displays all attribute values in the specified section. SHOW section attribute Displays the specified attribute value in the specified section.
SHOW ID	SHOW ID Identifies the current version and patch level of RMSDES.

2.2.1 CLEAR Command

The CLEAR command restores attribute values in the design buffer to their initial default values. You can clear attribute values for all sections, for one section, or for one attribute.

You should clear all sections after you have saved or created a file; the buffer will then be ready to accept a design for another file.

Format

```
CLEAR { ALL              }
      { section ALL      }
      { section attribute }
```

ALL

Restores all attribute values in all sections to their initial default values.

section ALL

Restores all attribute values in the specified section to their default values. Section must be one of the section keywords listed in Table 2-2 (Section 2.3). Note that for this form of the command, you must type "ALL" in full ("AL" and "A" are legal abbreviations for the Allocate attribute).

section attribute

Restores the specified attribute value in the specified section to its initial default value. Section and attribute each must be one of the keywords listed in Table 2-2 (Section 2.3).

After RMSDES executes a CLEAR command, the DES: prompt is returned. If you omit any part of the command string, you will be prompted for it.

Examples

```
DES: CLEAR
  SECTION: FILE
  ATTRIBUTE: ORGANIZATION
DES:
```

RMSDES prompts for the omitted section and attribute keywords. When the keyword is supplied, the default file organization (sequential) is restored.

```
DES: CLEAR KEY 2 ALL
DES:
```

This command clears all the attribute values in key 2. Note that if you have defined multiple keys (or areas), and you issue the CLEAR command, the keys (or areas) will be renumbered and you will have one fewer keys (or areas).

For example, if you have defined keys 0 through 4 and you clear key 2, key 3 will become key 2, and key 4 will become key 3. Four keys, not five, will now be defined. If you want to change the attributes in a key, you should use the SET command to replace the previous attribute values with the new ones, without altering key numbers.

2.2.2 CREATE Command

The CREATE command creates an empty data file that has the attribute values specified in the design buffer. The empty data file is capable of containing data records for use by your programs.

If you create an indexed file for which you defined areas and/or keys, the areas and/or keys must be numbered in sequence, starting with 0. RMSDES will create only those areas and/or keys that are numbered sequentially, starting with 0. For example, if you have defined areas 0, 1, and 3, only areas 0 and 1 will be created when you issue the CREATE command. No error will be returned.

You can allow RMSDES to define areas in an indexed file by default. If you have not defined areas when you issue the CREATE command, you will be prompted as to whether you want RMSDES to define areas by default:

```
Areas defaulted:
```

If you respond YES or press RETURN (YES is the default), RMSDES will prompt for additional information and define areas based on your responses and on the number of keys you specified in the key section. See also Section 2.6.5.

Format

```
CREATE [filename[.typ]]
```

filename[.typ]

Names an empty data file that has the attribute values specified in the design buffer. If specified, the file name and type in the command string will supersede those specified in the design buffer. The default file type is .DAT.

If you do not specify a file name, the data file will have the file name and type specified in the design buffer. If a file name and type are not specified in either the design buffer or the command line, the file will be created as FILE.DAT.

After RMSDES executes a CREATE command, the DES: prompt is returned. If you want to design another file, you should issue a CLEAR ALL command to restore the attribute values in the design buffer to their defaults.

Examples

```
DES:  CREATE
DES:
```

This command creates an empty data file, using the file name and type specified in the design buffer.

```
DES:  CREATE PER.IDX
   Areas defaulted:  RET
   Bucket_Size (2):  RET
DES:
```

This command creates an empty indexed data file called PER.IDX. Areas were not defined in the design buffer, so the user is prompted "Areas defaulted:" and presses RETURN in response. RMSDES then prompts for bucket size and the user again takes the default. RMSDES will define the areas based on a bucket size of 2 and the number of keys specified in the design buffer.

2.2.3 CTRL/Z Command

The CTRL/Z command terminates RMSDES without saving the file design or creating an empty data file. All attribute values in the design buffer are lost.

Format

```
<CTRL/Z>
```

Examples

```
DES:  SET RECORD ALL
Starting RECORD section
    Size (0):  1000
    Format (Var):  CTRL/Z
```

The CTRL/Z command, typed in response to the format attribute prompt, terminates RMSDES. The system prompt is returned and all attribute values in the design buffer are lost.

2.2.4 ESC Command

The ESC command, entered in response to any RMSDES prompt, returns the DES: prompt and preserves all attribute values in the design buffer.

Format

```
<ESC>
```

The ESC command is especially useful when you want to interrupt an interactive design session. For example, you may want to make changes to previously set attributes or to resume your design session later. Typing ESC will return the DES: prompt and preserve all attribute values in the design buffer. You can then enter any RMSDES command in response to the DES: prompt.

Examples

```
DES:  SET RECORD ALL
Starting RECORD section
    Size (0):  1000
    Format (Var):  ESC
DES:
```

The ESC command, typed in response to the format attribute prompt, returns the DES: prompt. All attribute values in the design buffer are saved.

2.2.5 EXIT Command

The EXIT command stores your file design in the description file specified in the command string. Then, it terminates RMSDES, and the system prompt is returned.

Format

```
EXIT filename[.typ]
```

filename[.typ]

Names the description file in which the file design will be stored. The default file type is .DES.

Examples

```
DES: EXIT
    File_Name:
```

If you omit the file specification, you will be prompted for it.

```
DES: EXIT PERS
```

This command stores your file design in a description file called PERS.DES and terminates RMSDES. Because the file type, .DES, is the default, you can omit it from the command line.

2.2.6 GET Command

The GET command sets attribute values in the design buffer based on the information it reads from an external file; that is, a description file or a data file. A description file contains a file design. A data file contains, or is capable of containing, data records. The GET command:

- Reads the file design specified in a description file and sets the corresponding attribute values in the design buffer. (See also Section 2.3.2.1.)
- Reads the attribute values of a data file and sets the corresponding attribute values in the design buffer. (See also Section 2.3.2.2.)

When you issue a GET command, RMSDES first initializes the file attributes in the design buffer to their default values. Thus, any attributes that you may have set before issuing the GET command are cleared. Then, RMSDES sets the attribute values in the design buffer to match those of the external file.

Note, however, that if you specify an indexed data file, RMSDES cannot access the initial allocation for the areas: the allocation attribute for each area will be set to the number of blocks that remain available in each area of the data file, not to the initial allocation for each area in the data file. You can use the SHOW command to display the setting and then use the SET command to reset the allocation attribute as necessary.

If the external file is a data file, the file name and type in the design buffer will be set to those of the existing file. If you issue a CREATE command, the new data file will have the same file name and type as the existing data file and a version number 1 higher than the existing data file.

Format

GET filename[.typ] [kind]

filename[.typ] [kind]

Reads the file attributes from a description file and sets the corresponding attribute values in the design buffer. The default file type is .DES. If the description file was created using RMSDES, you need not specify kind (DES is the default kind).

GET also reads the attribute values of a data file and sets the corresponding attribute values in the design buffer. You must specify kind as DAT.

After RMSDES executes a GET command, the DES: prompt is returned.

Examples

```
DES:  GET FOO
DES:
```

This command reads the file attributes in FOO.DES, a description file created using RMSDES, and sets the corresponding attribute values in the design buffer. Because the file type, .DES, and kind, DES, are the defaults, they can be omitted from the command line.

```
DES:  GET
    File_Name:  FOO.BAR DAT
DES:
```

If you omit the file specification, you will be prompted for it. This command reads the file attributes in FOO.BAR, a data file, and sets the corresponding attribute values in the design buffer.

2.2.7 HELP Command

The HELP command displays help text. You can type the HELP form of the command only in response to the DES: prompt. You must type the ? form in response to any other prompt. In addition, for any form of the command, you can type a question mark (?) instead of HELP.

Formats

```
        { COMMANDS          }
        { command           }
        { SECTIONS          }
HELP    { section           }
        { section attribute }
```

?

If you type HELP in response to the DES: prompt, all help topics and instructions for displaying the text will be given.

COMMANDS

Lists the valid commands for which help is available.

command

Specifies a command for which help text is to be displayed. Command must be one of the command keywords listed in Table 2-1 (Section 2.2).

SECTIONS

Lists all section help topics and gives instructions for displaying the text.

section

Specifies a buffer section for which help text is to be displayed. Section must be one of the section keywords listed in Table 2-2 (Section 2.3).

section attribute

Specifies an attribute and section for which help text is to be displayed. Section and attribute must each be one of the keywords listed in Table 2-2 (Section 2.3).

?

Displays help text for the section, attribute, or value for which you are being prompted.

After RMSDES executes any form of the HELP command, the prompt from which you issued the command is returned.

Examples

```
DES: HELP SYSTEM

  Section     Attribute Keyword                    Default
  Keyword         and Variable

  System   Target argument

             argument must be one of:
             RSX                                  User's system
             RSTS

           Source[1]                              User's system

           File Placement[2]  logical             NO
```

In response to the HELP SYSTEM command, RMSDES displays the system attributes and the default settings for those attributes.

```
DES: HELP FILE NAME
```

Name string

Specifies the name of the file. For string, type a file specification.

In response to the HELP FILE NAME command, RMSDES displays information on the Name attribute.

2.2.8 QUIT Command

The QUIT command terminates RMSDES without creating a description file or an empty data file. The attribute values in the design buffer are lost.

---

1. Not user-settable; RMSDES automatically notes the user's source system.

2. Informational attribute.

Format

```
QUIT
```

After RMSDES executes the QUIT command, the system prompt is returned.

2.2.9 SAVE Command

The SAVE command stores the file design in a description file specified in the command string.

Format

```
SAVE filename[.typ]
```

filename[.typ]

Names the description file in which the file design will be saved. The default file type is .DES.

If you omit any part of the command string, you will be prompted for it. After RMSDES executes the SAVE command, the DES: prompt is returned. If you want to design another file, you should issue a CLEAR ALL command to restore the attribute values in the design buffer to their defaults.

Examples

```
DES:  SAVE
    File_Name:  FOO.TYP
```

If you omit the file specification, you will be prompted for it.

```
DES:  SAVE FOO
DES:
```

This command stores the file design in the description file FOO.DES. Because the file type, .DES, is the default, it can be omitted from the command line.

2.2.10 SET Command

The SET command sets attribute values in the design buffer. You can set an individual attribute value or you can set attribute values for all sections or for one section. Note that prompting for attributes will be conditional based on the target system, file organization, and record format that you specify (either explicitly or by default). For example, you will be prompted for cluster size only if you have specified the target system as RSTS/E.

Format

```
SET { ALL                      }
    { section ALL              }
    { section attribute value  }
```

ALL

Specifies that you want to be prompted for all attribute values in all sections.

When you reach the area section, you will be prompted as to whether you want RMSDES to define areas by default:

Areas defaulted:

If you respond YES or press RETURN (YES is the default), RMSDES will prompt for additional information and define areas based on your responses and on the number of keys you specified in the key section (see Section 2.6.4). Appendix B describes how RMSDES defines areas by default.

section ALL

Specifies that you want to be prompted for all attribute values in the specified section. Section must be one of the section keywords listed in Table 2-2 (Section 2.3). Note that if you use this form of the command, you must type "ALL" in full ("AL" and "A" are legal abbreviations for the Allocate attribute).

For the key and area sections, you will be asked for the number of keys or areas you want to define. After you specify the number of keys or areas, you will be prompted for the attribute values of all the keys or areas you specified. See Sections 2.6.4 and 2.6.5.

section attribute value

Specifies that you are setting the specified attribute value in the specified section. Section and attribute each must be keywords and value must be one of the variables listed in Table 2-2 (Section 2.3).

After RMSDES executes the SET command, the DES: prompt is returned. If you omit any part of the command string, you will be prompted for it.

Examples

```
DES:  SET FILE NAME
  VALUE:
```

If you omit the section or attribute keyword or the value, you will be prompted for it.

```
DES:  SET FILE NAME FOO.BAR
DES:
```

This command names the file FOO.BAR.

```
DES:  SET ALL
```

In response to this command, you will be prompted for all attributes in all sections. Note that the prompting will be conditional; for example, if you type RSX in response to the target prompt, you will not be prompted for a cluster size value, which is required only if the target system is RSTS/E (see Section 2.6).

```
DES:  SET FILE ALL
```

In response to this command, you will prompted for all attributes in the file section. Note that the prompting for file attributes will be conditional based on the file organization and format that you specify. For example, if you specify that the file organization is indexed, you will not be prompted for a magnetic tape block size value (see Section 2.6).

2.2.11 SHOW Command

The SHOW command displays attribute values. You can display an individual attribute value or you can display attribute values for all sections or for one section. Note that the display will be conditional; that is, only the attributes for which you were prompted by the SET command will be displayed (see Section 2.2.10). The display also indicates whether attribute values were set explicitly or by default.

Format

```
SHOW  { ALL                }
      { section ALL        }
      { section attribute  }
```

ALL

Displays all attribute values in all sections.

section ALL

Displays all attribute values in the specified section. Section must be one of the section keywords listed in Table 2-2 (Section 2.3). Note that if you use this form of the command, you must type "ALL" in full ("AL" and "A" are legal abbreviations for the Allocate attribute).

section attribute

Displays the specified attribute value. Section and attribute must be keywords listed in Table 2-2 (Section 2.3).

After RMSDES executes the SHOW command, the DES: prompt is returned. If you omit any part of the command string, you will be prompted for it.

Examples

```
DES:  SHOW FILE
  ATTRIBUTE:
```

If you omit the section, attribute, or value keyword, you will be prompted for it.

```
DES:  SHOW FILE NAME
  NAME            FOO.BAR    (Explicit)
DES:
```

This command displays the name of the file, FOO.BAR, and indicates that the file name was set explicitly.

```
DES: SET FILE NAME INFILE.DAT
DES: SET FILE ORGANIZATION INDEXED
DES: SHOW FILE ALL
FILE
  Name                      INFILE.DAT  (Explicit)
  Organization              INDEXED  (Explicit)
  Allocation                0
  Extension                 0
  Bucket_Size               1
  Protection
  Owner                     [303,24]
  Max_Record_Number         0
  Contiguous                NO
  Supersede                 NO
```

This command displays attribute values in the file section.

2.2.12 SHOW ID Command

The SHOW ID command displays the current version of the RMSDES utility. You can type SHOW ID with the DES command (in response to the system prompt), or in response to the DES: prompt (after you have invoked RMSDES).

The last two digits of the version number are the patch level of the utility.

Format

```
SHOW ID
```

After RMSDES executes the SHOW ID command, the prompt from which you issued the command will be returned.

Examples

```
DES SHOW ID
%DES-I-ID, Version 2.000
```

Typed in response to the system prompt, SHOW ID displays the RMSDES version number. The last two digits (00 in this example) are the patch level. The system prompt will be returned after the version number is displayed.

```
DES: SHOW ID
%DES-I-ID, Version 2.000
DES:
```

In this example, SHOW ID is typed in response to the DES: prompt, which is returned after the command is executed.

2.3 USING THE RMSDES BUFFER TO DESIGN FILES

The RMSDES design buffer is a work space that you use to design files and to fine tune your designs. The design buffer is divided into five sections: system, file, record, key, and area. Each of these sections contains file attributes that can be set to specific values.

Table 2-2 lists the attributes grouped within each design buffer section. Section 2.6 describes the sections and attributes of the design buffer in greater detail.

When you begin a design session, all design buffer attributes are initialized to default values. Most of the attributes in the design buffer are RMS-11 file attributes and are initialized to RMS-11 default values. Some attributes, however, such as target system, are informational attributes required by the RMSDES utility.

Table 2-2 lists the defaults assigned to each attribute when you begin a design session and indicates whether the attribute is informational. In designing a file, you can set these attributes to specific values that you supply, or you can accept the default values (by pressing RETURN).

You can set attribute values interactively by using the SET command. You will be prompted conditionally so that you set only the attributes that pertain to your file design. For example, if you specify RSX as your target system, you will not be prompted for cluster size (which is required only if the target system is RSTS/E).

You can also set attribute values by using the GET command to load into the design buffer either (1) a file design from a description file or (2) the attribute values of a data file. The attribute values that you specify, either interactively or from external description or data files, replace the initial RMSDES default values. Section 2.3.1 describes setting attribute values interactively. Section 2.3.2 describes setting attribute values using external files.

Attribute values that are not replaced by user-set values (by use of either the SET or GET command) retain default values. You can also allow RMSDES to automatically define areas in an indexed file.

If you issue a SET ALL command for indexed files, RMSDES will prompt you when you reach the area section. You will also be prompted if you issue a CREATE command but have defined no areas for an indexed file:

```
Areas defaulted:
```

You can respond YES or NO:

- **To define areas explicitly,** type NO. If you issued a SET ALL or SET AREA ALL command, RMSDES will prompt you:

  ```
  Number of areas:
  ```

  Type a decimal number to specify the number of areas you want to define. RMSDES will prompt you for each attribute in every area you are defining. Areas will be numbered sequentially starting with 0.

  If you define no areas but issue a CREATE command, the file will be created with a single area (the RMS-11 default).

- **To define areas by default,** type YES or press RETURN (YES is the default). RMSDES will prompt you:

  ```
  Bucket_Size (2):
  ```

  Type a decimal number to indicate bucket size (the default is 2).

  When you issue a CREATE command, RMSDES will define areas by default based on this bucket size and the number of keys you specified in the key section.

See also Section 2.6.5 for more information on allowing RMSDES to define areas by default.

In addition, you can clear and display attribute values during a design session by issuing a CLEAR or SHOW command. See Sections 2.3.3 and 2.3.4.

At the end of the design session, to capture the file that you have designed in the design buffer, you must save the design with the SAVE or EXIT command (see Section 2.4) or create the data file with the CREATE command (see Section 2.5). If you terminate the design session with the QUIT or CTRL/Z command, the information in the buffer will be lost.

Table 2-2: RMSDES Section and Attribute Keywords

Section Keyword	Attribute Keyword and Variable	Default
System	TARGET[1] argument	
	argument must be one of:	
	RSX RSTS	User's system
	SOURCE[1] [2]	User's system
	FILE PLACEMENT[1] logical	NO
File	NAME string	FILE.DAT
	ORGANIZATION argument	
	argument must be one of:	
	SEQUENTIAL RELATIVE INDEXED	SEQUENTIAL
	CLUSTER_SIZE number (RSTS/E only)	0 blocks
	ALLOCATION number	0 blocks
	EXTENSION number	0 blocks
	BUCKET_SIZE number	1 block
	PROTECTION string	System protection
	OWNER string (RSX-11M/M-PLUS only)	User's UIC

(continued on next page)

1. Informational attribute.

2. Not user settable; RMSDES automatically notes the user's source system.

Table 2-2 (Cont.): RMSDES Section and Attribute Keywords

Section Keyword	Attribute Keyword and Variable	Default
File	MAGTAPE_BLOCK_SIZE number	512 bytes
	REWIND_MAGTAPE logical	NO
	MAX_RECORD_NUMBER number	0 records
	CONTIGUOUS logical	NO
	SUPERSEDE logical	NO
Record	SIZE number	0 bytes
	FORMAT argument	
	argument must be one of:	
	VARIABLE STREAM FIXED VFC	VARIABLE
	CONTROL_FIELD_SIZE number	2 bytes
	BLOCK_SPAN logical	YES
	CARRIAGE_CONTROL argument	
	argument must be one of:	
	CARRIAGE_RETURN FORTRAN NONE	CARRIAGE_RETURN
Key n[1]	NAME string	No name
	TYPE argument	
	argument must be one of:	
	STRING BIN2 BIN4 INT2 INT4 DECIMAL	STRING
	NULL_KEY logical	NO

(continued on next page)

1. You must specify a number (n) for each key, key segment, and/or area you define.

Table 2-2 (Cont.): RMSDES Section and Attribute Keywords

Section Keyword	Attribute Keyword and Variable	Default
Key n[1]	NULL_VALUE argument	
	argument must be one of:	
	An ASCII character A decimal number	' ' (space)
	DUPLICATES logical	NO (primary key) YES (alternate key)
	SEGn_POSITION[1] number	Byte 0
	SEGn_LENGTH[1] number	0 bytes
	SEG0_POSITION number	Byte 0
	CHANGES logical	YES (alternate key)
	DATA_FILL number	100 %
	DATA_AREA number	Area 0
	INDEX_FILL number	100 %
	LEVEL1_INDEX_AREA number	Area 0
	INDEX_AREA number	Area 0
Area n[1]	ALLOCATION number	0 blocks
	EXTENSION number	0 blocks
	BUCKET_SIZE number	0 blocks
	CONTIGUOUS logical	NO
	POSITION argument	
	argument must be one of:	
	NONE	NONE
	VIRTUAL number (RSX-11M/M-PLUS only)	
	LOGICAL number	
	EXACT_POSITIONING logical (RSX-11M/M-PLUS only)	NO

1. You must specify a number (n) for each key, key segment, and/or area you define.

2.3.1 Setting Attribute Values

You use the SET command (Section 2.2.10) to specify attribute values in the design buffer individually, in one section, or in all sections.

2.3.1.1 Setting Individual Attribute Values - When you issue the SET command for an individual attribute, the command is executed to set the specified attribute and the DES: prompt is returned. For example:

```
DES:  SET FILE ORGANIZATION INDEXED
DES:
```

You can accept the default for an attribute by omitting the value from the command string and pressing RETURN in response to the VALUE: prompt. The following examples illustrate how to set the individual attribute values for a 1-segment primary key (key 0).

```
DES:  SET KEY 0 TYPE
  VALUE:  (RET)
```

Pressing RETURN in response to the VALUE: prompt specifies the default for TYPE (STRING).

```
DES:  SET KEY 0 SEG0_POSITION 0
DES:
```

The SET command specifies that the first byte of the primary key will begin at byte 0 of the record.

```
DES:  SET KEY 0 SEG0_LENGTH 5
DES:
```

The SET command specifies that the primary key will be 5 bytes long.

2.3.1.2 Setting Attribute Values in Sections - When you issue the SET section command, RMSDES will prompt you for attribute values in that section only. When you issue the SET ALL command, RMSDES will prompt you for attribute values in all sections.

Note that the prompting will be conditional based on your specifications for target system, file organization, and record format. That is, you will not be prompted for attributes that do not pertain to your file design; for example, you will be prompted for cluster size only if your target system is RSTS/E.

In response to each attribute prompt, you can either type a specific value or press RETURN to accept the default. Table 2-2 lists the defaults for attributes.

If you are designing an indexed file, you can allow RMSDES to define areas by default based on the number of keys you specified in the key section; see Section 2.6.5.

When you issue the SHOW ALL, SAVE, or CREATE command, RMSDES will assign the user-set and default attributes you specified in the design buffer; see Sections 2.3.4.2, 2.4, and 2.5.

While being prompted for attribute values, you may want to interrupt your design session to change previously set attributes, or to save your file design and resume the session later. Typing ESC will interrupt the design session, return the DES: prompt, and preserve all attribute values in the design buffer. In response to the DES: prompt, you can:

- Reset previously set attributes -- Use the CLEAR command to clear any section or attribute in the design buffer. Then use the SET command to reset them.

- Resume the design session -- Use the SET section command to return to the section in which you were setting attributes. Once the attribute prompt returns, press RETURN until you reach the attribute at which you left the design session; the attribute values you set earlier will be preserved. Then set attribute values as usual.

- Save the design -- Use the SAVE command to store your design in a description file. To resume your design session later, use the GET command to load the file design in the description file into the design buffer.

- Begin a new design session -- Use the CLEAR ALL command to clear the whole buffer. Then issue the appropriate SET command and design your file.

When attribute prompting is completed, the DES: prompt is returned. You can then:

- Use the SHOW command to display attribute values

- Use the SET command to set or reset more attribute values

- Use the CLEAR command to restore any attribute, section, or sections to their default values

- Use the SAVE command to save the file design

- Use the EXIT command to save the file design and terminate RMSDES

- Use the CREATE command to create an empty data file

- Use the QUIT command to terminate RMSDES (attribute values in the design buffer will not be saved)

Example

The following example illustrates how you are prompted for and can set attribute values using the SET ALL command. It also illustrates the use of the ESC command to interrupt the design session to reset an attribute value.

```
DES:  SET ALL
Starting SYSTEM section
    Target:  RSX
  File Placement:  (RET)
Starting FILE section
    Name:  PERS.DIS
    Organization (Seq):  (RET)
```

Pressing RETURN specifies the defaults: NO for file placement and sequential for file organization.

```
    Allocation (0):  2
    Extension (0):  (ESC)
DES:
```

You realize that you should have specified the default (NO) for Allocation. Typing ESC interrupts the design session, returns the DES: prompt, and preserves all previously set attribute values. To verify the Allocation setting, use the SHOW command:

```
DES:  SHOW FILE ALLOCATION
  ALLOCATION          2    (Explicit)
```

The SHOW command displays the value set for the allocation attribute. You specified 2 bytes instead of the default, 0.

```
DES:  CLEAR FILE ALLOCATION
DES:
```

The CLEAR command restores the default for Allocation, 0.

```
DES:  SET ALL
```

The SET ALL command resumes the design session.

```
Starting SYSTEM section
    Target:  (RET)
  File Placement:  (RET)
Starting FILE section
    Name:  (RET)
    Organization (Seq):  (RET)
    Allocation (0):  (RET)
```

Pressing RETURN preserves the previously set values.

```
        .
        .
        .
  Rewind_Magtape (NO):  (RET)
        .
        .
        .
```

You can now continue your design session, and will be prompted for the remainder of the file section, as well as for all other sections. Note that in this example, you will be prompted for whether the magnetic tape is to be rewound because you specified sequential file organization earlier. Pressing RETURN in response to this prompt specifies the default, NO.

NOTE

When RMSDES prompts for attribute values, it always displays the same default value; it does not show the current value which you may have set in the design buffer. You must issue a SHOW command to determine the current value.

In addition, note that a carriage return in response to a prompt will not change any existing value which has been set in the design buffer.

Under these circumstances, RMSDES will prompt for a value, displaying a default value. Responding with a carriage return will mean that a previously set value is used by RMSDES, not the default value that was displayed.

2.3.2 Setting Attribute Values Using External Files

You use the GET command (Section 2.2.6) to direct RMSDES to read attribute value information from an external file and to set the corresponding attributes in the design buffer.

When you issue a GET command, RMSDES first initializes the attributes in the design buffer to their original (default) values, and then reads in the attribute value information from the external file.

RMSDES reads attributes from two kinds of external files:

- Description file -- an ASCII sequential file that contains a file design; that is, it contains file attribute values set in a design session.
- Data file -- a file that contains or is capable of containing data records. RMSDES ignores any data records that the file may contain, loading only the attributes of the data files into the design buffer.

Sections 2.3.2.1 and 2.3.2.2 describe how description and data files are used.

Reading attribute values from an external file into the design buffer allows you to:

- Redesign some attributes of a file to improve its efficiency, and then to save the design or create the empty data file.
- Create an empty data file from the file design specified in a description file.
- Create an empty data file from the attributes of a data file. This allows you to use the data file as a template to copy the structure of a file without copying the records.
- Create a description file from the attributes of a data file.

Once the attribute values are set in the design buffer, you can:

- Use the SHOW command to display the attribute values of the file
- Use the SET command to redesign the file
- Use the SAVE command to save the file design in a description file
- Use the CREATE command to create an empty data file

2.3.2.1 Setting Attribute Values from a Description File - The following examples show how to load into the design buffer the file design attributes contained in description files created using RMSDES.

```
DES:  GET FOO
DES:
```

This command loads the file design attributes of FOO.DES into the design buffer. (The default file type, .DES, and default file kind, DES, need not be specified.)

```
DES:  GET FOO.BAR
DES:
```

This command loads the file design attributes of FOO.BAR into the design buffer. The file type must be specified because it is not the default. However, the kind, DES, need not be specified; RMSDES will assume that the file was created using RMSDES if no type is specified.

2.3.2.2 Setting Attribute Values from a Data File - Loading the attributes of a data file into the design buffer makes it possible for you to rapidly set attribute values in the design buffer without going through the whole design sequence. Once the attribute values are set in the buffer, you can change the file design, save the design, or create a new, empty data file. Once the new data file is created, you can store records in it.

Note, however, that if you specify an indexed data file, RMSDES cannot access the initial allocation for the areas: the allocation attribute for each area will be set to the number of blocks that remain available in each area of the data file, not to the initial allocation for each area in the data file. You can use the SHOW command to display the setting and then use the SET command to reset the allocation attribute as necessary.

If the external file is a data file, the file name and type in the design buffer will be set to those of the existing file. If you issue a CREATE command, the new data file will have the same file name and type as the existing data file and a version number 1 higher than the existing data file.

Reading the attributes of a data file into the design buffer is useful in the following cases:

- A file that you use processes records efficiently. As a result, you want to duplicate its file design. After loading the attributes of that data file into the design buffer with the GET command, you can save the design or create an empty data file. This procedure allows you to use the original data file as a template to save or copy the structure of the file without copying the records.

- A file that you use is largely efficient, but a few of the attributes need to be reset. After loading the attributes of the data file into the design buffer, you can rapidly make changes with the SET command. You can then save the design or create an empty data file.

For example:

```
DES:  GET FOO.BAR DAT
DES:
```

This command loads the attributes of FOO.BAR, a data file, into the design buffer.

2.3.3 Clearing Attribute Values

You use the CLEAR command (Section 2.2.1) to restore the attribute values of one attribute, of one section, or of all sections to their default values.

2.3.3.1 Clearing Individual Attribute Values - When you issue the CLEAR command for an individual attribute, the attribute value is restored to its default. You may want to restore an attribute value to its default if you find that the default value would improve your file design.

For example:

```
DES: CLEAR RECORD FORMAT
DES:
```

This command restores the record format to the default, VARIABLE.

2.3.3.2 Clearing Attribute Values in Sections - You clear user-set attribute values in sections when values you previously defined are incorrect or no longer useful.

The following example restores all the attributes in the file section to their default values:

```
DES: CLEAR FILE ALL
DES:
```

You can now use the SET command to reset the attributes in the file section. The attributes in the other sections of the design buffer will retain the values you have previously set.

You should also clear all sections after you have saved a file design or created a file, as follows:

```
DES: CLEAR ALL
DES:
```

The buffer will now be ready to accept a design for another file.

2.3.4 Showing Attribute Values

You use the SHOW command (Section 2.2.11) to display the attribute values in the buffer individually, in one section, or in all sections. Attribute values are displayed conditionally; that is, only the attributes that pertain to the file design are shown. For example, a cluster size value will be displayed only if you have specified RSTS/E as the target system.

2.3.4.1 Showing Individual Attribute Values - The SHOW command is useful when:

- You are setting an attribute value that depends on a previously set attribute value. Displaying the previously set attribute value confirms what value is already defined.
- You want to confirm that you have set an attribute to the correct value.

The following example shows how to display the file organization attribute value:

```
DES: SHOW FILE ORGANIZATION
  ORGANIZATION          INDEXED    (Explicit)
  DES:
```

2.3.4.2 Showing Attribute Values in Sections - The SHOW section command displays user-set and default attribute values in one section. The SHOW ALL command displays attribute values in all sections.

The SHOW command allows you to:

- Confirm what attribute values you have set.
- Inspect a file design that is stored in a description file. Inspecting the attribute values of the file design gives you an accurate profile of the file.
- Inspect a file that you did not finish designing, but stored as an incomplete design session in a description file. Inspecting the attribute values of the file design informs you what values are already set and defaulted, and where you interrupted your design session.
- Confirm that you are going to clear the right section. For example, if your file design contains multiple keys, you might want to inspect the specific key section to make sure you are going to clear the right key.

For example:

```
DES:  SET FILE ORGANIZATION INDEXED
DES:  SET FILE NAME INFILE.DAT
DES:  SHOW ALL
SYSTEM
  Target                    RSX
  Source                    RSX
FILE
  Name                      INFILE.DAT  (Explicit)
  Organization              INDEXED  (Explicit)
  Allocation                0
  Extension                 0
  Bucket_Size               1
  Protection
  Owner                     [303,24]
  Max_Record_Number         0
  Contiguous                NO
  Supersede                 NO
RECORD
  Size                      0
  Format                    VARIABLE
  Carriage_Control          CARRIAGE_RETURN
```

2.4 SAVING THE FILE DESIGN

When you have completed a design session, you can use the SAVE command (Section 2.2.9) or the EXIT command (Section 2.2.5) to store the file design in a description file. If you issue a SAVE command, the RMSDES prompt will be returned after the file design is stored and you can continue the design session. If you issue an EXIT command, RMSDES will be terminated and the system prompt will be returned after the file design is stored. If, after you save a file design, you want to design another file, you should issue a CLEAR ALL command to restore the attribute values in the design buffer to their defaults.

Saving the file design is useful when you would prefer not to create the empty data file, but to store the file design for later use instead. You may need to interrupt your design session, or may want to fine tune your design at a another time. Saving the file design allows you to resume your session later.

For example, to save your file design in a file called INFILE.DES, type:

```
DES:  SAVE INFILE
DES:
```

This command stores the file design in the file INFILE.DES. Because the default file type is .DES, you can omit the file type from the command line.

When you want to create the empty data file or resume the design session, you use the GET command to load the file design stored in INFILE.DES into the design buffer. You can then resume your design session or create the empty data file.

The description file is also a useful reference tool. Because it is an ASCII sequential file, the attributes of a file can be displayed or printed. If the attributes of the file have changed over time, the description file provides a record of those changes. In addition, if the data file is lost or destroyed, the description file can be used to create an exact copy of the file (without the original data records).

2.5 CREATING AN EMPTY DATA FILE

You can use the CREATE command (Section 2.2.2) to create an empty data file, based on the attribute values in the design buffer. The empty data file is capable of containing records for use by your programs.

If you are designing an indexed file but have defined no areas, you will be asked, when you issue the CREATE command, whether RMSDES is to define areas by default. If you respond YES or press RETURN (YES is the default), RMSDES will define areas based on the number of keys you specified in the key section.

If, after you create a file, you want to design another file, you should issue a CLEAR ALL command to restore the attribute values in the design buffer to their defaults.

Note that if you define areas and/or keys for an indexed file, they must be numbered sequentially starting with 0 (0, 1, 2,...) when you issue the CREATE command (see Section 2.2.2).

As described in Section 2.1.1, an empty data file can also be created from an existing description or data file. The new data file will have the attributes of the existing external file. This method of creating a file is the equivalent of issuing a GET command to set the attributes in the design buffer to match those of an external file, and then issuing a CREATE command to create an empty data file with those attributes. See Section 2.3.2 for more information on using external files.

The following examples show how to create an empty data file:

```
DES:  CREATE
DES:
```

This command creates an empty data file with the file name and type specified in the design buffer.

```
DES: CREATE NEWFIL
DES:
```

This command creates an empty data file with the name specified in the command line, NEWFIL.DAT. Specifying a file name and type in the command string is useful when you want to create many different data files with the same attributes.

```
DES INFILE
```

This command is issued from your system prompt. It creates an empty data file based on the file design specified in the description file, INFILE.DES. The empty data file is given the file name and type specified in the description file.

2.6 SECTIONS AND ATTRIBUTES

The RMSDES design buffer is divided into five sections:

- **System section** contains attributes that define the operating system on which you are using RMSDES and the operating system on which the file be will be used.
- **File section** contains attributes that define the file's characteristics, such as name, organization, owner.
- **Record section** contains attributes that define the characteristics of the records within the file, such as format, size, carriage control.
- **Key section** contains attributes that define the characteristics of keys and key segments in indexed files, such as name and data type for keys, and position and length for key segments.
- **Area section** contains attributes that define the characteristics of areas in indexed files, such as position, allocation, contiguity.

Sections 2.6.1 through 2.6.5 describe the design buffer sections and attributes. For each attribute, the attribute keyword and variable are given.

The description of each attribute gives the attribute's function, the default if you do not supply a value for the variable, whether the attribute is informational, and whether the attribute is conditional.

Informational attributes are those required by RMSDES; they are not RMS-11 file attributes. Some informational attributes, such as source, are saved in the description file; others, such as file placement, are only used by RMSDES to determine conditional prompting and are not saved in the description file.

Conditional attributes are those for which you are prompted only if you have given a particular response to some previous prompt. For example, you will be prompted for a control field size number only if you have specified variable with fixed control as the record format.

2.6.1 System Section

In the system section, you define the target system, that is, the operating system on which the file you are designing will be used. The source system (the operating system on which you are using RMSDES) is automatically set for you in the design buffer. You are not prompted for this attribute. Both the target and source attributes are informational attributes.

In addition, if you issue a SET ALL command, you will be prompted for FILE PLACEMENT after the TARGET prompt, as follows:

```
DES:  SET ALL
Starting SYSTEM section
     Target:  (RET)
   File Placement:
```

In response to the FILE PLACEMENT prompt, type either YES or NO to indicate whether you will specify POSITION in the area section (see Section 2.6.5):

- YES -- indicates that you will specify POSITION in the area section. You will not be prompted for allocation, extension, and bucket size in the file section; rather, you will be prompted for this information in the area section (area 0 is automatically defined for sequential and relative files).

- NO -- indicates that you will not specify POSITION in the area section. You will be prompted for allocation, extension, and bucket size in the file section.

The default is NO. FILE PLACEMENT is an informational attribute used to determine conditional prompting; it is not saved in the description file.

TARGET argument

Specifies the operating system on which the data file will be created. TARGET is an informational attribute and is saved in the description file.

For argument, type one of the following:

- RSX

- RSTS

The default is the system on which the file is being designed.

Specifying a particular operating system does not necessarily limit the use of the file to that system. As long as your description or empty data file is not system specific, the file can be processed on other operating systems. However, if your description file or empty data file is system specific, the file may not be usable on another operating system.

SOURCE

This attribute is not user set; RMSDES automatically sets this attribute to the operating system you are using to design the file. Source is an informational attribute and is saved in the description file.

2.6.2 File Section

NAME string

Specifies the name of the file. For string, type a file specification.

If you do not specify a file in response to the NAME prompt and also do not specify a file name and type when you issue the CREATE command (see Section 2.2.2), the file will be created as FILE.DAT.

ORGANIZATION argument

Specifies the organization of the file.

For argument, type one of the following:

- SEQUENTIAL -- specifies a sequential file. You can store sequential files on disk or magnetic tape.
- RELATIVE -- specifies a relative file. You can store relative files only on disk.
- INDEXED -- specifies an indexed file. You can store indexed files only on disk.

The default is SEQUENTIAL.

CLUSTER_SIZE number

Specifies the number of blocks in a cluster for this file. You are prompted for this attribute only if you have specified the target system as RSTS/E.

For number, type a value from 0 to 256 or from -1 to -256 indicating the cluster size value in blocks. See the RSTS/E System User's Guide for more information on specifying cluster size.

The default is 0 blocks which indicates that the pack cluster size of the disk the file will be created on will be used.

ALLOCATION number

Specifies the initial size of the file in blocks. You will not be prompted for this attribute if you responded YES to the FILE PLACEMENT prompt (see Section 2.6.1).

For number, type a decimal value indicating the allocation quantity in blocks. The minimum allocation you can specify is 0 blocks; the maximum depends on the amount of available space on the disk or magnetic tape.

If you are allowing RMSDES to define areas for an indexed file by default, RMSDES will automatically calculate an allocation value for each area it defines. However, if you choose to define areas explicitly and specify an allocation value for each area, accept the default for allocation in this section, and set the allocation attribute in each area section (see Section 2.6.5).

The default is 0 blocks.

EXTENSION number

Specifies the number of blocks that should be added to the file each time it is extended. You will not be prompted for this attribute if you responded YES to the FILE PLACEMENT prompt (see Section 2.6.1).

For number, type a decimal value indicating the extension quantity in blocks. This quantity should be a multiple of the file bucket size value.

When RMS-11 extends a file on RSX-11M/M-PLUS systems, it requests the number of blocks you have specified from the operating system.

When RMS-11 extends a file on RSTS/E systems, it requests the number of blocks you have specified from the operating system. Then, it extends the file to the next full cluster larger than the request.

If you are allowing RMSDES to define areas for an indexed file by default, RMSDES will automatically calculate an extension value for each area it defines. However, if you choose to define areas explicitly and specify an extension value for each area, accept the default for extension in this section and set the extension attribute in each area section (see Section 2.6.5).

The default is 0 blocks.

BUCKET_SIZE number

Specifies the number of blocks in a bucket for the file. This number determines the number of blocks read into memory during each file access operation, and therefore affects processing speed and the amount of memory a program accessing this file requires. You will not be prompted for this attribute if you responded YES to the FILE PLACEMENT prompt (see Section 2.6.1) or if you specified sequential as the file organization.

If your target system is RSX-11M/M-PLUS, for number, type a decimal value from 0 to 32, indicating the bucket size in blocks. The minimum value permitted is the number of blocks that can contain one record; the maximum is 32 blocks.

If your target system is RSTS/E, for number, type a decimal value from 0 to 15, indicating the bucket size in blocks. The minimum value permitted is the number of blocks that can contain one record; the maximum is 15 blocks.

In choosing a bucket size for RSTS/E files, you should consider the file cluster size. Because file cluster size governs file extension quantities and the way the RSTS/E file processor handles disk read operations, you can sometimes improve performance by aligning buckets with clusters. (See the RSTS/E Programming Manual.)

If you are allowing RMSDES to define areas for an indexed file by default, RMSDES will automatically calculate an allocation value for each area it defines. However, if you choose to define areas explicitly and specify a bucket size value for each area, you should accept the default for bucket size in this section and set the bucket size attribute in each area section (see Section 2.6.5).

The default is 1 block.

PROTECTION string

Specifies the type of access different categories of users can have to the file.

RSX-11M/M-PLUS. If your target system is RSX-11M/M-PLUS, the protection code is expressed as a string of characters that defines the legal categories of users. The categories of users are system, owner, group, and world.

For each category of user, you specify one or more of the letters representing the access types: RWED (representing Read, Write, Extend, and Delete).

The user categories can be specified in any order in the following format:

```
(GROUP:RWED,OWNER:RWED,WORLD:RWED,SYSTEM:RWED)
```

To deny access to a user category, simply omit the letter representing the access type:

```
(GROUP:RWE,OWNER:RWED,WORLD:,SYSTEM:RWED)
```

The users in the group are denied access to delete the file. World users are denied all access. (For more information on protection, see the RSX-11M/M-PLUS MCR Operations Manual and the RSX-11M-PLUS Command Language Manual.)

You can set individual protection fields. For example:

```
DES: SET FILE PROTECTION (GROUP:RW)
```

The users in the group can read and write the file; all other user categories have the default access.

You can also set one protection field at one time and set another later. For example, suppose you had previously set group protection as above and now set world protection as follows:

```
DES: SET FILE PROTECTION (WORLD:R)
```

The previously set group protection (RW) would remain set, world users would have read access, and all other users would have the default access.

If you use the SHOW command to display protection, only the fields that you explicitly set will be displayed. For example, suppose you made the settings in the above examples and then issued the SHOW command to verify the settings:

```
DES: SHOW FILE PROTECTION
  PROTECTION      ( GROUP:RW , WORLD:R )    ( Explicit )
```

The CLEAR command restores all fields to their default values. You cannot clear an individual field. You can, however, use the SET command to reset a field to a new explicit setting.

If you use the GET command and specify an external file, RMSDES will set the protection to correspond to the setting in the file. If the external file is a data file and protection is set to the system default (the default for this attribute), the protection will be displayed, if you issue a SHOW command, as if it had been explicitly set. However, if the external file is a description file and protection is set to the system default, the protection will not be displayed if you issue a SHOW command.

RSTS/E systems. If your target system is RSTS/E, the protection code is expressed as a decimal number ranging from 0 to 255. See the RSTS/E System User's Guide for more information on calculating protection numbers.

The default is the target system default.

OWNER string

Specifies the owner of the file.

If your target system is RSX-11M/M-PLUS, you will be prompted for file owner. For string, type the User Identification Code (UIC) of the user who will own the file.

The default is the UIC of the person using RMSDES.

If your target system is RSTS/E, you will not be prompted for this attribute.

MAGTAPE_BLOCK_SIZE number

Specifies the number of bytes in each magnetic tape block. This attribute applies to magnetic tape files only. You will be prompted for this attribute only if you specified sequential as the file organization.

For number, type a decimal value from 18 through 8192.

The default is 512 bytes.

REWIND_MAGTAPE logical

Specifies that the magnetic tape is to be rewound, and that the file is to be stored at the beginning of the tape. This attribute applies to magnetic tape files only. You will be prompted for this attribute only if you specified sequential as the file organization.

For logical, type YES or NO:

- YES -- specifies that the tape will be rewound.
- NO -- specifies that the tape will not be rewound.

The default is NO.

MAX_RECORD_NUMBER number

Specifies the maximum number of records that a relative file will contain. If you are designing a sequential or indexed file, you will also be prompted for this value; RMSDES will use the number you supply for information if you allow RMSDES to define areas by default (area 0 is automatically defined for sequential and relative files; see the FILE PLACEMENT attribute, Section 2.6.1).

If you allow RMSDES to define areas by default, you must explicitly set this attribute.

RMS-11 checks the maximum record number whenever a record access operation is requested for this file. If the relative record number specified for the access operation exceeds the maximum record number, RMS-11 will issue an error message.

For number, type a decimal value indicating the maximum record number value.

Accepting the default value allows the file to contain as many records as physically possible.

The default is 0 records.

CONTIGUOUS logical

Specifies whether the disk space for this file will be allocated in contiguous blocks. File contiguity minimizes head movement when the file is being accessed.

For logical, type YES or NO:

- YES -- specifies that the file will be contiguous.
- NO -- specifies that the file will not be contiguous.

If you are defining areas explicitly, you should take the default for this attribute and specify contiguity in the area section (see Section 2.6.5).

If you are defining areas for an indexed file and your target system is RSX-11M/M-PLUS, it is possible that the areas will not be contiguous. However, the records within each area will be contiguous if you accept the default (NO) for this attribute and set the contiguous attribute in the area section to YES. (See Section 2.6.5.)

If your target system is RSTS/E and you declare the file to be contiguous, it will not be able to be extended unless you specifically make it non-contiguous. To achieve both file contiguity and extension, find contiguous space on your disk and specify the logical block number where you want the file to begin.

The default is NO.

SUPERSEDE logical

Specifies whether you want the present version of the file to supersede an earlier version with an identical file name, type, and (if your target system is RSX-11M/M-PLUS) version.

For logical, type YES or NO:

- YES -- specifies that this file will supersede an existing file.
- NO -- specifies that if RMSDES encounters a file with an identical file specification, it will not create the file.

The default is NO.

2.6.3 Record Section

SIZE number

Specifies the record size in bytes.

For number, type a decimal value from 1 to 16,384 indicating the record size. If you are allowing RMSDES to define areas for an indexed file by default, you must explicitly set this attribute.

If you accept the default, RMS-11 will accept records of any length up to 16K bytes into the file. Note, however, that the actual record length accepted by RMS-11 may depend on other factors, such as file organization, record format, target operating system. If your target system is RSX-11M/M-PLUS, see the RSX-11M/M-PLUS RMS-11 User's Guide; if your target system is RSTS/E, see the RSTS/E RMS-11 User's Guide.

The meaning of the record size information depends on the record format:

- For variable-length and stream records, the record size you specify is a maximum length, the size of the largest possible record for that file. RMS-11 rejects any write operation using a record size greater than the maximum.

- For fixed-length records, the record size you specify is the same length for every record in the file. RMS-11 rejects any write operation that specifies a record of the wrong size.

- For variable-with-fixed-control records, there are two size specifications:

  - Maximum length of the variable area

  - Length of the fixed control area

  Specifying the size attribute sets the maximum length of only the variable area of your record. To specify the length of the fixed control area, set the CONTROL_FIELD_SIZE attribute (described below).

  RMS-11 treats the specifications for both the variable area and the fixed-control area the same way it treats the sizes for variable-length records and fixed-length records, respectively.

The default is 0 bytes.

FORMAT argument

Specifies the record format for the file.

For argument, type one of the following:

- VARIABLE -- specifies that records in the file will have differing, or variable, lengths.

- STREAM -- specifies that the records in the file will have no specific format, but will be delimited only by record terminator characters. The stream format is permitted for sequential disk files only.

- FIXED -- specifies that the records in the file have the same, fixed length.
- VFC -- specifies that each record in the file has a control area with a fixed length and a data area of no standard length.

The default is VARIABLE.

CONTROL_FIELD_SIZE number

Specifies the size in bytes of the fixed control area of each record in the file. You will be prompted for this attribute only if you specified the record format as variable with fixed control.

For number, type a decimal value from 1 through 255.

The default control field size is 2 bytes.

BLOCK_SPAN logical

Specifies whether records can cross block boundaries. You will be prompted for this attribute only if you specified the file organization as sequential.

Spanning block boundaries maximizes use of space on a disk. Otherwise, the size of a record is limited to 1 block.

For logical, type YES or NO:

- YES -- specifies that records can cross disk block boundaries.
- NO -- specifies that records cannot cross disk block boundaries.

The default is YES.

CARRIAGE_CONTROL argument

Specifies how your records are written to a unit record device (terminal or printer).

For argument, type one of the following:

- CARRIAGE_RETURN -- specifies that each record will be preceded by a line feed character and followed by a carriage return character when it is written to a unit record device.
- FORTRAN -- specifies that the first byte of each record is to be interpreted as a FORTRAN forms-control character when the record is written to a unit record device.
- NONE -- specifies no carriage control.

The default is CARRIAGE_RETURN.

2.6.4 Key Section

The key section allows you to define keys for an indexed file. You can define the following key characteristics:

- Number of keys
- Position, size, type, and name of each key
- Whether records can have duplicate key values
- Whether alternate key values can change during update operations
- Whether keys are segmented
- Null key value for duplicate keys
- Fill factors for keys

If you are designing an indexed file, RMSDES will prompt you for the number of keys that the file will contain. If you issue the SET ALL command, RMSDES will prompt you for the number of keys when you reach the key section. If you issue the SET KEY ALL command, RMSDES will prompt you immediately for the number of keys, as follows:

```
Number of keys?
```

There is no default response to this prompt. Type a decimal number to specify the number of keys you are defining for the area, and press RETURN. You can specify up to 255 keys per area; RMSDES will number them from 0 to 254. You will be prompted for all of the key attributes for every key you are defining. Note that keys must be numbered sequentially starting with 0 (0, 1, 2,...) when you issue the CREATE command (see Section 2.2.2).

If you are setting only one key, you must specify the key number to be prompted for key attributes. For example, to set a primary key, you would use the following SET command:

```
SET KEY 0 ALL
```

If you are allowing RMSDES to define areas by default, RMSDES will define the areas based on the number of keys you specify in this section (see Section 2.6.5 and Appendix B for more information on how RMSDES defines areas).

NAME string

Specifies the name of the key.

For string, type a key name of up to 32 ASCII characters.

Accepting the default specifies that no name is assigned to the key.

TYPE argument

Specifies the data type of the key.

For argument, type one of the following:

- STRING -- specifies that the key value is a string of alphanumeric characters.
- BIN2 -- specifies that the key value is a 2-byte unsigned integer.
- BIN4 -- specifies that the key value is a 4-byte unsigned integer.
- INT2 -- specifies that the key value is a 2-byte signed integer.
- INT4 -- specifies that the key value is a 4-byte signed integer.
- DECIMAL -- specifies that the key value is a packed decimal number.

The default is STRING.

NULL_KEY logical

Specifies whether the file will contain some records that cannot be accessed by this key. These records will be assigned a null value for this key. The null value is specified in the NULL_VALUE attribute, described below. The null key attribute applies only to alternate keys.

For logical, type YES or NO:

- YES -- specifies that this alternate key will contain a null key value.
- NO -- specifies that this alternate key will not contain a null key value.

The default is NO.

NULL_VALUE argument

This attribute applies to alternate keys only. It specifies the null value for this key; that is, the file will contain some records that cannot be defined by the key. You will be prompted for this attribute only if you have specified YES for the NULL_KEY attribute, described above.

For argument, you can type:

- An ASCII character -- to indicate a character null value, type a character or a space, enclosed in apostrophes. For example, 'X' indicates that the character X is a null value.
- A decimal number -- to indicate a numeric null value, type a number from 0 to 255.

The default is a space (' ').

DUPLICATES logical

Specifies whether the file can contain more than one record with the same value for this key.

For logical, type YES or NO:

- YES -- specifies that duplicate key values will be allowed. If you type YES for alternate keys and are allowing RMSDES to define areas by default, you will be prompted for the average number of duplicates per alternate key; RMSDES uses this value to calculate area allocation (see Section 2.6.5).
- NO -- specifies that duplicate key values will not be allowed.

The default for alternate keys is YES; for primary keys, NO.

CHANGES logical

Specifies whether an alternate key can change values during an update operation; that is, whether the record can be read with one value for the key and rewritten with another value for the same key. This attribute applies only to alternate keys.

For logical, type YES or NO:

- YES -- specifies that the value of this alternate key can be changed.
- NO -- specifies that the value of this alternate key cannot be changed.

The default is YES.

DATA_FILL number

Specifies the percentage of the data level bucket that should be filled when the file is initially loaded.

For number, type a decimal value from 50 to 100, indicating the percentage to which each data bucket should be filled.

The default is 100 percent.

DATA_AREA number

Specifies the area that will contain the data records for the primary key and pointers to the data records for an alternate key.

For number, type a decimal value indicating the area.

If you are allowing RMSDES to define areas by default, you must accept the default for this attribute.

Note that if you specify a data area, you also must specify the allocation, extension, bucket size, and contiguous attributes of this area in the area section (see Section 2.6.5). Otherwise, an error message will be issued and the area will not be created.

The default is area 0.

INDEX_FILL number

Specifies the percentage of the index level bucket that should be filled when the file is initially loaded.

For number, type a decimal value from 50 to 100, indicating the percentage to which each index level bucket should be filled. Because the index level buckets contain only pointers rather than actual records, you may want to specify a different fill factor for the index level buckets than for the data level buckets.

The default is 100 percent.

LEVEL1_INDEX_AREA number

Specifies the area that will contain the lowest-level index area for this key.

For number, type a decimal value indicating the area.

If you are allowing RMSDES to define areas by default, you must accept the default for this attribute.

Note that if you specify a level-1 index area, you must make sure that the area is defined by setting the allocation, extension, bucket size, and contiguous attributes for the area in the area section (see Section 2.6.5). If you have issued a SET ALL command, you will be prompted for these attributes automatically. Otherwise, you can issue a SET AREA command to be prompted for them. You can take the defaults for any of the attributes. However, if the area is not defined, an error message will be issued and the area will not be created when you issue a CREATE command.

The default is area 0.

INDEX_AREA number

Specifies the area that will contain the upper levels of the index for this key.

For number, type a decimal value indicating the area.

If you are allowing RMSDES to define areas by default, you must accept the default for this attribute.

Note that if you specify an index area, you must make sure that the area is defined by setting the allocation, extension, bucket size, and contiguous attributes for the area in the area section (see Section 2.6.5). If you have issued a SET ALL command, you will be prompted for these attributes automatically. Otherwise, you can issue a SET AREA command to be prompted for them. You can take the defaults for any of the attributes. However, if the area is not defined, an error message will be issued and the area will not be created when you issue a CREATE command.

The default is area 0.

2.6.4.1 Key Segments - This part of the key section allows you to define segments for keys. Note that only string keys can have more than one segment. Nonstring keys, by definition, have one segment (segment 0).

If you have specified a string key type, and are defining keys interactively, you will be prompted for the number of segments you are defining for your key, as follows:

```
Number of segments?
```

This prompt occurs after the DUPLICATES prompt. Type a decimal number to specify the number of segments you want to define for that key, and press RETURN. You can specify up to eight segments, numbered from 0 to 7.

You will be prompted for the SEGn_POSITION and SEGn_LENGTH attributes, described below. You must define all segments you have specified. For example, if you have specified that a key will have three segments, you will be prompted for the position and length of all three segments.

SEGn_POSITION number

Specifies the position of the first byte of the segment within the record. This attribute applies to string keys only.

For number, type a decimal value indicating the position of the first byte of the segment within the record. Bytes within a record are numbered sequentially starting with 0.

The default is byte 0.

SEGn_LENGTH number

Specifies the length, in bytes, of the segment within the record. This attribute applies to string keys only.

For number, type a decimal value from 0 to 255 indicating the length of the segment within the record.

Note that the maximum length of a key segment depends on the record size you have specified.

The default is 0 bytes.

SEG0_POSITION number

Specifies the position of the first byte of of a nonstring key within the record. Nonstring keys cannot have more than one segment; however, this attribute allows you to position position a single-segment nonstring key (segment 0) within the record.

For number, type a decimal value indicating the position of the first byte of segment 0 within the record.

The default is byte 0.

2.6.5 Area Section

The area section allows you to define areas for an indexed file.

Areas are portions of an indexed file that are treated independently for position, initial allocation, extension, bucket size, and

contiguity. Defining areas allows you to improve the performance of your file by dividing indexed files into separate units for each index and for the data levels.

Defining Areas Explicitly. To define areas explicitly, you first should accept the defaults for the contiguous, allocation, extension, and bucket size attributes in the file section. Note, however, that if you set these attributes in both the file and area sections, the values set in the area section will override the values set in the file section.

Then, you must specify which area will contain the data records and which areas will contain the indexes of each key by setting the data area, level-1 index area, and index area attributes in the key section (see Section 2.6.4).

Finally, you must set the attributes of each area by explicitly setting values in this section.

If you issue a SET ALL command, RMSDES will prompt you when you reach the area section:

```
Areas defaulted:
```

Since you are defining areas explicitly, type NO. RMSDES will then prompt you:

```
Number of areas:
```

If you issue a SET AREA ALL command, RMSDES will prompt you immediately for the number of areas. Type a decimal number to specify the number of areas you are defining for the file. You can specify up to 255 areas; RMSDES will number them from 0 to 254.

RMSDES will prompt you for each area attribute in every area you define. Note that areas must be numbered sequentially starting with 0 (0, 1, 2,...) when you issue the CREATE command (see Section 2.2.2).

If you define no areas but issue a CREATE command, the file will be created with a single area (the RMS-11 default); if you issue a SHOW ALL command, no area will be created.

Defining Areas by Default. In most cases, letting RMSDES define areas by default is the best option. To do so, first you should accept the defaults for the allocation, extension, bucket size, and contiguous attributes in the file section (see Section 2.6.2), and explicitly set the size attribute in the record section (see Section 2.6.3).

If you issue a SET ALL command, RMSDES will prompt you when you reach the area section, as follows:

```
Areas defaulted:
```

Type YES or press RETURN (YES is the default) to indicate that you want RMSDES to define the areas.

You will also be prompted for bucket size:

```
Bucket_Size (2):
```

Type a decimal number indicating the bucket size in blocks. The minimum value that you should enter is the number of blocks that can contain one record. If your target system is RSX-11M/M-PLUS, the

maximum is 32 blocks. If your target system is RSTS/E, the maximum is 15 blocks. The default is 2 blocks for all target systems. (See also the description of the BUCKET_SIZE attribute below.)

These prompts will also appear if you issue a CREATE command but have defined no areas.

Based on your responses to the prompts, RMSDES will begin to define the areas. First, it will construct the primary index and issue an informational message:

```
The primary index has a depth of n
```

where **n** is the number of index levels, excluding the data level (0).

If you have specified multiple keys, and answered YES to the DUPLICATES prompt in the key section (see Section 2.6.4), RMSDES will print the next alternate key number and prompt:

```
Average number of duplicates?
```

There is no default response to this prompt; type a decimal number to indicate the average number number of duplicates per record for this key. RMSDES will use the value you enter in its area calculations.

After you respond to the prompt, RMSDES will construct the alternate index and issue an informational message:

```
The index for alternate key x
  has a depth of n
```

where **x** is the alternate key number and **n** is the number of index levels, excluding the data level (0).

The prompt and informational message will be repeated for each alternate key.

When you issue a CREATE command, RMSDES will define areas based on your responses to the prompts and on the number of keys you specified in the key section. See also Appendix B.

Note that you should issue the CLEAR command to clear any area attributes you may have previously set. If any area attributes are set when you issue a CREATE command, you will not receive the "Areas defaulted" prompt; RMSDES will assume that you want to set areas explicitly and will prompt you for the area settings.

ALLOCATION number

Specifies the initial size of the area in blocks.

For number, type a decimal value indicating the allocation quantity in blocks.

If you specify an allocation value for this area, you should also accept the default for allocation in the file section (see Section 2.6.2). If the allocation attribute is set in both the file and the area sections, the value in the area section will override the value in the file section.

If you are allowing RMSDES to define areas by default, RMSDES will automatically calculate this attribute value.

The default is 0 blocks.

EXTENSION number

Specifies the number of blocks that will be added to the area each time it is extended.

For number, type a decimal value indicating the extension quantity in blocks. This extension quantity should be a multiple of the area bucket size value.

If you are allowing RMSDES to define areas by default, RMSDES will automatically calculate this attribute value.

When RMS-11 extends an area on RSX-11M/M-PLUS systems, it requests the number of blocks you have specified from the operating system.

When RMS-11 extends an area on RSTS/E systems, it requests the number of blocks you have specified from the operating system. Then, it extends the area to the next full cluster larger than the request.

If the extension attribute is set in both the file and the area sections, the value in the area section will override the value in the file section.

The default is 0 blocks.

BUCKET_SIZE number

Specifies the number of blocks in a bucket for the area. You are prompted for this attribute only if you specified relative or indexed file organization in the file section (see Section 2.6.1).

The bucket size number determines the number of blocks read into memory during each area access operation, and therefore affects processing speed and the amount of memory a program accessing this area requires.

If your target system is RSX-11M/M-PLUS, for number, type a decimal value from 1 to 32 indicating the bucket size in blocks. The minimum value that you should enter is the number of blocks that can contain one record; the maximum is 32 blocks.

If your target system is RSTS/E, for number, type a decimal value from 1 to 15 indicating the bucket size in blocks. The minimum value that you should enter is the number of blocks that contain one record; the maximum is 15 blocks.

In choosing a bucket size for RSTS/E files, you should consider the file cluster size. Because file cluster size governs file extension quantities and the way the RSTS/E file processor handles disk read operations, you can sometimes improve performance by aligning buckets with clusters.

If the bucket size attribute is set in both the file section and area section, the value in the area section will override the value in the file section.

If you are allowing RMSDES to define areas by default, RMSDES will prompt you for this attribute value; note that, in this case, the default is 2 blocks (see the discussion at the beginning of this section titled Defining Areas by Default).

The default is 0 blocks if you are explicitly defining areas; 2 blocks if you allow RMSDES to define areas by default.

CONTIGUOUS logical

Specifies whether the disk space for this area will be allocated in contiguous blocks. Area contiguity minimizes head movement when the area is being accessed.

For logical, type YES or NO:

- YES -- specifies that the records within the area will be contiguous.
- NO -- specifies that the records within the area will not be contiguous.

If you declare area contiguity, the records within each area will be contiguous; however, the areas themselves may not be contiguous.

If you have specified area contiguity, you should also have accepted the default for contiguity (NO) in the file section (see Section 2.6.2). If the contiguous attribute is set in both the file and the area sections, the entire file will be contiguous.

The default is NO.

If indexed areas are present, and (a) no placement is specified, (b) contiguous areas are not desired, and (c) the file was requested contiguous, the area sizes will be summed and the file created as one contiguous area. The areas will be allocated from that.

POSITION argument

Specifies file placement, that is, where the area will be positioned on disk. This attribute applies to disk files only. You will be prompted for this attribute only if you responded YES to the FILE PLACEMENT prompt in the system section (see Section 2.6.1).

For argument, type one of the following:

- NONE -- specifies that the area is to be positioned wherever there is space on the disk.
- VIRTUAL number -- specifies that the area will begin near the designated virtual block.

  For number, type a decimal number indicating the virtual block number (VBN).

- LOGICAL number -- specifies that the area is to begin at the designated logical block.

  For number, type a decimal number indicating the logical block number (LBN).

The default is NONE.

EXACT_POSITIONING logical

Specifies whether the area will be positioned beginning exactly at the logical block number (LBN) specified for the POSITION attribute. You will be prompted for this attribute only if you have specified an LBN for POSITION.

For logical, type YES or NO:

- YES -- specifies that the area must begin at the specified LBN. If the area cannot be positioned there, RMSDES will not create the file.
- NO -- specifies that if the area cannot be positioned exactly at the specified LBN, RMSDES will create the file, positioning it as close as possible to the specified LBN.

The default is NO.

2.7 RMSDES EXAMPLES

```
DES:  SET ALL
Starting SYSTEM section
    Target:  RSX
  File Placement:
Starting FILE section
    Name:  PAYROL.DAT
    Organization (Seq):  INDEXED
    Allocation (0):  3
    Extension (0):
    Bucket_Size (1):
    Protection:
    Owner:
    Max_Record_Number (0):
    Contiguous (No):
    Supersede (No):  YES
Starting RECORD section
    Size (0):
    Format (Var):  VARIABLE
    Carriage_Control (CR):  CARRIAGE
    Number of keys:  2
Starting Key  0
    Name:
    Type (String):  STRING
    Duplicates (Yes if alternate):  YES
    Number of Segments:  1
    Seg0_Position:  0
    Seg0_Length:  12
    Data_Fill (100):
    Data_Area (0):
    Index_Fill (100):
    Level1_Index_Area (0):
    Index_Area (0):
Starting Key  1
    Name:
    Type (String):  STRING
    Null_Key (No):  NO
    Duplicates (Yes if alternate):  YES
    Changes (Yes):  YES
    Number of Segments:  1
    Seg0_Position:  0
    Seg0_Length:  17
```

```
    Data_Fill (100):
    Data_Area (0):
    Index_Fill (100):
    Level1_Index_Area (0):
    Index_Area (0):
Areas defaulted:  NO
    Number of areas:  1
Starting Area  0
    Allocation (0):  3
    Extension (0):
    Bucket_Size (1):
    Contiguous (No):
    Position (None):
DES:  CREATE
DES:  SAVE PAYROL.DES
DES:  ^Z
```

The SET ALL command is used in this example to create an indexed file, PAYROL.DAT, which has an initial allocation of 3 blocks, variable-length records with carriage-control, and two string keys. One area is defined.

The contents of the design buffer are also saved in a file called PAYROL.DES.

```
DES:  SET FILE NAME STAT.DAT
DES:  SHOW ALL
SYSTEM
  Target                    RSX
  Source                    RSX
FILE
  Name                      STAT.DAT  (Explicit)
  Organization              SEQUENTIAL
  Allocation                0
  Extension                 0
  Protection
  Owner                     [70,5]
  Magtape_Blocksize         512
  Rewind_Magtape            NO
  Max_Record_Number         0
  Contiguous                NO
  Supersede                 NO
RECORD
  Size                      0
  Format                    VARIABLE
  Control_Field_Size        0
  Block_Span                YES
  Carriage_Control          CARRIAGE_RETURN
DES:  CREATE
```

This example illustrates the default values which are initialized in the design buffer. The file name is the only value which is explicitly set; the remaining values are the default values.

The sequential file, STAT.DAT, is created in the default account. The owner of the file is the default account, and the records are variable length with the carriage-control attribute. The file is not contiguous and does not supersede any existing file with the same name.

```
DES:  CLEAR ALL
DES:  SET FILE NAME EMPLY.DAT
DES:  SET FILE ORGANIZATION RELATIVE
DES:  SET RECORD FORMAT
  VALUE:  ?
```

Format argument

Specifies the record format for the file.

For argument, type one of the following:

- VARIABLE -- specifies that records in the file will have differing, or variable, lengths.
- STREAM -- specifies that the records in the file will have no specific format, but will be delimited only by record terminator characters. The stream format is permitted for sequential disk files only.
- FIXED -- specifies that the records in the file have the same, fixed length.
- VFC -- specifies that each record in the file has a control area with a fixed length and a data area of no standard length.

The default is VARIABLE.

```
  VALUE:  VFC
DES:  SET RECORD SIZE 80
DES:  SET FILE BUCKET 2
DES:  SHOW ALL
SYSTEM
  Target                    RSX
  Source                    RSX
FILE
  Name                      EMPLY.DAT  (Explicit)
  Organization              RELATIVE  (Explicit)
  Allocation                0
  Extension                 0
  Bucket_Size               2  (Explicit)
  Protection
  Owner                     [70,5]
  Max_Record_Number         0
  Contiguous                NO
  Supersede                 NO
RECORD
  Size                      80  (Explicit)
  Format                    VFC  (Explicit)
  Control_Field_Size        0
  Carriage_Control          CARRIAGE_RETURN
DES:  SET FILE ALLOCATION 30
DES:  CREATE
```

This example illustrates the interactive nature of RMSDES.

First, some design parameters are specified: file name and file organization. When record format is specified, however, the help facility is invoked in order to choose the VFC record format.

Next, record size and file bucket size are specified. Then, the SHOW ALL command is used to display the entire contents of the design buffer.

Finally, the CREATE command is used to create the EMPLY.DAT file.

```
DES:  CLEAR ALL
DES:  SET FILE ALL
Starting FILE section
    Name:  ACCNT.DAT
    Organization (Seq):  INDEXED
    Allocation (0):  30
    Extension (0):
    Bucket_Size (1):  3
    Protection:
    Owner:
    Max_Record_Number (0):  ?

Max_Record_Number number

     Specifies the maximum number of records that a relative file will
     contain.  If you are designing a sequential or indexed file,
     you supply this as an informational value if you allow RMSDES to
     define areas by default. (Area 0 is automatically defined for
     sequential and relative files.)

     If you allow RMSDES to define areas by default, you must
     explicitly set this attribute.

     RMS-11 checks the maximum record number whenever a record access
     operation is requested for this file. If the relative record
     number specified for the access operation exceeds the maximum
     record number, RMS-11 will issue an error message.

     For number, type a decimal value indicating the maximum record
     number value.

     Accepting the default value allows the file to contain as many
     records as physically possible.

     The default is 0 records.

    Max_Record_Number (0):
    Contiguous (No):
    Supersede (No):  YES
DES:  SET RECORD ALL
Starting RECORD section
    Size (0):  80
    Format (Var):  FIXED
    Carriage_Control (CR):  NONE
DES:  SET KEY 0 ALL
Starting Key  0
    Name:
    Type (String):  STRING
    Duplicates (Yes if alternate):
    Number of Segments:  1
    Seg0_Position:  0
    Seg0_Length:  15
    Data_Fill (100):
    Data_Area (0):
    Index_Fill (100):
    Level1_Index_Area (0):
    Index_Area (0):
```

```
DES:  SHOW ALL
SYSTEM
  Target                    RSX
  Source                    RSX
FILE
  Name                      ACCNT.DAT  (Explicit)
  Organization              INDEXED  (Explicit)
  Allocation                30  (Explicit)
  Extension                 0
  Bucket_Size               3  (Explicit)
  Protection
  Owner                     [70,5]
  Max_Record_Number         0
  Contiguous                NO
  Supersede                 YES  (Explicit)
RECORD
  Size                      80  (Explicit)
  Format                    FIXED  (Explicit)
  Control_Field_Size        0
  Carriage_Control          NONE  (Explicit)
KEY    0
  Name
  Type                      STRING  (Explicit)
  Null_Key                  NO
  Null_Value                0
  Duplicates                NO
  Changes                   NO
  Seg0_Position             0  (Explicit)
  Seg0_Length               15  (Explicit)
  Data_Fill                 100
  Data_Area                 0
  Index_Fill                100
  Level1_Index_Area         0
  Index_Area                0
DES:  CREATE
Areas defaulted:  ?
```

If you issue a SET ALL, a SET AREA ALL, or a CREATE command on an indexed file and have not defined areas, RMSDES will prompt you when you reach the area section:

```
Areas defaulted:
```

If you are defining areas explicitly, or do not want RMSDES to define areas for you, type NO. If you issued a SET ALL command, RMSDES will prompt you for the number of areas you wish to define. If you issued a CREATE command, the file will be created with the RMS-11 default area. If you issued a SHOW ALL command, no action will be taken.

Type YES or press RETURN to indicate that you want RMSDES to define the areas. The areas will be defined based on the number of keys you specified in the key section.

```
Areas defaulted:  NO
DES:  QUIT
```

In this example, the design is completed by specifying the file a "section" at a time.

The SET FILE ALL command is used to define all file attributes and help facility is used to get more information on the MAX_RECORD_NUMBER attribute. The SET RECORD ALL and SET KEY 0 ALL commands complete the definition of the file, and the SHOW ALL command displays the contents of the design buffer.

The CREATE command is invoked to create the file; since areas have not been defined, RMSDES queries about defaulting the area definition. In this case, the answer "NO" is given causing the file to be created with the RMS-11 default area.

2.8 RMSDES MESSAGES

This section documents the error messages returned by RMSDES. Each message is displayed in the format:

?DES-F-xxx, message text

?DES

Identifies the message as an RMSDES message.

F

Severity code. A severity code of F is "fatal": the operation will not be performed. After the error message is displayed, the DES: prompt will reappear. Retry the operation.

xxx

Message code: an abbreviation of the message text.

message text

Indicates the reason that the error occurred.

The RMSDES messages are listed below in alphabetical order by the message code. Each message is accompanied by an explanation of the error and recommended user action to correct the error.

?DES-F-AMA, Ambiguous attribute specified

Explanation: You entered an attribute keyword using an abbreviation that was not unique.

User Action: Reenter the command using a valid abbreviation for the attribute keyword.

?DES-F-AMC, Ambiguous command

Explanation: You entered a command using an abbreviation that was not unique.

User Action: Reenter the command using a valid abbreviation.

?DES-F-AMS, Ambiguous section specified

Explanation: You entered a section keyword using an abbreviation that was not unique.

User Action: Reenter the command using a valid abbreviation for the section keyword.

?DES-F-ANS, Area not defined

Explanation: You specified an area that has not been defined.

User Action: Reenter the command using the number of a defined area. Use the SHOW AREA ALL command to display the defined areas.

?DES-F-AOR, Area number out of legal range

Explanation: Areas are numbered from 1 through 255. You specified an area number outside of this range.

User Action: Reenter the command specifying a valid area number.

?DES-F-CSE, Syntax error

Explanation: You entered a string in response to the Protection or the Owner attribute prompt using incorrect syntax.

User Action: Reenter the string using correct syntax. Protection should be specified in the format:

```
(group:RWED,owner:RWED,world:RWED,system:RWED)
```

Owner should be specified as the UIC of the user who will own the file:

```
[n,n]
```

The value n is an octal number.

?DES-F-IAN, Illegal area number

Explanation: You entered an illegal value for an area number.

User Action: Reenter the command specifying a positive number from 0 through 254.

?DES-F-IBR, Illegal response given, must be YES or NO

Explanation: You were prompted for an attribute value that requires a logical argument.

User Action: Enter YES or NO.

?DES-F-ICL, Incomplete command line within a command file

Explanation: You specified an incomplete command line within an indirect command file. Execution of the command file will stop at that point.

User Action: Correct the indirect command file so that the command line is complete and re-execute the job.

?DES-F-IFT, Illegal file type value specified

Explanation: You issued a GET command or you used the DES filename[.typ] [kind] form of invocation and used an invalid specification for file kind.

User Action: Reenter the command with a valid file kind specification: DES (the default) for description files created using RMSDES, or DAT for data files.

?DES-F-IKV, Illegal key number

Explanation: You specified an illegal value for a key number.

User Action: Reenter the command specifying a positive number from 0 through 254.

?DES-F-ILA, Illegal attribute specified

Explanation: You specified an attribute keyword that is not a valid attribute.

User Action: Reenter the command using a valid attribute keyword; make sure the keyword is spelled correctly.

?DES-F-ILC, Illegal command

Explanation: You entered an invalid command.

User Action: Reenter the command using a valid command name; make sure the command is spelled correctly.

?DES-F-ILN, Illegal response given, must be numeric

Explanation: You entered a nonnumeric value in response to a prompt for number of keys, areas, or segments.

User Action: Enter a valid numeric value in response to the prompt. The range for keys and areas is 1 through 255. The range for segments is 1 through 8.

?DES-F-IPC, Illegal protection class

Explanation: You specified an illegal category of user.

User Action: Enter a valid user category: SYSTEM, OWNER, GROUP, or WORLD.

?DES-F-IPO, Illegal protection option specified

Explanation: You specified an illegal access type.

User Action: Enter a valid access type: R, W, E, or D (representing Read, Write, Extend, or Delete; on VAX/VMS systems, E represents Execute).

?DES-F-IRO, Illegal response, must be octal numeric

Explanation: You used a non-octal number in the UIC you specified in response to the Owner attribute prompt.

User Action: Enter a valid UIC.

?DES-F-ISE, Illegal section specified

Explanation: You specified an invalid section keyword.

User Action: Reenter the command using a valid section keyword; make sure the keyword is spelled correctly.

?DES-F-IVS, Illegal value specified

Explanation: You entered an illegal value in response to an attribute prompt.

User Action: Enter a legal value; see Section 2.6 for descriptions of the attributes and their legal values.

?DES-F-KMA, Key missing corresponding area

Explanation: You defined keys and assigned key levels to areas but did not define the areas. When you issued the CREATE command, RMSDES could not define the areas.

User Action: Define the area(s) to correspond to your key definitions.

?DES-F-KNS, Key has not been defined

Explanation: You specified a key that has not been defined.

User Action: Reenter the command specifying the number of a defined key. Use the SHOW KEY ALL command to display the defined keys.

?DES-F-KOR, Key number out of legal range

Explanation: Keys are numbered from 1 through 255. You specified a key outside of this range.

User Action: Reenter the command specifying a valid key number.

?DES-F-NAN, No area number given

Explanation: You issued a CLEAR, HELP, SET, or SHOW command for an area but did not specify an area number.

User Action: Reenter the command specifying a valid area number.

?DES-F-NKN, No key number given

Explanation: You issued a CLEAR, HELP, SET, or SHOW command for a key but did not specify a key number.

User Action: Reenter the command specifying a valid key number.

?DES-F-NUS, Not a user settable attribute

Explanation: You tried to set the Source attribute. This attribute is not user settable.

User Action: None. RMSDES automatically notes the user's (source) system.

?DES-F-VOR, Value out of legal range

Explanation: You entered a value in response to an attribute prompt that was not in the legal range of values for that attribute.

User Action: Enter a value within the legal range; see Section 2.6 for descriptions of the attributes and their legal values.

CHAPTER 3

RMS-11 INDEXED FILE LOAD UTILITY: RMSIFL

The RMS-11 Indexed File Load Utility (RMSIFL) reads records from any type of RMS-11 file and loads them into an RMS-11 indexed file created to store them. In addition, you can use RMSIFL to reestablish areas of contiguity to indexed files (that is, to compress indexed files) and to convert ASCII stream files to RMS-11 indexed files.

Unlike RMSCNV (see Chapter 4), RMSIFL does not use the standard RMS-11 access methods to build the output file. Rather, RMSIFL exploits the basic structure of indexed files (see the RSX-11M/M-PLUS RMS-11 User's Guide) to quickly insert data records and construct indexes. As it does so, RMSIFL creates optimally dense buckets at all levels of each index.

RMSCNV, on the other hand, optimizes only the primary index when the input file is sorted by the output file's primary key. However, you must use RMSCNV, not RMSIFL, if you do not have room for RMSIFL's sort work files.

RMSIFL reads records from an input file and loads them into an indexed output file in six phases:

1. RMSIFL examines the command string for errors and determines whether the input file is to be sorted. If the input file resides on a serial, nondisk device (magnetic tape, for example), RMSIFL will capture the input file on disk.

2. If the input file is to be sorted, RMSIFL will read the input file in sequential access mode, extract the value for the primary key and the record's file address (RFA) for each record, and write the primary key and RFA to a temporary file. The key values will then be sorted in ascending order.

3. RMSIFL loads the ordered records into the output file, constructing level 0 of the primary index a bucket at a time and examining each record for compatibility with the output file. Incompatible records are processed as "exception records" (see Section 3.4).

   RMSIFL also extracts the values for each of the alternate keys, if any, defined for the file and writes them to a temporary file. Finally, RMSIFL assembles the index levels of the primary index.

4. RMSIFL sorts the values for each of the alternate keys in the temporary file by key number and ascending key value. The keys are sorted one at a time into another temporary file.

5. RMSIFL builds the alternate indexes.

6. When processing is finished, RMSIFL prints closing messages and terminates.

Sections 3.4 and 3.5 describe these six processing phases in greater detail.

The following sections describe how to invoke and terminate RMSIFL (Section 3.1) and the RMSIFL command string and switches (Sections 3.2 and 3.3). Section 3.6 provides examples illustrating the use of RMSIFL. Appendix C, Section C.2, documents RMSIFL messages.

3.1 INVOKING AND TERMINATING RMSIFL

The RMSIFL utility can be either installed or uninstalled. If RMSIFL is installed, invoke it with the command:

```
IFL [command-string]
```

If you include a command string, RMSIFL will execute the command and return the system prompt.

If you do not include a command string, RMSIFL will prompt for one:

```
IFL>
```

If RMSIFL is uninstalled on RSX-11M/M-PLUS systems, invoke it with the MCR command:

```
RUN $RMSIFL
```

RMSIFL will respond with the prompt:

```
IFL>
```

In response to the prompt, you can type a command string or you can type CTRL/Z to terminate the utility. When RMSIFL has executed a command string, it returns the IFL> prompt. Section 3.2 describes the RMSIFL command string.

Indirect Command Files. You can execute series of RMSIFL command strings by means of indirect command files. The command strings in an indirect command file are executed when an at sign (@) followed by the file's name is specified to the utility. The default file type of .CMD is assumed if none is provided.

Installed RMSIFL will accept indirect command files either from the system prompt or from the utility prompt. For example, to execute the command strings contained in the indirect command file IFLCOM.CMD, you could type either of the following commands:

```
IFL @IFLCOM
   or
IFL> @IFLCOM
```

Uninstalled RMSIFL will accept indirect command files only from the utility prompt. For example:

```
IFL> @IFLCOM
```

If you executed the indirect command file from the system prompt, the system prompt will be returned. If you executed the indirect command file from within RMSIFL, the IFL> prompt will be returned.

3.2 RMSIFL COMMAND STRING

Format

```
outfile[/switch...]=infile[/switch...]
```

outfile

The specification of an RMS-11 indexed file that is to receive the records of the input file.

Wild card characters are not permitted in any field of the specification; the default version is the highest version.

The file must exist before you run RMSIFL, it must have no data in it other than the prologue, and it must have no more than 20 keys defined. Bucket size should generally be restricted to 5 or fewer blocks; however, the actual bucket size supported by RMSIFL will vary depending on such factors as number of keys, size of keys, record size, amount of memory available.

Note that RMSIFL may build an output file that is smaller than the input file (RMSCNV may do the same; however, the same input file processed by both RMSIFL and RMSCNV may result in different-size output files because of the different algorithms that the two utilities use to build files).

infile

The specification of the file from which records will be written to the output file. Wild card characters are not permitted in any field of the specification. The file can have any RMS-11 file organization but must have no more than 20 keys defined. You cannot use a back-up file produced by RMSBCK as the input file (see Chapter 6). RMSIFL ignores the fixed-control area of variable-with-fixed-control (VFC) records in the input file.

NOTE

When RMSIFL opens the input file, it allows the file to be accessed for reading but not for writing. Thus, no writing task can access the input file during RMSIFL processing.

/switch

One of the switches listed in Table 3-1 and described in Section 3.3.

In place of the command string, you can type HELP or ? to display help text about RMSIFL, or /ID to display the current version of the utility.

Table 3-1: RMSIFL Switches

Type	Switch	Description
Global	/ID	Identify current version Default: No identification
Output File	/ER[:file-spec]	Write primary keys of exception records to the terminal if no file-spec; or write exception records into the specified file Default: Write primary keys of exception records to the terminal
	/NOER[:S]	Stop processing if input record is incompatible Default: Write primary keys of exception records to the terminal
	/LO	Honor fill numbers Default: Fill buckets
	/PD[:[#]x]	Pad input records to output record length Default: Handle input as exception records if different lengths
	/TR	Truncate input records to output record length Default: Handle input as exception records if different lengths
Input File	/DE:dvn1:[dvn2:...dvn5:]	Reassign devices for sort work files Default: Create and use sort work files on SY:
	/KR:n	Key of reference number Default: Primary key (n=0)
	/NOSO	Do not sort records before loading Default: Sort records in input file before loading

3.3 RMSIFL SWITCHES

This section describes the global, output file, and input file switches that you can specify in RMSIFL command strings. Global switches are switches that can be specified anywhere in the command string.

3.3.1 Global Switch

The global switch is /ID.

/ID

Displays the current version number of the RMSIFL utility. For example:

```
IFL>OUTFIL.DAT/ID=INFILE.DAT
IFL -- Version 2.000
```

The last two digits of the version number (00 in this example) are the patch level of the utility. The /ID switch can also be specified by itself on the command line:

```
IFL>/ID
IFL -- Version 2.000
```

3.3.2 Output File Switches

The output file switches are /ER, /NOER, /LO, /PD, and /TR.

/ER[:file-spec]

Directs RMSIFL to continue processing if it encounters a record in the input file that cannot be written to the output file, that is, an exception record (see Section 3.4). If you specify a file, the exception records will be written into that file. If you do not specify a file, the primary key of each exception record will be written to the terminal. RMSIFL also issues error messages noting the reason for the exception record.

If you specify a nondisk file or a terminal, RMSIFL will create the file as an RMS-11 sequential file with variable-length records when it encounters the first exception record. Then, RMSIFL will write both the error message and the exception record's primary key to the file. RMSIFL will do this for every exception record until it finishes processing.

If you specify a disk file, RMSIFL will create the file as an RMS-11 sequential file with variable-with-fixed-control (VFC) records when it encounters the first exception record. Then, RMSIFL will write the exception record and a 4-byte exception code into the fixed-control area of the record.

The codes are:

Code	Meaning
001:	Record is out of primary key sequence
002:	Record contains an invalid primary key
003:	Record is too short to contain output primary key
004:	Record is too long to fit in output bucket
005:	Record is not correct size for output fixed-length record
006:	Record is too long for maximum record size
007:	Record contains an invalid packed decimal key

See also the /NOER switch.

By default, if you do not specify /ER, RMSIFL will write the appropriate error message and the exception record's primary key to your terminal.

/NOER[:S]

Directs RMSIFL to print an error message, delete the output file (unless you specify the save argument, :S), and terminate processing immediately if it encounters an input record that cannot be written into the output file (an exception record; see Section 3.4).

If you specify the save argument (:S), RMSIFL will not delete the output file. Termination at this point, however, leaves the output file incomplete: it is not a valid file of any organization and cannot be reused for any RMS-11 purpose. Using the save argument is useful, for example, when your output file is large and contiguous and you do not want to lose any of the contiguous space to other users.

By default, if you do not specify /NOER, RMSIFL will respond according to the form of the /ER switch specified (taking the /ER default of writing the primary key of each exception record to the terminal if /ER was not specified).

/LO

Directs RMSIFL to write records into buckets according to the bucket fill size established when the file was created (see the RSX-11M/M-PLUS RMS-11 User's Guide for more information on fill numbers).

By default, if you do not specify /LO, RMSIFL will insert as many records as possible into each bucket.

/PD[:[#]x]

Directs RMSIFL to pad records read from the input file to the output file's record length before writing them to the output file. Specify the padding character as follows:

Switch	Character
PD	NULL
PD:x	x is an ASCII character A-Z, 0-9, or any special character except #, ?, or @
PD:#x	x is an octal number 000-377 (for example, 40 for SPACE, 43 for #, 77 for ?, or 100 for @)

You can use the /PD switch only if the output file was created for fixed-length records.

When both the input and output files contain fixed-length records, RMSIFL requires either the /PD or /TR switch if the fixed-length records are not equal. If neither switch is specified, RMSIFL will terminate with the fatal error message:

?IFL -- Input and output fixed length record sizes unequal

When the input file contains variable-length records and the output file contains fixed-length records, RMSIFL may require both the /PD and /TR switches. If both are not specified, RMSIFL will terminate with the fatal error message:

?IFL -- /TR and/or /PD required with fixed length output records

By default, if you do not specify /PD and the input file's records are shorter than the output file's records, RMSIFL will treat the input records as exception records (see Section 3.4.3, and the /ER and /NOER switch descriptions in this section).

/TR

Directs RMSIFL to truncate records read from the input file to the output file's maximum record size before writing the input records to the output file. The trailing bytes of the record are truncated.

When both the input and output files contain fixed-length records, RMSIFL requires either the /TR or /PD switch if the fixed-length records are not equal. If neither switch is specified, RMSIFL will terminate with the fatal error message:

?IFL -- Input and output fixed length record sizes unequal

When the input file contains variable-length records and the output file contains fixed-length records, RMSIFL may require both the /TR and /PD switches. If both are not specified, RMSIFL will terminate with the fatal error message:

?IFL -- /TR and/or /PD required with fixed length output records

By default, if you do not specify /TR, RMSIFL will treat the input records as exception records (see Section 3.4.3, and the /ER and /NOER switch descriptions in this section).

3.3.3 Input File Switches

The input file switches are /DE, /KR, /NOSO.

/DE:dvn1:[dvn2:...dvn5:]

Directs RMSIFL to reassign the devices on which it allocates the three sort work files and the two primary and alternate key temporary files (see Section 3.4.2).

The argument dvn: is a physical device name and number.

If you specify /DE, RMSIFL will reassign all three sort work files and the two key temporary files, as follows:

- If you specify one device name, RMSIFL will reassign all five files to that device.
- If you specify two device names, RMSIFL will reassign one work file to the first device and four files (two work and two temporary) to the second.
- If you specify three device names, RMSIFL will reassign one work file to the first device, one work file to the second, and three files (one work and two temporary) to the third.
- If you specify four device names, RMSIFL will assign each of the three work files to the first three devices and both temporary files to the fourth.
- If you specify five device names, RMSIFL will assign one file to each device.

For optimal performance, you should assign each sort work file to a different device. For example:

```
IFL>OUTFIL.DAT=INFILE.DAT/DE:DM0:DM1:DM2:DM3:DM4:
```

However, if you have only two devices, assign the first, third, and fifth files to one device and the second and fourth files to the other device:

```
IFL>OUTFIL.DAT=INFILE.DAT/DE:DM0:DM1:DM0:DM1:DM0:
```

By default, if you do not specify /DE, RMSIFL will assign the sort work files and the key temporary files to SY:.

/KR:n

Directs RMSIFL to read an indexed input file according to the key of reference specified by n: n is 0 for the primary key, 1 for the first alternate key, and so on up to 19. The /KR switch can eliminate the need for RMSIFL to sort the input file (see Section 3.4.1).

You can specify /KR only with indexed input files. Otherwise, RMSIFL will terminate with a fatal error message.

By default, if you do not specify /KR and the input file is indexed, RMSIFL will read the file using the primary key as the key of reference.

/NOSO

Directs RMSIFL to bypass the primary key sorting phase of processing (see Section 3.4.1) because the input file is already sorted. If the input file is not in ascending order by the output file's primary key, RMSIFL's response will depend on whether you have specified /NOER or a form of /ER with the output file specification (see Section 3.3.2).

Note that RMSIFL will still sort the alternate keys even if /NOSO is specified.

By default, if you do not specify /NOSO, RMSIFL will sort all records from the input file into the proper order (see Section 3.4.2).

3.4 RMSIFL PROCESSING

RMSIFL reads records from an input file, sorts them (if necessary), and loads them into an empty indexed RMS-11 output file in six phases, described in the following sections.

3.4.1 Checking the Command String

RMSIFL first examines the command string for syntax and other errors. If it finds an error, it will print an error message and prompt for a new command string. When a valid command string is entered, RMSIFL determines whether the input file must be sorted in ascending order by the output file's primary key (the second phase of RMSIFL processing; see Section 3.4.2).

RMSIFL will bypass the primary key sorting phase if you have specified either the /NOSO or /KR input file switch (see Section 3.3.3). The /NOSO switch directs RMSIFL to bypass sorting because you have already sorted the input file.

The /KR switch specifies a key of reference for an indexed input file. If the key of reference specified is equal to or contains the output primary key, based on position only, RMSIFL will bypass the sorting phase. Nonstring key types must match exactly in starting position and length. String keys must start in the same position, but the output primary key can be the same length or shorter than the input key of reference.

For example, if the 10-byte string input primary key starts at byte 0 and the 8-byte string output primary key starts at byte 3, RMSIFL will sort the input file. However, if the 10-byte string input alternate key starts at byte 0 and is specified as the key of reference, and the 8-byte string output primary key starts at byte 0, RMSIFL will not sort the input file.

For another example, if the 3-byte packed decimal input alternate key 1 starts at byte 43 and is specified as the key of reference, and the 3-byte packed decimal output primary key starts at byte 42, RMSIFL will sort the input file.

RMSIFL notifies you that it has started processing by printing the message:

```
PRIMARY KEY
```

3.4.2 Reading and Sorting the Input Records

RMSIFL reads the input file in sequential access mode, depending on the organization of the input file:

Input File Organization	Processing
Sequential	RMSIFL reads the records in sequential access mode, starting with the first record in the file.
Relative	RMSIFL reads the records in sequential access mode, starting with record cell 1.
Indexed	RMSIFL reads the records in sequential access mode, following the key of reference.

RMSIFL passes the primary key from each record to the sort routines, which are provided with RMSIFL. These routines sort the keys into ascending order. At the beginning of this phase, RMSIFL initially allocates disk space for three sort work files (see Section 3.5). (You can use the /DE input file switch to specify the disks to which these files are assigned; see Section 3.3.3.)

RMSIFL indicates the start and finish of this phase by printing the messages:

```
SORT HAS STARTED
SORT MERGE PHASE HAS FINISHED
```

If an error occurs during the sorting, RMSIFL will terminate with the message:

```
SORTS ERROR CODE IN OCTAL:nn
```

See Appendix D for a description of the sort routine error codes.

At the end of this phase, RMSIFL deletes the sort work files. As output from this phase, RMSIFL creates a file, IFLP.TMP, containing the RFA from each record in the input file, sorted in ascending order by primary key. See Section 3.5.

3.4.3 Loading Records in the Output File

RMSIFL loads the ordered input records into the output file, constructing level 0 of the primary index (the data level) a bucket at a time (see the RSX-11M/M-PLUS RMS-11 User's Guide for information on the primary index).

If the records were sorted in the previous processing phase, RMSIFL will read the RFA for each record from IFLP.TMP, then read the associated record, by RFA, from the input file. IFLP.TMP is deleted at the end of this phase.

If the records were not sorted in the previous processing phase (see Section 3.4.1), RMSIFL will read the records in sequential order from the input file.

RMSIFL examines each input record to ensure that it is compatible with the output file. If the record is not compatible, RMSIFL processes it as an exception record according to the /ER or /NOER input file switch specified in the command string (see Section 3.3.3). Exception records can occur for any of the following reasons:

- You specified the /NOSO input file switch and RMSIFL found a record that was not in ascending order by the output file's primary key.
- You did not allow duplicates in a key of the output file, but an input file record contains a duplicate value in that key field.
- An input record is not long enough to contain the primary key of the output file.
- An input record is too long to fit in a bucket of the output file.

- An input record is not compatible with the output file's fixed-length record format and you did not specify the /PD and/or the /TR switch.

- An input record is longer than the output file's variable-length maximum record size and you did not specify the /TR switch.

- An input record contains an invalid packed decimal key: a digit half-byte is greater than 9 or the sign half-byte is less than 10.

In all these cases, RMSIFL issues an appropriate nonfatal error message noting that an exception record has occurred and why, and continues processing.

If RMSIFL encounters an exception record while processing alternate index(es), it flags as deleted the data records in the primary index. RMSIFL does not remove the record from the file. The record can be compressed during later use of the indexed file (see the RSX-11M/M-PLUS RMS-11 User's Guide).

You can add corrected exception records to the output file after RMSIFL terminates by using either an application program or RMSCNV (possibly using your terminal as the input device; see Chapter 4).

If the record being read is compatible, RMSIFL inserts it into a level-0 bucket being built in memory.

As RMSIFL places each record in the output file, it extracts the values for each of the alternate keys defined for the file, if any:

- RMSIFL combines the alternate key value with the record file address (RFA) of the user data record that supplied the key value. RFAs are 6 bytes long. RMSIFL then appends a byte with the key number.

- RMSIFL writes the alternate key records into a separate temporary file, IFLA.TMP, which is marked for deletion. RMSIFL allocates disk space for this file as described in Section 3.5.

During this phase, RMSIFL also assembles the index levels of the primary index. RMSIFL puts an index record into a level-1 bucket for each level-0 bucket it writes to the file. When it writes a level-1 bucket, RMSIFL inserts an index record in a level-2 bucket and so on up to and including the root.

RMSIFL indicates the start and finish of this phase by printing the messages:

```
LEVEL-0 LOADING HAS STARTED
LOADING HAS FINISHED
```

3.4.4 Sorting the Alternate Key Temporary File

RMSIFL sorts the alternate key temporary file by key number and ascending key value, using the same sort routines as in phase 2 (see Section 3.4.2). Nonstring key data types are converted to a string-like representation for sorting only.

If there is more than one alternate key, RMSIFL will sort one at a time, building each alternate index after that index's keys have been sorted (see Section 3.4.5).

For each alternate key, RMSIFL allocates the disk space for three sort work files, as described in Section 3.5.

RMSIFL indicates the start and finish of this phase by printing the message:

```
ALTERNATE KEY(S)
```

At the end of each alternate key sort, RMSIFL deletes the sort work files. As output from each sort, RMSIFL creates a temporary file, IFLA1.TMP, which contains the key values and RFAs for an alternate key.

3.4.5 Building the Alternate Indexes

RMSIFL builds the alternate indexes one at a time from level 0 up to the root, using the sorted temporary file, IFLA1.TMP, for each key for input.

RMSIFL prints the following message as it constructs each alternate index:

```
KEY nnn
```

The variable nnn starts at 1.

3.4.6 Terminating Processing

RMSIFL prints the following messages when the output file is completely loaded, then terminates processing of the current command string:

```
NUMBER OF INPUT RECORDS:  nnnnnnnnn
NUMBER OF OUTPUT RECORDS:  nnnnnnnnn
NUMBER OF EXCEPTION RECORDS:  nnnnnnnnn
```

NOTE

RMSIFL extends the output file if it cannot contain the input records and the index(es). The utility minimizes this overhead by extending the file a set quantity each time it requires space. This quantity is the integral multiple of bucket size less than or equal to 50.

3.5 CALCULATING TEMPORARY FILE WORK SPACE

If the input file is on a serial, nondisk device, RMSIFL must capture the file on disk before it can process the file. The file captured on disk is the same size as the input file. This file is deleted after the primary index is built.

At the beginning of phase 2 (see Section 3.4.2), RMSIFL allocates disk space for three sort work files, calculating the space for each as follows:

1.5 * NDRF * (6 + PKS + PAD)/512.

or

1/8 the size of the input file, whichever is larger.

NRDF is the number of user data records in the input file.

PKS is the size of the primary key, in bytes (plus 1 for packed decimal keys).

PAD is the number of null bytes added to word-align the record:

PAD = 0 if PKS is an even number

PAD = 1 if PKS is an odd number

The output from this phase, IFLP.TMP, is allocated disk space as follows:

6 * NRDF/512.

In phase 3, if alternate keys are defined for the output file (see Section 3.4.3), RMSIFL will create the temporary file IFLA.TMP with an initial allocation of 100 blocks and a default extension quantity of 100 blocks. The final size of this temporary file is calculated as follows:

(NDRF * NAK * (7 + LKS + PAD))/512.

NDRF is the number of user data records in the input file.

NAK is the number of alternate keys defined for the output file.

LKS is the size of the largest alternate key defined for the output file, in bytes (plus 1 for packed decimal keys).

PAD is the number of null bytes added to word align the record:

PAD = 0 if PKS is an even number

PAD = 1 if PKS is an odd number

In phase 4, RMSIFL sorts the alternate keys one at a time (see Section 3.4.4). To sort each key, RMSIFL creates three sort work files, calculating the disk space for each as follows:

1.5 * NDRF * (6 + AKS + PAD)/512.

NDRF is the number of user data records in the input file.

AKS is the size of the alternate key, in bytes (plus 1 for packed decimal keys)

PAD is the number of null bytes added to word align the record:

PAD = 0 if PKS is an even number

PAD = 1 if PKS is an odd number

The output from each alternate key sort, IFLA1.TMP, is allocated disk space as follows:

NDRF * (6 + AKS)/512.

Note that if the /DE output file switch is specified (see Section 3.3.3), IFLA1.TMP will be assigned to the same device to which IFLP.TMP was assigned.

3.6 RMSIFL EXAMPLES

```
>IFL PAYROL.DAT=PAYROL.SEQ
        Primary key
    Sort has started
    Sort merge phase has finished
    Level-0 loading has started
    Loading has finished
        Alternate key(s)
        Alternate key 1

    Number of input records: 448
    Number of output records: 448
    Number of exception records: 0
```

In this example, RMSIFL loads a sequential input file into an empty output indexed file. RMSIFL sorts the input records and reports the total number of records processed.

```
>IFL VHC.INX/TR/NOER=VHC.SEQ
        Primary key
    Sort has started
    Sort merge phase has finished
    Level-0 loading has started
    Loading has finished
        Alternate key(s)
        Alternate key 1

    Number of input records: 11
    Number of output records: 11
    Number of exception records: 0
```

RMSIFL sorts the input records from the file VHC.SEQ and truncates them to fit the maximum record size of the output file records in VHC.INX. There are no exception records, so the RMSIFL command completes successfully.

```
>TYP VHCKR0.DES
GET VHC.INX DAT
SET FILE NAME VHCKR0.INX
SET FILE SUPERSEDE YES
SHOW ALL
CREATE
>DES @VHCKR0.DES
SYSTEM
  Target                    RSX
  Source                    RSX
```

```
FILE
  Name                      VHCKR0.INX  (Explicit)
  Organization              INDEXED
  Allocation                62
  Extension                 1
  Bucket_Size               4
  Protection                (SYSTEM:RWED, OWNER:RWED, GROUP:RWED, WORLD:R   )
  Owner                     [70,5]
  Max_Record_Number         0
  Contiguous                NO
  Supersede                 YES  (Explicit)
RECORD
  Size                      80
  Format                    FIXED
  Control_Field_Size        0
  Carriage_Control          CARRIAGE_RETURN
KEY    0
  Name                      NPANXX
  Type                      STRING
  Null_Key                  NO
  Null_Value                0
  Duplicates                NO
  Changes                   NO
  Seg0_Position             31
  Seg0_Length               6
  Data_Fill                 50
  Data_Area                 0
  Index_Fill                50
  Level1_Index_Area         0
  Index_Area                0
KEY    1
  Name                      PLACE
  Type                      STRING
  Null_Key                  NO
  Null_Value                0
  Duplicates                YES
  Changes                   YES
  Seg0_Position             59
  Seg0_Length               12
  Data_Fill                 50
  Data_Area                 0
  Index_Fill                50
  Level1_Index_Area         0
  Index_Area                0
AREA    0
  Allocation                40
  Extension                 0
  Bucket_Size               4
  Contiguous                NO
  Position                  NONE  0
  Exact_Positioning         NO
>IFL VHCKR0.INX/ER:VHC.ERR=VHC.INX/KR:1/NOSO
        Primary key
    Level-0 loading has started
    Loading has finished
        Alternate key(s)
        Alternate key 1

    Number of input records: 11
    Number of output records: 6
    Number of exception records: 5
```

First, RMSDES is used to create an indexed file, VHCKR0.INX.

Note that a RMSDES GET command is used to get the file attributes of the VHC.INX file: the contents of the DES GET file are shown initially, and then the DES file is invoked.

Next, RMSIFL takes the input records from the indexed file VHC.INX, in the order of the first alternate key, and puts them in the output indexed file.

Because the /NOSO switch is specified, RMSIFL assumes that the records are ordered properly already. In this case, five exception records are generated and placed in the file VHC.ERR because the records are actually out of sequence for the primary key.

CHAPTER 4

RMS-11 FILE CONVERSION UTILITY: RMSCNV

The RMS-11 File Conversion Utility (RMSCNV) reads records from an RMS-11 file of any type and loads them into another RMS-11 file of any type. For example, RMSCNV can:

- Create a new sequential file from the records of an existing sequential, relative, or indexed file
- Supersede an existing sequential file with the records of another sequential file, or the records of a relative or indexed file
- Append the records of one sequential file to another sequential file, or the records of a relative or indexed file to a sequential file
- Write the records of a sequential, relative, or indexed file into another file of any organization

On operating systems with DECnet network capabilities and on which RMSCNV has Data Access Protocol (DAP) support, RMSCNV file conversions can take place over the network. That is, in addition to local file conversions, you can convert files between a local and a remote node, or between two remote nodes.

In Figure 4-1, the user is running RMSCNV on an RSX-11M/M-PLUS system to convert a file on a RSTS/E system to a new file on a VAX/VMS system. The /CA switch creates the output file on the VAX/VMS system with the same attributes as the input file and copies the input file records into it. The /SL switch writes the summary listing to a file on the RSX-11M/M-PLUS system under the user's UIC.

See Section 4.3 for more information about the /CA and /SL switches. See the RSX-11M/M-PLUS RMS-11 User's Guide for more information about file specifications.

RMSCNV uses the standard RMS-11 access methods to build the output file. Unlike RMSIFL (see Chapter 3), RMSCNV optimizes only the primary index when the input file is sorted by the indexed output file's primary key. However, RMSCNV allows you to add records from an input file to an output file that already contains records (RMSIFL loads only empty output files).

In addition, RMSCNV can read and write records in block mode.

```
CNV>NODEV"USER USER"::DBA1:[USER]OUT.DAT/CA/SL:SUM.LIS=-
_CNV>NODER"100,100 DEMO"::DB1:[100,100]IN.DAT
```

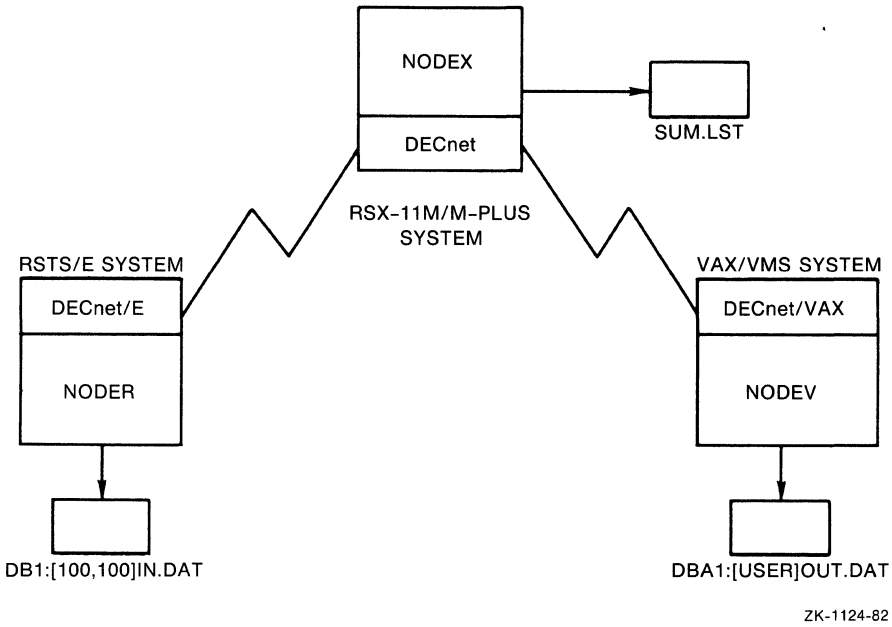

Figure 4-1: Using RMSCNV with DECnet

You can use RMSCNV to perform the following operations:

- Copy any RMS-11 file to magnetic tape. RMSCNV will convert the file to a sequential file. You can recopy the file to disk by redefining the original file using RMSDES (see Chapter 2) and then rerunning RMSCNV (or running RMSIFL for indexed files) to load the records from the tape.

- Print any RMS-11 file with carriage-return control. RMSCNV can convert the file for output directly to a line printer, or it can convert the file's records to variable format with carriage return control so the file can be queued to a line printer.

NOTE

RMS-11 does not convert numeric or binary data to ASCII data.

- Reestablish areas of contiguity to files restored by RMSRST. Use RMSDES to create the file with the attributes you want and use RMSCNV to populate the file with the single-area, noncontiguous file produced by RMSRST (for indexed files, it is suggested that you use RMSIFL; see Chapter 3).

- Convert ASCII stream files to RMS-11 file organizations by first creating the new file using RMSDES. When possible, if you are building a relative or indexed file, sort the ASCII file by relative record number or by primary key. Then build the new file using RMSCNV.

For example, one application of RMSCNV might be as follows: daily processing is done using a small file designed for that purpose. At the end of the day, RMSCNV can be used to add the records in the small file to a larger, master file. In addition, RMSCNV's ability to convert ASCII stream files to RMS-11 file organizations can be useful in intersystem migration (for example, RSTS/E to RSX-11M).

All RMSCNV functions involve the movement of records from one file to another. The manner in which RMSCNV reads records from the specified input file and writes them to the specified output file depends on the file organization of both the input and output files and on the switches you specify in the command string. Section 4.4 describes RMSCNV processing in greater detail.

The following sections describe how to invoke and terminate RMSCNV (Section 4.1) and the RMSCNV command string and switches (Sections 4.2 and 4.3). Section 4.5 provides examples illustrating the use of RMSCNV. Appendix C documents messages.

4.1 INVOKING AND TERMINATING RMSCNV

The RMSCNV utility can be either installed or uninstalled. If RMSCNV is installed, invoke it with the command:

```
CNV [command-string]
```

If you include a command string, RMSCNV will execute the command and return the system prompt.

If you do not include a command string, RMSCNV will prompt for one:

```
CNV>
```

If RMSCNV is uninstalled on RSX-11M/M-PLUS systems, invoke it with the MCR command:

```
RUN $RMSCNV
```

Note, however, that uninstalled RMSCNV will respond with a prompt that reflects the current process name (abbreviated to 3 characters). For example:

```
RUN $RMSCNV
>TT1:
```

You can install RMSCNV with another name as follows:

```
INS $RMSCNV/TASK=...xxx
```

where xxx is a 3-character name that you chose for the utility. The utility will then prompt:

```
xxx>
```

RMSCNV will respond with the prompt:

```
CNV>
```

In response to the prompt, you can type a command string or you can type CTRL/Z to terminate the utility. When RMSCNV has executed a command string, it returns the CNV> prompt. Section 4.2 describes the RMSCNV command string.

Indirect Command Files. You can execute series of RMSCNV command strings by means of indirect command files. The command strings in an indirect command file are executed when an at sign (@) followed by the file's name is specified to the utility. The default file type of .CMD is assumed if none is provided.

Installed RMSCNV will accept indirect command files either from the system prompt or from the utility prompt. For example, to execute the command strings contained in the indirect command file CNVCOM.CMD, you could type either of the following commands:

```
CNV @CNVCOM
   or
CNV> @CNVCOM
```

Uninstalled RMSCNV will accept indirect command files only from the utility prompt. For example:

```
CNV> @CNVCOM
```

If you executed the indirect command file from the system prompt, the system prompt will be returned. If you executed the indirect command file from within RMSCNV, the CNV> prompt will be returned.

4.2 RMSCNV COMMAND STRING

On systems with DECnet network capabilities and on which RMSCNV has DAP support, both the output file and input file can be on remote network nodes. See the RSX-11M/M-PLUS RMS-11 User's Guide for more information on file specifications.

If either the input file or the output file is relative or indexed, the bucket size should, in general, be no greater than 15 blocks. The actual allowable bucket size for either file, however, will depend on many factors directly related to processing, such as: maximum record size for either file, number of keys for either file, whether /SL and/or /IM is specified, whether the network is configured, and so forth.

Note that if the maximum record sizes for the input and output files are different, you may have to specify the /PD or /TR switch or both (see Section 4.3).

Format

```
[outfile[/switch...]=]infile[/switch...]
```

outfile

The specification of the file that is to receive the records of the input file. Wild card characters are not permitted in any field of the specification.

RMSCNV will automatically create a sequential output file if one does not already exist.

Note, however, that when you specify the /EO switch, an output file specification is not allowed, and no output file will be created. See Section 4.3 for more information on the /EO switch. The default version for existing sequential output files (those specified with the /AP or /SU switch; see Section 4.3), for relative files, and for indexed files is the highest version; if an existing sequential file is specified and neither the /AP nor the /SU switch is specified, RMSCNV will use the highest existing version number plus 1.

A relative or indexed output file must exist before RMSCNV is invoked (unless you specify the /CA switch; see Section 4.3). You can create the file using either an application program or the RMSDES utility (see Chapter 2).

RMSCNV may build an output file that is smaller than the input file. (RMSIFL may do the same; however, the same input file processed by both RMSCNV and RMSIFL may result in different-size output files because of the different algorithms that the two utilities use to build files.)

You can specify a unit record device, such as a terminal or a line printer, as the output file. Specify the current user's terminal as TI:. By default, if no output file is specified, the records of the input file are written to your terminal (that is, TI: is the default).

infile

The specification of the file from which records will be written to the output file. Wild card characters are not permitted in any field of the specification.

You cannot use a back-up file produced by RMSBCK as the input file (see Chapter 6), except to copy the file using the /IM switch (see Section 4.3).

You can specify a unit record device, such as a card reader or terminal, as the input file. Specify the current user's terminal as TI:. End terminal input with CTRL/Z.

/switch

One of the switches listed in Table 4-1 and described in Section 4.3. All switches are global; that is, they can be specified anywhere in the command string.

In place of the command string you can type HELP or ? to display help text about RMSCNV, or /ID to display the current version of the utility.

Table 4-1: RMSCNV Switches

Switch	Description
/AP	Append records to an existing sequential file Default: No appending
/BL[:n]	Set magnetic tape block size Default: 512 bytes
/CA[:file-spec]	Create an output file with the attributes of an existing input file Default: Output file must exist or a sequential file will be created
/EO	Convert CTRL/Z end-of-file character in an ASCII stream file to null and pad the file with nulls to the physical end of file Default: Assume null-filled stream file
/FO:x	Set output file organization, where x is S, R, or I Default: Sequential (S)
/ID	Identify current version Default: No identification
/IM	Process files in block mode Default: Use standard RMS-11 access methods
/KN:["]keyname["]	Read an indexed file using the key of reference specified by **keyname** Default: Read file using primary key
/KR:n	Read an indexed file using the key of reference specified by **n** Default: Read file using primary key (0)
/LO	Honor bucket fill size when filling buckets in the output indexed file Default: Fill buckets to capacity
/MA	Use mass-insertion mode and sequential PUT operations Default: No mass insertion; random PUT operations

(continued on next page)

Table 4-1 (Cont.): RMSCNV Switches

Switch	Description
/ML:n	Explicitly set limit of buffer allocation Default: Allow RMSCNV to calculate the amount of memory available for allocation
/PD[:[#]["]x["]]	Pad input records to output record length Default: No record padding
/SL[:file-spec]	Provide summary listing (in file, if specified) Default: No summary
/SU	Supersede existing sequential file Default: No supersession
/TR	Truncate input records to output record length Default: No record truncating
/WF	Write or read fixed-control area Default: Ignore fixed-control area

4.3 RMSCNV SWITCHES

/AP

Directs RMSCNV to append records to an existing sequential output file. That is, RMSCNV adds the input file records to the end of the output file. You cannot specify /AP and /SU in the same command string. If the output file is not sequential, RMSCNV will ignore the /AP switch.

By default, if you do not specify /AP, RMSCNV will respond according to whether you have specified /SU:

- If you specified /SU, RMSCNV supersedes the existing sequential output file.
- If you did not specify /SU and the output file exists, RMSCNV will create the next higher version of the file.

See also the description of the /SU switch.

/BL[:n]

Specifies the physical block size of the output file when it is being created on magnetic tape. The argument **n** must be a number from 18 to 8192 (decimal). If you do not specify **n**, RMSCNV will use a tape block size of 512 (decimal) characters. You can specify /BL only with sequential tape files. If you specify /BL with other files, RMSCNV will return the following error message:

```
%CNV-F-IAS, Illegal use of /AP, /BL, or /SU with file
            organization
```

By default, if you do not specify /BL, RMSCNV will use a tape block size of 512 (decimal) characters.

/CA[:file-spec]

Directs RMSCNV to create an output file with the attributes of an existing file and to copy the records of the input file into it.

If you specify a file with /CA, RMSCNV will create the output file with the attributes of the specified file.

If you do not specify a file with /CA, RMSCNV will create the output file with the attributes of the input file.

Note, however, that if the existing file is indexed and contains multiple areas, these areas will not be preserved in the newly created file. The new file will be a single-area file. Its bucket size will be that of the largest bucket size in the multiple-area file. Bucket fill size also will not be preserved. RMSCNV will insert as many records as possible into each bucket. Two informational messages will be issued when this occurs:

```
%CNV-W-OOA, Output file will contain only one area

%CNV-W-FQL, Output indexed file fill quantities will not be
            preserved
```

By default, if you do not specify /CA, the requirements and defaults for output file specifications apply (see Section 4.2).

/EO

Converts a CTRL/Z end-of-file character in ASCII stream files to a null character (ASCII 000) and fills the remainder of the file, to the physical end-of-file, with null characters.

Note that when you specify /EO, an output file specification is not allowed, and no output file will be created. /EO operates directly on the input file. For example:

```
CNV>INFILE.DAT/EO
```

After processing, INFILE.DAT will be null-filled to end-of-file.

If the CTRL/Z signals both end-of-record and end-of-file, the CTRL/Z is retained as the end-of-record character and the remainder of the file is null-filled.

NOTE

RMS-11 no longer recognizes CTRL/Z as an end-of-file character. Therefore, ASCII stream files that end with CTRL/Z should be converted, using this switch, to the format recognized by RMS-11 (null-filled). If you specify a file that ends with CTRL/Z and do not specify /EO, the CTRL/Z character will become part of the data in the file; it will no longer signal end-of-file.

By default, if you do not specify /EO, RMSCNV will assume that the input stream file is padded with null characters from logical to physical end-of-file.

/FO:x

Specifies the organization of the output file. The argument **x** must be one of the following:

S -- sequential file
R -- relative file
I -- indexed file

By default, if you do not specify /FO, RMSCNV will use sequential organization and create an output file with the attributes of the input file (see Section 4.4.1).

/ID

Displays the current version of the RMSCNV utility. For example:

```
CNV>OUTFIL.DAT/ID=INFILE.DAT
%CNV-I-ID, Version 2.000
```

The last two digits of the version number (00 in this example) are the patch level of the utility. The /ID switch can also be specified by itself on the command line:

```
CNV>/ID
%CNV-I-ID, Version 2.000
```

/IM

Directs RMSCNV to read the records of the input file and write the records to the output file block by block, that is, in block access mode.

You can use the /ML switch with /IM to specify the number of blocks to be processed in a single READ or WRITE operation. See the description of the /ML switch.

By default, if you do not specify /IM, RMSCNV will convert files using the standard RMS-11 access methods.

/KN:["]keyname["]

Directs RMSCNV to read an indexed file based on the key named in the argument **keyname** (key name assignment is an optional capability provided by RMSDES; see Chapter 2).

The quotation marks are required only if the key name contains spaces, special characters, or lowercase letters.

You can specify /KN only for indexed files. Otherwise, RMSCNV will terminate with the fatal error message:

?CNV-F-IAX, Illegal use of /KN, /KR, /LO, or /MA with file organization

See also Section 4.4.2 for a way to use /KN to examine an indexed input file at your terminal.

You cannot specify /KN and /KR in the same command string.

By default, if you do not specify /KN (and you also have not specified /KR) and the input file is indexed, RMSCNV will read the file using the primary key as the key of reference.

/KR:n

Directs RMSCNV to read an indexed input file based on the key numbered **n**: **n** is 0 for the primary key, 1 for the first alternate key, and so on.

You can specify /KR only with indexed input files. Otherwise, RMSCNV will terminate with the fatal error message:

?CNV-F-IAX, Illegal use of /KN, /KR, /LO, or /MA with file organization

See also Section 4.4.2 for a way to use /KR to examine an indexed input file's contents.

You cannot specify /KR and /KN in the same command string.

By default, if you do not specify /KR (and you also have not specified /KN) and the input file is indexed, RMSCNV will read the file using the primary key (0) as the key of reference.

/LO

Directs RMSCNV to write records into buckets of an indexed file according to the bucket fill sizes established when the file was created. See the RSX-11M/M-PLUS RMS-11 User's Guide for more information on bucket fill sizes. You can specify /LO only with indexed output files. Otherwise, RMSCNV will terminate with the following error message:

?CNV-F-IAX, Illegal use of /KN, /KR, /LO, or /MA with file organization

By default, if you do not specify /LO, RMSCNV will insert as many records as possible into each bucket.

/MA

Directs RMSCNV to use mass-insertion mode (see the RSX-11M/M-PLUS RMS-11 User's Guide) and sequential PUT operations to insert

records into an indexed output file. Your indexed input file must be sorted in ascending order by the output file's primary key. Otherwise, RMSCNV will terminate with the fatal error message:

```
?CNV-F-ER$SEQ, Record not in proper key sequence for file
               <file-spec>
```

Mass-insertion mode provides improved performance only if records are to be inserted at the logical end-of-file. Therefore, either the output file should be empty, or the input file should contain records with primary key values greater than the primary key value of the highest record that currently exists in the output file. If the input records do not meet this requirement, RMSCNV will continue to process them. However, it will not achieve the improved performance otherwise possible with mass-insertion mode.

For example, suppose you have two indexed files, IDX.IN and IDX.OUT. The primary key of IDX.IN is defined as bytes 0 through 15 of each record, and the primary key of IDX.OUT is defined as bytes 0 through 8 of each record. Because RMS-11 orders records in indexed files according to primary key value, you can convert IDX.IN to IDX.OUT using the /MA switch with the default key of reference.

Note that if the primary key of IDX.OUT is the same as an alternate key of IDX.IN, you can still use the /MA switch if you specify the proper alternate key with the /KR input file switch (see Section 4.3.3).

You can specify /MA only with indexed output files. Otherwise, RMSCNV will terminate with the following error message:

```
?CNV-F-IAX, Illegal use of /KN, /KR, /LO, or /MA with file
            organization
```

By default, if you do not specify /MA, RMSCNV will use random PUT operations to insert each record in the output file.

/ML:n

If you are using block access mode (that is, also specifying /IM), /ML specifies the number of blocks to be processed in a single READ or WRITE operation. (See also the description of the /IM switch.)

If you are using the standard RMS-11 access modes, /ML specifies the amount of memory available to be allocated to all buffers for the conversion operations, where **n** specifies K words (1K=1024).

You can use this switch to override RMSCNV's memory calculation routines and explicitly specify memory allocation.

You can specify any positive number for **n**; if **n** is too large for the amount of memory available, RMS-11 will issue a fatal error message:

```
?CNV-F-MLE, Block value of /ML switch exceeds limit or
            available memory
```

If **n** is too small or if you specify /ML:1, RMS-11 will allocate minimal amounts of memory. If **n** is 0, RMSCNV's memory calculation routines will be used (this is the same as not specifying /ML).

By default, if you do not specify /ML, RMSCNV's memory calculation routines will be used to calculate the amount of memory available for the conversion operations.

/PD[:[#]["]x["]]

Directs RMSCNV to pad records read from the input file. You use /PD when the output file has fixed-length records and a maximum record size greater than the input file's maximum record size. (The input file's records can be either variable or fixed length.)

In addition, when the output file contains fixed-length records, the input file contains variable-length records, and the output file's maximum record size is less than the input file's, you must specify both /PD and /TR. If both are not specified, RMSCNV will terminate with the fatal error message:

?CNV-F-TPM, Switch /TR or /PD or both are needed for this CONVERT

(See also the description of the /TR switch.)

Specify the padding character as follows:

Switch	Character
/PD[:]	NULL (byte value 000).
/PD:[#]x	**x** is an octal number 000 to 377 (43 for #, 77 for ?, and 100 for @). The number sign (#) is optional; RMSCNV will check first for an octal number.
/PD:["]x["]	**x** is an ASCII character A to Z, 0 to 9, or any special character. The quotation marks are required for #, lowercase letters, or a space; otherwise, the quotation marks are optional. If the quotation marks are not present, RMSCNV will check first for an octal number; then it will check for an ASCII character.

By default, if you do not specify /PD and the input file's records are shorter than the output file's records, RMSCNV will terminate with the fatal error message:

?CNV-F-MRS, Input and output record sizes do not correspond

/SL[:file-spec]

Directs RMSCNV to produce a summary of file conversion processing. The summary listing consists of the following:

- The options (switches) in effect
- The input and output file organization, file attributes, record format, and record attributes
- A summary of messages produced during RMSCNV execution
- The number of input records read and output records written
- The count of padded records if /PD was selected
- The count of truncated records if /TR was selected

If you specify a file as an argument to /SL, and a file of the same name and type already exists, RMSCNV will append the summary to that file. You can thus use one file to keep a record of all conversions.

If you specify a file, and a file of the same name and type does not exist, RMSCNV will create a new file and write the summary to that file.

If you do not specify a file, the summary will be written to your terminal.

Do not specify either the input file or the output file as the argument to /SL.

By default, if you do not specify /SL, no summary listing will be produced. The messages written to your terminal (if RMSCNV encounters any errors) will be the only record of file conversion processing.

/SU

Directs RMSCNV to supersede an existing sequential output file. RMSCNV deletes the existing file, creates one of the same name, type, and version, and writes the records of the input file to it.

You cannot specify /SU and /AP in the same command string.

By default, if you do not specify /SU, RMSCNV will respond according to whether you have specified /AP:

- If you specified /AP, RMSCNV will append the input records to the output file.
- If you did not specify /AP and the output file exists, RMSCNV will create the next higher version of the file.

See also the description of the /AP switch.

/TR

Directs RMSCNV to truncate records read from the input file to the output file's maximum record size. The trailing bytes of the record are ignored.

You must use /TR when the input file's maximum record size is greater than the output file's.

In addition, when the output file contains fixed-length records and the input file contains variable-length records, you must specify both /TR and /PD. If both are not specified, RMSCNV will terminate with the fatal error message:

```
?CNV-F-TPM, Switch /TR or /PD or both are needed for this CONVERT
```

(See also the description of the /PD switch.)

By default, if you do not specify /TR and the input file's records are longer than the output file's records, RMSCNV will terminate with the fatal error message:

```
?CNV-F-MRS, Input and output record sizes do not correspond
```

/WF

Directs RMSCNV to handle variable-with-fixed-control (VFC) records in either the input or output file. RMSCNV processes the possible combinations of input and output file record formats, as follows:

Input File[1]	Output File[1]	Processing
VFC	VFC	If the fixed-control areas are the same size, RMSCNV will perform a straightforward copy operation. If the areas are not the same size, RMSCNV will terminate, without converting the file, with the fatal error message: ?CNV-F-VFC, Input and output fixed control header sizes do not correspond
VFC	FIX VAR STM	The fixed-control area of each input record is written as the first n bytes of each output record; n is the size of the fixed-control area. The variable portion of the input record completes the output record.
FIX VAR STM	VFC	The first n bytes of each input record are written into the fixed-control area of each output record; n is the size of the fixed-control area. The remaining data in the input record is written into the variable portion of the output record.

If you specify /WF and neither file contains VFC records, RMSCNV will terminate with the fatal error message:

?CNV-F-IWF, Illegal use of /WF with record format

By default, if you do not specify /WF and one of the files contains VFC records, RMSCNV will ignore the fixed-control area of each record, processing the records as follows:

Input File	Output File	Processing
VFC	any	Only the variable portion of each record is read from the input file and written to the output file.
any	VFC	Data read from the input file is written only into the variable portion of each output record.

4.4 RMSCNV PROCESSING

The manner in which RMSCNV reads and writes files depends on the organization of the files and on the switches you specify in the command string.

---

1. The expansions of the record format abbreviations used in the list are: variable with fixed control (VFC), fixed (FIX), variable (VAR), and stream (STM).

4.4.1 Output File Organization and RMSCNV Processing

4.4.1.1 Sequential Output File Organization - If the output file organization is specified as sequential (by default or by using the switch /FO:S), and the /AP switch is not specified, RMSCNV will create the output file with the attributes of the input file. RMSCNV will then read records from the input file and write them sequentially into the new output file.

If the output file already exists, RMSCNV will create the next higher version.

(For information on the /FO and /AP switches, see Section 4.3.)

If the output file organization is specified as sequential and the /SU switch is specified, RMSCNV will supersede an existing file with the records of the input file. If the output file does not exist, RMSCNV will create one with the attributes of the input file. (For more information on the /SU switch, see Section 4.3.)

If the output file organization is specified as sequential and the /AP switch is specified, RMSCNV will append records to an existing file by reading records from the input file and writing them sequentially into the output file, starting with the record position following the last record that is already in the output file. If the output file does not exist, RMSCNV will terminate with the fatal error message:

```
?CNV-F-ER$FNF, File <file-spec> not found
```

(For more information on the /AP switch, see Section 4.3.)

4.4.1.2 Relative Output File Organization - If the output file organization is specified as relative (using the switch /FO:R), RMSCNV will read records from the input file and write them into successive record cells of the output file, beginning with cell 1. If RMSCNV encounters a cell in the output file that already contains a record, it will terminate with the fatal error message:

```
?CNV-F-ER$REX, Relative record already exists for file <file-spec>
```

All records written up to that point are still in the file. You should examine the input and output files to determine the extent of processing.

If the relative output file does not exist, RMSCNV will terminate with the fatal error message:

```
?CNV-F-ER$FNF, File <file-spec> not found
```

4.4.1.3 Indexed Output File Organization - If the output file organization is specified as indexed (using the switch /FO:I), RMSCNV will read each record from the input file and apply the output file's record structure (that is, key placement within the record) to the data. This structure is an attribute of the output file and is not dependent on input file organization. The record is inserted into the output file on the basis of the value in the primary key field. Then, RMSCNV will update the output file's primary and (if any) alternate indexes to reflect the presence of the new record.

If the indexed output file does not exist, RMSCNV will terminate with the fatal error message:

```
?CNV-F-ER$FNF, File <file-spec> not found
```

4.4.2 Input File Organization and RMSCNV Processing

4.4.2.1 Sequential Input File Organization - If the input file organization is sequential, RMSCNV will read the records in sequential access mode, starting with the first record in the file.

4.4.2.2 Relative Input File Organization - If the input file organization is relative, RMSCNV will read the records in sequential access mode, starting with record cell 1.

4.4.2.3 Indexed Input File Organization - If the input file organization is indexed, RMSCNV will read the records in sequential access mode, following the key of reference specified by the /KR or /KN switch in the command string (the primary key is the default if none is specified).

If a record in an indexed input file is shorter than the primary key of an indexed output file, RMSCNV will terminate with the fatal error message:

```
?CNV-F-ER$RSZ, Illegal record size specified for file <file-spec>
```

Any records processed before the too-short record may or may not have been written to the output file, depending on such factors as buffer sizes and I/O techniques. You can use the /KR or /KN switch (see Section 4.3) to examine the input file at your terminal as follows:

```
TI:=INFILE.IDX/KR:1    or    TI:=INFILE.IDX/KN:NAME
```

4.5 RMSCNV EXAMPLES

```
>DES PAYROL.DES
>CNV PAYROL.DAT/FO:I/SL=PAYROL.SEQ
RMS-11 Convert -- Version 2.0 Summary Listing

Options in effect:

DB0:[70,5]PAYROL.SEQ;2
  File organization: Sequential
  File attributes:  Allocation=31, Extend=0
  Record format:    Variable length
  Record attributes: Carriage return

DB0:[70,5]PAYROL.DAT;2
  File organization: Indexed, using 2 key(s)
  File attributes:  Allocation=3, Extend=0, Bucket size = 1
  Record format:    Variable length
  Record attributes: Carriage return
```

```
Message summary:

Processing summary:
  Records read:      448
  Records written:   448
```

First, RMSDES is used to create a new indexed file, PAYROL.DAT. Then, RMSCNV reads the input records from the input sequential file PAYROL.SEQ and writes the records to the PAYROL.DAT file. A summary of the RMSCNV operations is printed on the terminal.

```
>CNV STAT.DAT/AP=STAT.INP
```

RMSCNV appends the records from the input file, STAT.INP, to the output (sequential) file, STAT.DAT.

```
>CNV EMPLY.DAT/FO:REL/TR/WF=EMPLY.INP
```

RMSCNV reads each record of the input file, EMPLY.INP, and writes the records to the output file, EMPLY.DAT, which is a relative file with VFC format records. As each record is written, the first bytes of the input record become the fixed-control area of the output record. Since the maximum record size of the output file is 80 bytes, and the input file record size is 132 bytes, trailing bytes are discarded.

```
>CNV ACCNT.DAT/FO:IDX/TR/PD:*=ACCNT.INP
```

RMSCNV reads each record of the input file and writes the record to the indexed output file. The input file has variable length records and the output file has fixed length records. If necessary, RMSCNV truncates input records or pads them with the character "*".

```
>CNV VHC.KR0/CA/SL=VHC.INX/KR:0
RMS-11 Convert -- Version 2.0 Summary Listing

Options in effect:
  Create copying attribute data (CA)
  Select key by number (KR)

DB0:[70,5]VHC.INX;7
  File organization: Indexed, using 2 key(s)
  File attributes:   Allocation=62, Extend=1, Bucket size = 4
  Record format:     Fixed length 80 byte records
  Record attributes: Carriage return

DB0:[70,5]VHC.KR0;1
  File organization: Indexed, using 2 key(s)
  File attributes:   Allocation=40, Extend=1, Bucket size = 4
  Record format:     Fixed length 80 byte records
  Record attributes: Carriage return

Message summary:
%CNV-I-IKY, Input key of reference is #0 (NPANXX)

Processing summary:
  Records read:      11
  Records written:   11
```

RMSCNV creates an output file, VHC.KR0, with the input file's attributes. Then, RMSCNV reads the records from the indexed input file, based on the primary key. RMSCNV lists a summary of its operations on the terminal.

```
>CNV VHC.KR1/CA/SL=VHC.INX/KR:1
RMS-11 Convert -- Version 2.0 Summary Listing

Options in effect:
  Create copying attribute data (CA)
  Select key by number (KR)

DB0:[70,5]VHC.INX;7
  File organization: Indexed, using 2 key(s)
  File attributes:   Allocation=62, Extend=1, Bucket size = 4
  Record format:     Fixed length 80 byte records
  Record attributes: Carriage return

DB0:[70,5]VHC.KR1;1
  File organization: Indexed, using 2 key(s)
  File attributes:   Allocation=40, Extend=1, Bucket size = 4
  Record format:     Fixed length 80 byte records
  Record attributes: Carriage return

Message summary:
%CNV-I-IKY, Input key of reference is #1 (PLACE)

Processing summary:
  Records read:      11
  Records written:   11
```

RMSCNV creates an output file, VHC.KR1, with the input file's attributes. Then, RMSCNV reads the records from the indexed input file, based on the first alternate key. RMSCNV lists a summary of its operations on the terminal.

```
>CNV TI:=VHC.KR0
0505801480          10060000201221          005415   070BERNARDSVLNJ060481 00
0507301348          00060000201222          002885   070LONGBRANCHNJ071578 00
0511101332          10060000201223          002885   070MANASQUAN NJ011582 00
0498201417          00060000201224          006882   070CLIFFSIDE NJ060581 00
0506901429          00060000201225          002885   070METUCHEN  NJ060481 00
0500701457          00060000201226          005613   070CALDWELL  NJ111881 00
0500701457          00060000201227          005613   070CALDWELL  NJ111881 00
0500701457          00060000201228          005613   070CALDWELL  NJ111881 00
0507301348          00060000201229          002885   070LONGBRANCHNJ071578 00
0508901466          10060000201231          005643   070SOMERVILLENJ100679 00
0504801441          00060000201232          005613   070WESTFIELD NJ111881 00
```

In this example, RMSCNV reads the contents of the input indexed file and writes the records to the current TI: assignment.

```
>CNV CUST.FIL/CA/SL=NODA"SMITH XUK9"::[70,5]CUST.DAT
RMS-11 Convert -- Version 2.0 Summary Listing

Options in effect:
  Create copying attribute data (CA)
```

```
NODA"SMITH XUK9"::DB0:[70,5]CUST.DAT;1
  File organization: Sequential
  File attributes:   Allocation=20, Extend=0
  Record format:     Variable length
  Record attributes: Carriage return

DB0:[70,5]CUST.FIL;1
  File organization: Sequential
  File attributes:   Allocation=20, Extend=0
  Record format:     Variable length
  Record attributes: Carriage return

Message summary:

Processing summary:
  Records read:      124
  Records written:   124
```

RMSCNV creates an output file, CUST.FIL, from the attributes of the output file. The output file, CUST.DAT, is on a remote node, NODA. Each record is read from the output file and written to the input file, and a summary of the operations is displayed on the terminal.

CHAPTER 5

RMS-11 FILE DISPLAY UTILITY: RMSDSP

The RMS-11 File Display Utility (RMSDSP) displays a concise description of any RMS-11 file, including container files. A container file is a magnetic tape file that contains RMS-11 files that were backed up using RMSBCK (see Chapter 6).

You can use RMSDSP to display a file's attributes so you can create a file like it using RMSDES, or to determine what kind of file your high-level language task created so you can match and/or enhance it using RMSDES. (See Chapter 2 for information on RMSDES.)

The basic display that RMSDSP produces for disk sequential, relative, and indexed files includes:

- A complete file specification
- Prologue version (for relative and indexed files)
- File size
- File owner
- File identification
- Creation and revision dates
- File protection
- File organization, including numbers of keys and areas for indexed files
- File attributes
- Record format
- Record attributes

The basic display that RMSDSP produces for magnetic tape files includes:

- A complete file specification
- File organization
- File attributes
- Record format
- Record attributes

The basic display that RMSDSP produces for RMSBCK back-up files (on disk or on magnetic tape) includes:

- A complete file specification
- File size
- Creation date

In addition, for indexed files and back-up files, you can request a more detailed display (see Section 5.3). For indexed files, RMSDSP also displays information on areas, primary and alternate keys, and allocation.

For back-up files, the detailed display includes file protection, file organization, record format, contiguity, and whether the version number is octal (for files created on RSX-11M/M-PLUS systems) or decimal (for files created on VAX/VMS systems).

Although RMSDSP does not display allocation information for magnetic tape files, you can display this information by using the Peripheral Interchange Program (PIP; see the RSX-11M/M-PLUS Utilities Manual).

The following sections describe how to invoke and terminate RMSDSP (Section 5.1) and the RMSDSP command string and switches (Sections 5.2 and 5.3). Section 5.4 provides examples illustrating the RMSDSP displays. Appendix C documents messages.

5.1 INVOKING AND TERMINATING RMSDSP

The RMSDSP utility can be either installed or uninstalled. If RMSDSP is installed, invoke it with the command:

```
DSP [command-string]
```

If you include a command string, RMSDSP will execute the command and return the system prompt.

If you do not include a command string, RMSDSP will prompt for one:

```
DSP>
```

If RMSDSP is uninstalled on RSX-11M/M-PLUS systems, invoke it with the command:

```
RUN $RMSDSP
```

RMSDSP will respond with the prompt:

```
DSP>
```

In response to the prompt, you can type a command string or you can type CTRL/Z to terminate the utility. When RMSDSP has executed a command string, it returns the DSP> prompt. Section 5.2 describes the RMSDSP command string.

Indirect Command Files. You can execute series of RMSDSP command strings by means of indirect command files. The command strings in an indirect command file are executed when an at sign (@) followed by the file's name is specified to the utility. The default file type of .CMD is assumed if none is provided.

Installed RMSDSP will accept indirect command files either from the system prompt or from the utility prompt. For example, to execute the command strings contained in the indirect command file DSPCOM.CMD, you could type either of the following commands:

```
DSP @DSPCOM
   or
DSP> @DSPCOM
```

Uninstalled RMSDSP will accept indirect command files only from the utility prompt. For example:

```
DSP> @DSPCOM
```

If you executed the indirect command file from the system prompt, the system prompt will be returned. If you executed the indirect command file from within RMSDSP, the DSP> prompt will be returned.

5.2 RMSDSP COMMAND STRING

Format

```
[outfile=]infile[/switch...][,infile[/switch...]...]
```

outfile

The specification of a file that RMSDSP creates to write the attributes or list the contents of the specified input file(s). Wild card characters are not permitted in any field of the specification. The default version is the highest existing version number plus 1.

By default, if you do not specify an output file, RMSDSP will display the information on your terminal.

infile

The file specification of any RMS-11 file whose attributes you want displayed or a container file whose contents you want listed. For container files, you must specify the /BP switch (see Section 5.3). Wild card characters (*) are permitted in any field of the specification.

RMSDSP requires ANSI-standard labels on magnetic tapes. You must mount a tape volume to inform your operating system's file processor that you are accessing an ANSI-standard tape.

See the RSX-11M/M-PLUS RMS-11 User's Guide for more information on RMS-11 magnetic tape handling.

If you do not mount your tape volume, RMSDSP will not accept wild card characters in the specification and will respond to an explicit file name with the tape device's attributes.

/switch

One of the switches listed in Table 5-1 and described in Section 5.3. All switches are global; that is, they can be specified anywhere in the command string.

In place of the command string you can type HELP or ? to display help text about RMSDSP, or /ID to display the current version of the utility.

Table 5-1: RMSDSP Switches

Switch	Description
/BP	List contents of back-up file(s) Default: Provide basic display only
/FU	Provide detailed display for indexed files or back-up files Default: Provide basic display only
/ID	Identify current version Default: No identification

5.3 RMSDSP SWITCHES

/BP

Directs RMSDSP to read the specified input files as RMSBCK back-up files (see the description of the RMSBCK utility, Chapter 6) and to display the names, sizes, and creation dates of the RMS-11 files backed up to tape or disk (see Example 5-6 in Section 5.4).

If you specify /BP with both back-up and non-back-up files, RMSDSP will provide displays for the back-up files only. It will ignore the non-back-up files.

By default, if you do not specify /BP, RMSDSP will provide the basic display for disk or magnetic tape files: file organization and attributes, and record format and attributes (see Example 5-5 in Section 5.4).

/FU

Directs RMSDSP to provide the detailed display for either an indexed file or a back-up file (when used with /BP).

If you specify /FU with an indexed file, the display will include information on the individual keys and areas, and on allocation (see Example 5-4 in Section 5.4).

If you specify /FU with /BP for a back-up file, the display will include information on the owner, protection, file organization, record format, whether the file is contiguous, and whether the version number is octal (for a file created on an RSX-11M/M-PLUS system) or decimal (for a file created on a VAX/VMS system). See also Example 5-7 in Section 5.4.

Knowing whether the version number is octal or decimal is useful in transferring RMS-11 files among operating systems using RMSRST (see Chapter 7). If the file is displayed with no version number, it was created and backed up on a RSTS/E system.

Under RSTS/E systems, the decimal project-programmer number (PPN) fields are the reverse of the octal User Identification Code (UIC) fields under VAX/VMS and RSX-11M/M-PLUS. Thus, in the owner's field of the display for files created under RSTS/E, the PPN fields will be reversed and shown in their octal equivalents.

By default, if you do not specify /FU, RMSDSP will provide the basic displays for indexed files (see Example 5-3 in Section 5.4) and container files (see Example 5-6 in Section 5.4).

/ID

Displays the current version of the RMSDSP utility. For example:

```
DSP>OUTFIL.DAT/ID=INFILE.DAT
%DSP-I-ID, Version 2.000
```

The last two digits of the version number (00 in this example) are the patch level of the utility. The /ID switch can also be specified by itself on the command line:

```
DSP>/ID
%DSP-I-ID, Version 2.000
```

5.4 RMSDSP DISPLAYS

Examples 5-1 through 5-8 illustrate the RMSDSP display formats.

NOTE

The File ID field of the displays for disk sequential, relative, and indexed files is in octal for RSX-11M/M-PLUS systems.

Example 5-1: RMSDSP Display Format for Disk Sequential Files

```
>DSP PAYROL.SEQ

DB0:[70,5]PAYROL.SEQ;2
                        Size:     31./31.          Created: 1-APR-1983 05:45
                        Owner:    [070,005]        Revised: 1-APR-1983 14:11 (3)
                        File ID: (11766,3,0)
  File protection:      System:RWED, Owner:RWED, Group:RWED, World:R
  File organization:    Sequential
  File attributes:      Allocation=31, Extend=0
  Record format:        Variable length, no maximum defined
  Record attributes:    Carriage return
```

Example 5-2: RMSDSP Display Format for Relative Files

```
>DSP EMPLY.DAT

SY:[070,005]EMPLY.DAT;1                          RMS Prologue Version=1
                        Size:    30./30.         Created: 2-APR-1983 01:43
                        Owner:   [070,005]       Revised: 2-APR-1983 01:45 (2)
                        File ID: (6343,55,0)
  File protection:      System:RWED, Owner:RWED, Group:RWED, World:R
  File organization:    Relative, no maximum record number defined
  File attributes:      Allocation=30, Extend=0, Bucket size=2
  Record format:        VFC, 2 byte header, maximum 80 bytes
  Record attributes:    Carriage return
```

Example 5-3: RMSDSP Display Format for Indexed Files

```
>DSP PAYROL.DAT

DB0:[70,5]PAYROL.DAT;2                           RMS Prologue Version=1
                        Size:    63./63.         Created: 2-APR-1983 01:44
                        Owner:   [070,005]       Revised: 2-APR-1983 01:47 (2)
                        File ID: (27743,10,0)
  File protection:      System:RWED, Owner:RWED, Group:RWED, World:R
  File organization:    Indexed, using 2 key(s) with 1 area(s)
  File attributes:      Allocation=63, Extend=0, Bucket size=1
  Record format:        Variable length, no maximum defined
  Record attributes:    Carriage return
```

Example 5-4: RMSDSP Display Format for Indexed Files (with /FU Specified)

```
>DSP PAYROL.DAT/FU

DB0:[70,5]PAYROL.DAT;2                           RMS Prologue Version=1
                        Size:    63./63.         Created: 2-APR-1983 01:44
                        Owner:   [070,005]       Revised: 2-APR-1983 01:47 (2)
                        File ID: (27743,10,0)
  File protection:      System:RWED, Owner:RWED, Group:RWED, World:R
  File organization:    Indexed, using 2 key(s) with 1 area(s)
  File attributes:      Allocation=63, Extend=0, Bucket size=1
  Record format:        Variable length, no maximum defined
  Record attributes:    Carriage return


  Area number 0:
        Bucket size=1
        Allocation remaining=1, Extend=0

  Primary key:
        Name=<not_specified>, Minimum record length=12
        Key information:
                Type=String, Length=12, Duplicates, No changes
                Position(s) and size(s):
                        Segment=0, position=0, length=12
        Area information:
                Index:  Area number=0
                        Lowest level area=0
                        Bucket size=1, Fill factor=512
```

Example 5-4 (Cont.): RMSDSP Display Format for Indexed Files (with /FU Specified)

```
            Data:   Area number=0
                    Bucket size=1, Fill factor=512
        Allocation information:
                Root virtual block number=4, Root level=1
                First data bucket virtual block number=5

  Alternate key number 1:
        Name=<not_specified>, Minimum record length=17
        Key information:
                Type=String, Length=17, Duplicates, Changeable
                Position(s) and size(s):
                        Segment=0, position=0, length=17
        Area information:
                Index:  Area number=0
                        Lowest level area=0
                        Bucket size=1, Fill factor=512
                Data:   Area number=0
                        Bucket size=1, Fill factor=512
        Allocation information:
                Root virtual block number=7, Root level=1
                First data bucket virtual block number=8
```

Example 5-5: RMSDSP Display Format for Magnetic Tape Files

```
>DSP MM0:*.*;*

MM0:PAYROL.CON;1
  File organization:    Sequential
  File attributes:      Magnetic tape, Block_size=2064
  Record format:        Undefined
  Record attributes:    None


MM0:MASTER.BCK;1
  File organization:    Sequential
  File attributes:      Magnetic tape, Block_size=2064
  Record format:        Undefined
  Record attributes:    None
```

Example 5-6: RMSDSP Display for Back-Up Files

```
  >DSP MM0:*.*;*/BP

  Contents of container file MM0:MASTER.BCK;1

    [070,005]PAYROL.DAT;2                63./63.       2-APR-1983 01:44
    [070,005]STAT.DAT;1                  20./20.       2-APR-1983 01:43
    [070,005]EMPLY.DAT;1                 30./30.       2-APR-1983 01:43
    [070,005]ACCNT.DAT;1                 30./30.       2-APR-1983 01:43

  Contents of container file MM0:PAYROL.CON;1

    [070,005]PAYROL.DAT;2                63./63.       2-APR-1983 01:44
    [070,005]PAYROL.LOG;1                294./294.     1-APR-1983 05:45
    [070,005]PAYROL.DES;2                4./4.         14-JAN-1983 08:36
    [070,005]PAYROL.SEQ;2                31./31.       1-APR-1983 05:45
    [070,005]PAYROL.MST;1                3./3.         14-JAN-1983 08:43
```

Example 5-7: RMSDSP Display Format for Back-Up Files (with /FU Specified)

```
>DSP MM1:*.*/BP/FU

Contents of container file: MM1:RSX11M.CON;1

  [100,101]HART.SEQ;1                    0./0.          15-JAN-1982 14:06
          [100,101]     [RWED,RWED,RWED,R]              15-JAN-1982 14:06
          Organization=Sequential, Record_format=Variable
          Not contiguous, Version_number=Octal
  [100,101]HOGAN.REL;1                   1./1.          15-JAN-1982 14:06
          [100,101]     [RWED,RWED,RWED,R]              15-JAN-1982 14:06
          Organization=Relative, Record_format=Fixed
          Not contiguous, Version_number=Octal
  [100,101]SIEMBO.IDX;1                  100./200.      15-JAN-1982 14:12
          [100,101]     [RWED,RWED,RWED,R]              15-JAN-1982 14:12
          Organization=Indexed, Record_format=Fixed
          Not contiguous, Version_number=Octal
  [100,101]RMSDSP.TSK;11                 162./162.      15-JAN-1982 14:12
          [100,101]     [RWED,RWED,RWED,R]              15-JAN-1982 14:12
          Organization=Sequential, Record_format=Fixed
          Contiguous, Version_number=Octal
  [100,101]DEMO.CMD;1                    1./5.          15-JAN-1982 14:28
          [100,101]     [RWED,RWED,RWED,R]              15-JAN-1982 14:28
          Organization=Sequential, Record_format=Variable
          Not contiguous, Version_number=Octal
```

Example 5-8: RMSDSP Display Format Using Wildcard Characters

```
>DSP PAYROL.*

SY:[070,005]PAYROL.DAT;2                          RMS Prologue Version=1
                        Size:    63./63.          Created: 2-APR-1983 01:44
                        Owner:   [070,005]        Revised: 2-APR-1983 01:47 (2)
                        File ID: (27743,10,0)
  File protection:      System:RWED, Owner:RWED, Group:RWED, World:R
  File organization:    Indexed, using 2 key(s) with 1 area(s)
  File attributes:      Allocation=63, Extend=0, Bucket size=1
  Record format:        Variable length, no maximum defined
  Record attributes:    Carriage return


SY:[070,005]PAYROL.DES;2
                        Size:    4./4.            Created: 14-JAN-1983 08:36
                        Owner:   [070,005]        Revised: 2-APR-1983 01:44 (7)
                        File ID: (11342,13,0)
  File protection:      System:RWED, Owner:RWED, Group:RWED, World:R
  File organization:    Sequential
  File attributes:      Allocation=4, Extend=0
  Record format:        Variable length, no maximum defined
  Record attributes:    Carriage return


SY:[070,005]PAYROL.SEQ;2
                        Size:    31./31.          Created: 1-APR-1983 05:45
                        Owner:   [070,005]        Revised: 1-APR-1983 14:11 (3)
                        File ID: (11766,3,0)
  File protection:      System:RWED, Owner:RWED, Group:RWED, World:R
  File organization:    Sequential
  File attributes:      Allocation=31, Extend=0
  Record format:        Variable length, no maximum defined
  Record attributes:    Carriage return
```

CHAPTER 6

THE RMS-11 FILE BACK-UP UTILITY: RMSBCK

The RMS-11 File Back-Up Utility (RMSBCK) transfers the contents of any RMS-11 disk file to another disk or to a container file. A container file is a magnetic tape file that contains RMS-11 files that were backed up using RMSBCK.

When RMSBCK transfers RMS-11 files, the back-up copy retains the source file's attributes: account number, creation and revision dates, protection code, contiguity, and so on, with the exception of placement. A file cannot be restored to the same physical place on the device from which it was backed up.

The back-up files are specially formatted so that they can be accessed only by the RMSRST utility (see Chapter 7); they cannot be read or modified by user programs or other utility programs. RMSDSP, however, can display information about RMSBCK back-up files (see Chapter 5).

You can use RMSBCK and RMSRST to protect your data against hardware or software failure. When you back up files on a regular basis, files lost or damaged can be restored from the back-up copies.

You can also use RMSBCK and RMSRST to transfer files between different operating systems, for example, between RSX-11M/M-PLUS and VAX/VMS, and between RSTS/E and RSX-11M/M-PLUS or VAX/VMS. See Chapter 7 for more information on using RMSRST to restore files to different operating systems.

RMSBCK backs up files to magnetic tape volumes with ANSI-standard labels only. However, the back-up data written to the volume by RMSBCK need not comply with ANSI standards.

Note that RMSBCK cannot back up a system disk as a bootable volume.

RMSBCK provides four processing features:

- Explicit and implicit input file selection -- You can specify one or more input files with or without wild card characters.
- Data integrity checks -- You can check the reliability of the backed up data files.
- Extended diagnostic messages -- Query mode gives you the option of continuing processing if RMSBCK encounters an error condition.
- Summary listing -- You can create a record of RMSBCK processing.

Section 6.4 describes these processing features in greater detail.

The following sections describe how to invoke and terminate RMSBCK (Section 6.1) and the RMSBCK command string and switches (Sections 6.2 and 6.3). Section 6.5 provides examples illustrating the use of RMSBCK. Appendix C documents messages.

6.1 INVOKING AND TERMINATING RMSBCK

The RMSBCK utility can be either installed or uninstalled. If RMSBCK is installed, invoke it with the command:

```
BCK [command-string]
```

If you include a command string, RMSBCK will execute the command and return the system prompt.

If you do not include a command string, RMSBCK will prompt for one:

```
BCK>
```

If RMSBCK is uninstalled on RSX-11M/M-PLUS systems, invoke it with the MCR command:

```
RUN $RMSBCK
```

RMSBCK will respond with the prompt:

```
BCK>
```

In response to the prompt, you can type a command string or you can type CTRL/Z to terminate the utility. When RMSBCK has executed a command string, it returns the BCK> prompt. Section 6.2 describes the RMSBCK command string.

Indirect Command Files. You can execute series of RMSBCK command strings by means of indirect command files. The command strings in an indirect command file are executed when an at sign (@) followed by the file's name is specified to the utility. The default file type of .CMD is assumed if none is provided.

Installed RMSBCK will accept indirect command files either from the system prompt or from the utility prompt. For example, to execute the command strings contained in the indirect command file BCKCOM.CMD, you could type either of the following commands:

```
BCK @BCKCOM
   or
BCK> @BCKCOM
```

Uninstalled RMSBCK will accept indirect command files only from the utility prompt. For example:

```
BCK> @BCKCOM
```

If you executed the indirect command file from the system prompt, the system prompt will be returned. If you executed the indirect command file from within RMSBCK, the BCK> prompt will be returned.

6.2 RMSBCK COMMAND STRING

Before you issue a command string to RMSBCK, the back-up medium (disk or ANSI-labeled magnetic tape) must be online and mounted.

Format

```
outfile[/switch...]=infile[/switch...][,infile[/switch...]...]
```

outfile

The specification of a disk or magnetic tape file to be created by RMSBCK. If the back-up medium is disk, both the file name and type must be wild card characters (*.*). The file names and types of the container files will be the same as the associated input data files. You can omit the version; RMSBCK will use the version of the associated input file.

If you do not use wild card characters in specifying the output file, RMSBCK will terminate with the error message:

```
%BCK-W-NRN, No rename allowed
```

If the back-up medium is magnetic tape, multiple input files will be backed up in a single container file. You must explicitly name the output container file in the command string, but the name need not correspond to the name of any input file. You can omit the version; a value of 0 will be used. If you do not explicitly name the container file, RMSBCK will terminate with the fatal error message:

```
%BCK-W-NDM, Explicit container filename necessary
```

RMSBCK does not prevent you from putting multiple container files with the same file specification on the same magnetic tape volume. Therefore, you should specify unique file names, types, and versions for output container files written to the same volume. (You can use RMSDSP to display information about back-up files; see Chapter 5.)

infile

The specification of a disk file from which a back-up copy is to be created. You can use wild card characters (*) to implicitly specify a collection of files.

If you specify a nondisk device, RMSBCK will terminate with the fatal error message:

```
?BCK-F-ER$DEV, Bad or inappropriate device type for file <device>
```

<device> is the device name and unit you specified.

If you specify a file that is listed in several accounts by means, for example, of the PIP/ENTER command (see the RSX-11M/M-PLUS Utilities Manual), RMSBCK will copy the entire file even though it exists in another account. However, when you restore the file with RMSRST, the link to the other accounts will be broken.

For example, you have a file of names, ACCTSRECV.DAT, stored in account [30,1] which is pointed to by the directory entry [10,1]CLIENTS.DAT (by means of the PIP/ENTER command). If you back up [10,1]CLIENTS.DAT, RMSBCK will copy the file from [30,1]ACCTSRECV.DAT. When you restore the file, RMSRST will write the file into account [10,1]. The restored file will no longer be linked with the file of names in [30,1]ACCTSRECV.DAT; rather, it will be a duplicate of that file.

/switch

One of the switches listed in Table 6-1 and described in Section 6.3.

In place of the command string, you can type HELP or ? to display help text about RMSBCK, or /ID to display the current version of the utility.

Table 6-1: RMSBCK Switches

Type	Switch	Description
Global	/ID	Identify current version
		Default: No identification
	/[NO]QU	Enable or disable query mode
		Default: Enable query mode
	/SL[:file-spec]	Provide summary listing (in file, if specified)
		Default: No summary
Output File	/RA	Read-after-writing data integrity checking
		Default: No reading
	/RC	Check-after-writing data integrity checking
		Default: No checking
	/RW	Rewind magnetic tape before writing
		Default: No rewinding
	/SU	Supersede existing file
		Default: No supersession
Input File	/CD:dd-mmm-yy[:v]	Back up files based on creation date
		Default: No date checking
	/RD:dd-mmm-yy[:v]	Back up files based on revision date
		Default: No date checking

6.3 RMSBCK SWITCHES

This section describes the global, output file, and input file switches that you can specify in RMSBCK command strings. Global switches are switches that can be specified anywhere in the command string.

6.3.1 Global Switches

The global switches are /ID, /QU, and /SL.

/ID

Displays the current version of the RMSBCK utility. For example:

```
BCK>OUTFIL.DAT/ID=INFILE.DAT
%BCK-I-ID, Version 2.000
```

The last two digits of the version (00 in this example) are the patch level of the utility. The /ID switch can also be specified by itself on the command line:

```
BCK>/ID
%BCK-I-ID, Version 2.000
```

/[NO]QU

Enables or disables query mode. If you specify /QU, RMSBCK will allow you to chose to continue or to terminate processing when one of the following occurs:

- A read error occurs on an input file.
- A read-after-write error occurs on an output file when the /RA switch is specified (see Section 6.3.2).
- A read-after-write error occurs on an output file when the /RC switch is specified (see Section 6.3.2).
- The table allocated internally for data integrity checks or for automatic retry of read errors is full.

When one of these errors occurs, RMSBCK prints a diagnostic message specifying the type of error and the name of the file being processed. It also prompts you, asking whether you want to continue processing. If you answer Y, RMSBCK will continue processing the current file and command string. If you answer N, RMSBCK will terminate processing immediately, without processing the rest of the command string.

For example:

```
BCK>OUTFIL.DAT/QU=INFILE.DAT
?BCK-F-WCE, Check after write error on output file <file-spec>
            VBN 1 to 2.  Continue (Yes or No)? Y
?BCK-F-TER, Integrity check table full.  Continue (Yes or No)? N
```

You can disable query mode with the /NOQU switch. RMSBCK will print a diagnostic message when it encounters an error and terminate processing.

By default, if you do not specify /QU or /NOQU, query mode will be enabled.

/SL[:file-spec]

Directs RMSBCK to produce a summary of file back-up processing. The summary listing consists of:

- The command string
- The names of files successfully backed up
- The names of files that were not backed up, with associated error messages
- The diagnostic messages produced in query mode (if query mode was enabled)
- A summary of any input errors and a summary of any errors remaining in output back-up files after automatic retry of data integrity checks (if you specified /RA or /RC)

If you specify a file as an argument to /SL, and a file of the same name and type already exists, RMSBCK will append the summary to that file. You can thus use one file to keep a record of all file conversions.

If you specify a file, and a file of the same name and type does not exist, RMSBCK will create a new file and write the summary to that file.

If you do not specify a file, the summary will be written to your terminal.

Do not specify either the input file or the output file as the argument to /SL.

By default, if you do not specify /SL, RMSBCK will write messages to your terminal if it encounters errors (see Appendix C).

6.3.2 Output File Switches

The output file switches are /RA, /RC, /RW, and /SU.

/RA

Directs RMSBCK to read the back-up file after writing it. RMSBCK reads back each block of the back-up file. If the device hardware detects a read error, RMSBCK will respond according to whether query mode is enabled (see the description of the /[NO]QU switch in Section 6.3.1).

Note that you cannot specify /RA and /RC in the same command string. RMSBCK terminates with the fatal error message:

```
?BCK-F-CSW, Conflicting switch specified - /sw
```

The value /sw is /RA or /RC depending on which appears second in the command string.

By default, if you do not specify /RA, RMSBCK will not perform read-after-writing data integrity checking.

/RC

Directs RMSBCK to check the back-up file after writing it. After the file has been completely backed up, RMSBCK reads it back into memory, block by block. At the same time, it reads the input file back into memory, block by block. RMSBCK compares the corresponding blocks as it reads them. If an error occurs during the comparison (a hardware read error or a mismatch between the contents of the two blocks), RMSBCK will respond according to whether query mode is enabled (see the description of the /[NO]QU switch in Section 6.3.1).

Note that you cannot specify /RC and /RA in the same command string. RMSBCK terminates with the fatal error message:

?BCK-F-CSW, Conflicting switch specified - /sw

The value /sw is /RA or /RC depending on which appears second in the command string.

By default, if you do not specify /RC, RMSBCK will not perform check-after-writing data integrity checking.

/RW

Directs RMSBCK to rewind a magnetic tape before writing a container file to it. Rewinding logically deletes any existing files on the tape.

This switch can be specified only if the back-up medium is a magnetic tape. If the back-up medium is a disk and you specify /RW, RMSBCK will terminate with the fatal error message:

?BCK-F-LSW, Illogical switch in context of command - /RW

By default, if you do not specify /RW, RMSBCK will add a new container file to the logical end of the magnetic tape.

/SU

Directs RMSBCK to supersede an existing disk output file with records of the input file. RMSBCK does not create a new version of the output file: it deletes the existing file, creates one of the same name, type, and version, and writes the records of the input file to it.

This switch can be specified only if the back-up medium is a disk. If the back-up medium is a magnetic tape and you specify /SU, RMSBCK will terminate with the fatal error message:

?BCK-F-LSW, Illogical switch in context of command - /SU

By default, if you do not specify /SU, RMSBCK will not supersede the files on the output device. If RMSBCK encounters a file with the same file name, type, and version as an input file, it will print the following error message and continue processing:

?BCK-F-ER$FEX, File <file-spec> already exists

6.3.3 Input File Switches

The input file switches are /CD and /RD.

/CD:dd-mmm-yy[:v]

Directs RMSBCK to back up files based on their creation date. If you use wild card characters in the input file specification, you can identify a group of files, depending on the value of the argument**v**. If you do not specify **v**, RMSBCK will back up all files that satisfy the wild card specification and were created on the date specified. If you specify the value A for the argument **v**, RMSBCK will back up all files that satisfy the wild card specification and were created after the date specified. If you specify the value B for **v**, RMSBCK will back up all files that satisfy the wild card specification and were created before the date specified.

If a file has no or a null creation date, RMSBCK will treat it as though it had been created before January 1900.

By default, if you do not specify /CD, RMSBCK will back up all files that satisfy the input file specification (unless you specified /RD; see the description of the /RD switch in this section).

/RD:dd-mmm-yy[:v]

Directs RMSBCK to back up files based on their revision dates, that is, the dates they were last accessed for writing. If you use wild card characters in the input file specification, you can identify a group of files, depending on the value of the argument **v**. If you do not specify **v**, RMSBCK will back up all files that satisfy the wild card specification and were revised on the date specified. If you specify the value A for the argument **v**, RMSBCK will back up all files that satisfy the wild card specification and were revised after the date specified. If you specify the value B for **v**, RMSBCK will back up all files that satisfy the wild card specification and were revised before the date specified.

If a file has no or a null revision date, RMSBCK will treat it as though it had been revised before January 1900.

By default, if you do not specify /RD, RMSBCK will back up all files that satisfy the input file specification (unless you specified /CD; see the description of the /CD switch in this section).

6.4 RMSBCK PROCESSING

RMSBCK copies standard RMS-11 data files from one medium to another (disk to disk or disk to magnetic tape) so that, if the original data files are lost or damaged, you have the back-up copies from which you can restore the original data.

You must use ANSI-labeled magnetic tapes if the back-up medium is magnetic tape. However, the back-up data written to the tape need not comply with ANSI standards. RMSBCK translates the data into a special format; back-up files can be accessed properly only by RMSRST (see Chapter 7). User and other utility programs, therefore, cannot read or modify back-up data.

When the back-up medium is magnetic tape, RMSBCK uses 2064-character blocks (representing 16 bytes of container file overhead and 4 disk blocks as a record to the container file). You cannot alter the block size used by RMSBCK.

You can use multivolume magnetic tapes; this capability is implemented by means of the magnetic tape ancillary control process (ACP) interface.

NOTE

For multivolume backup sets, the /RA and /RC switches will not function properly since tape positioning cannot be accomplished on those files which span tape volumes.

RMSBCK handles unused blocks in a file according to file organization:

- If the file is sequential or relative, RMSBCK will not copy the unused blocks.
- If the file is indexed, RMSBCK will copy the unused blocks.

RMSBCK provides four processing features, as described in the following sections.

6.4.1 Explicit and Implicit Input File Specification

The RMSBCK command string allows you to specify input files either explicitly or implicitly. File specifications with wild card characters (*) implicitly identify a collection of files. When you use wild card characters, you can restrict the number of files selected for processing by including one of the date switches, /CD or /RD (see Section 6.3.3). A date switch directs RMSBCK to examine either the creation or the revision date of each file selected as a result of the wild card specification: RMSBCK will back up a file only if its creation or revision date matches the date specified with the switch.

6.4.2 Data Integrity Checking

You can direct RMSBCK to check data integrity as each file is created. If you specify the /RA output file switch, RMSBCK will read back the file contents after they are written to the back-up file. If you specify the /RC output file switch, RMSBCK will read back, block by block, the contents of the output (back-up) file and the contents of the original file and compare them. RMSBCK automatically retries read errors if processing continues to normal termination.

Thus, you have three choices in data integrity checking:

- You can specify neither /RA nor /RC and rely on software and hardware accuracy, backing up your files in minimum time.
- You can specify /RA and verify that the files can be read from the back-up medium, adding reading time to the minimum time.
- You can specify /RC and verify that the back-up files can be read and that they exactly match the original input files, adding reading and comparison times to the minimum time.

See Section 6.3.2 for descriptions of the /RA and /RC switches.

6.4.3 Extended Diagnostic Messages

Query mode allows you to specify that processing continues or terminates when RMSBCK encounters an error condition. By default, query mode is enabled. When RMSBCK encounters an error, it issues a diagnostic message to your terminal and prompts you:

```
Continue (Yes or No)?
```

Depending on your response, RMSBCK continues or terminates processing. With this feature, you can ensure that RMSBCK does not terminate processing of a group of files if errors are encountered in only one file.

You can disable query mode with the /NOQU output file switch. If RMSBCK encounters an error, it will issue a diagnostic message to your terminal and terminate processing automatically.

See Section 6.3.1 for a description of the /[NO]QU switch.

6.4.4 Summary Listing

You can use the /SL switch to produce a summary listing of RMSBCK processing, including data integrity checks and error messages. RMSBCK will either write the summary to your terminal or into a file that you specify, creating a permanent record of the back-up processing.

See the description of the /SL switch in Section 6.3.1, and the example summary listing in Section 6.5.

6.5 RMSBCK EXAMPLES

```
>MOU MM0:PAYROL
>BCK MM0:PAYROL.CON=PAYROL.*/QU/SL
%BCK-I-FPC, Processing complete for file SY:[070,005]PAYROL.DAT;2
%BCK-I-FPC, Processing complete for file SY:[070,005]PAYROL.LOG;1
%BCK-I-FPC, Processing complete for file SY:[070,005]PAYROL.DES;2
%BCK-I-FPC, Processing complete for file SY:[070,005]PAYROL.SEQ;2
%BCK-I-FPC, Processing complete for file SY:[070,005]PAYROL.MST;1
```

RMSBCK backs up, on ANSI-labeled magnetic tape, all PAYROL files in the default account on SY:, into a container file named PAYROL.CON.

Query mode is explicitly enabled; therefore, any error condition causes RMSBCK to issue a diagnostic message and query the user as to whether processing should continue. While RMSBCK processes the input files, it writes a summary to the terminal.

```
>BCK MM0:MASTER.BCK=*.DAT/CD:30-MAR-83:A/SL:BACKUP.LST
```

RMSBCK backs up, on ANSI-labeled magnetic tape, all files in the default account on SY: that were created after March 30, 1983 and have a file type of .DAT.

Query mode is enabled by default; therefore, any error condition causes RMSBCK to issue a diagnostic message and query the user as to whether processing should continue. RMSBCK also creates a summary listing file named BACKUP.LST in the default account on SY:. The file contains a listing of the files processed and the diagnostic messages and queries (and responses) that occurred during processing.

```
>BCK DB0:[70,1]*.*;*/SU=DB0:[70,10]*.*;*/RD:01-APR-83:B/NOQU
```

RMSBCK backs up all files in account [70,10] on DB0: that were revised before April 1, 1983. The back-up copies are written into account [70,1] on the same disk (DB0:), superseding files with the same file names and types.

Query mode is disabled; therefore, if RMSBCK encounters an error, it will issue a diagnostic message and terminate processing.

Note: Examples 5-5 and 5-6 in the RMSDSP chapter illustrate the contents of the tape, after the completion of the above backups. Examples showing restores of these files are in the RMSRST chapter.

CHAPTER 7

RMS-11 FILE RESTORATION UTILITY: RMSRST

The RMS-11 File Restoration Utility (RMSRST) transfers, to disk, files that were backed up on magnetic tape or disk using RMSBCK. The form, structure, contents, and attributes (except file placement) of the restored files are those of the original files when they were backed up.

File placement will not be restored. A file cannot be restored to the same physical place on the device from which it was backed up.

You can use RMSBCK and RMSRST to protect your data against hardware or software failure: files lost or damaged can be restored from back-up copies.

You can also use RMSBCK and RMSRST to transfer files between different operating systems, for example, between RSX-11M/M-PLUS and VAX/VMS, and between RSTS/E and RSX-11M/M-PLUS or VAX/VMS (see Section 7.4.3).

Files backed up using RMSBCK can be accessed properly only by RMSRST. RMSBCK translates data into a special format that cannot be read or modified by user programs or other utility programs. RMSRST translates the specially formatted data on the back-up files back into standard RMS-11 files. See Chapter 6 for information on RMSBCK.

RMSRST cannot restore a system disk as a bootable medium.

RMSRST provides six processing features:

- Explicit and implicit input file selection -- You can specify one or more input files with or without wild card characters.
- Choice of restoration account and/or volume -- You can restore files into any account on any disk device to which you have access privileges.
- File transfer to different operating systems -- You can restore files created and backed up on one operating system to another operating system.
- Data integrity checks -- You can check the reliability of the restored data files.
- Extended diagnostic messages -- Query mode gives you the option of continuing processing if RMSRST encounters an error condition.
- Summary listing -- You can create a record of RMSRST processing.

Section 7.4 describes these processing features in greater detail.

The following sections describe how to invoke and terminate RMSRST (Section 7.1) and the RMSRST command string and switches (Sections 7.2 and 7.3). Section 7.5 provides examples illustrating the use of RMSRST. Appendix C documents messages.

7.1 INVOKING AND TERMINATING RMSRST

The RMSRST utility can be either installed or uninstalled. If RMSRST is installed, invoke it with the command:

```
RST [command-string]
```

If you include a command string, RMSRST will execute the command and return the system prompt.

If you do not include a command string, RMSRST will prompt for one:

```
RST>
```

If RMSRST is uninstalled on RSX-11M/M-PLUS systems, invoke it with the MCR command:

```
RUN $RMSRST
```

RMSRST will respond with the prompt:

```
RST>
```

In response to the prompt, you can type a command string or you can type CTRL/Z to terminate the utility. When RMSRST has executed a command string, it returns the RST> prompt. Section 7.2 describes the RMSRST command string.

Indirect Command Files. You can execute series of RMSRST command strings by means of indirect command files. The command strings in an indirect command file are executed when an at sign (@) followed by the file's name is specified to the utility. The default file type of .CMD is assumed if none is provided.

Installed RMSRST will accept indirect command files either from the system prompt or from the utility prompt. For example, to execute the command strings contained in the indirect command file RSTCOM.CMD, you could type either of the following commands:

```
RST @RSTCOM
   or
RST> @RSTCOM
```

Uninstalled RMSRST will accept indirect command files only from the utility prompt. For example:

```
RST> @RSTCOM
```

If you executed the indirect command file from the system prompt, the system prompt will be returned. If you executed the indirect command file from within RMSRST, the RST> prompt will be returned.

7.2 RMSRST COMMAND STRING

Format

```
outfile[/switch...]=infile[/switch...][,infile[/switch...]...]
```

outfile

The specification of a disk file to which back-up files are to be restored by RMSRST. The file name and type must be wild card characters (*.*); you cannot rename files as they are restored by RMSRST. You can omit the version or use a wild card character; RMSRST will use the version of the original file.

If you omit the account field of the file specification, RMSRST will restore the files to your default account. If you specify [*,*], RMSRST will restore the files to the original accounts from which they were backed up. If you explicitly specify an account, RMSRST will restore the files to that account.

infile

The specification of a back-up file on disk or a container file. A container file is a magnetic tape file that contains RMS-11 files that were backed up using RMSBCK. You can use wild card characters (*) to specify multiple container files.

If you omit the version and the input file is a container file, RMSRST applies the following defaults:

- If wild card characters appear in either or both the container file name and type, RMSRST will restore all versions of the container file that satisfy the wild card specification.
- If both the container file name and the type are explicitly specified, RMSRST will restore the first container file that it encounters that satisfies the specification and has a version of 0.

To restore a specific data file from magnetic tape, you must specify the container file name and specify the data file name with the /SE input file switch (see Section 7.3.3). (You can use RMSDSP to display information about the files in a container file; see Chapter 5.)

/switch

One of the switches listed in Table 7-1 and described in Section 7.3.

In place of the command string, you can type HELP or ? to display help text about RMSRST, or /ID to display the current version of the utility.

Table 7-1: RMSRST Switches

Type	Switch	Description
Global	/ID	Identify current version Default: No identification
	/[NO]CV	Enable or disable file version number conversion Default: Conversion enabled, if appropriate
	/[NO]QU	Enable or disable query mode Default: Query mode enabled
	/SL[:file-spec]	Provide summary listing (in file, if specified) Default: No summary
Output File	/FR	Change protection code Default: Original protection
	/RA	Read-after-writing data integrity checking Default: No reading
	/RC	Check-after-writing data integrity checking Default: No checking
	/SU	Supersede existing files Default: No supersession
Input File	/BD:dd-mmm-yy	Restore disk files based on back-up date Default: No date checking
	/OA:[act nbr][1]	Restore files based on original account Default: No account criterion applied
	/SE:file-spec or /SE:(file-spec1,file-spec2 [,...,file-spec10])	Restore specified files from container file Default: Restore all files on container file

1. Square brackets are required with the **act nbr** argument.

7.3 RMSRST SWITCHES

This section describes the global, output file, and input file switches that you can specify in the RMSRST command string. Global switches are switches that can be specified anywhere in the command string.

7.3.1 Global Switches

The global switches are /CV, /ID, /QU, and /SL.

/[NO]CV

Enables or disables file-version number conversion: from decimal to octal for VAX/VMS files being restored to RSX-11M/M-PLUS systems.

You can use RMSDSP to display information about files backed up using RMSBCK, including the radix of the version number; see Chapter 5. For more information about file transfer to different operating systems, see Section 7.4.3.

For back-up files created using RMSBCK Version 1.8 or earlier, the default is /NOCV. You must specify /CV explicitly to force the conversion; RMSRST will assume that the file is from a "foreign" system (from RSX-11M/M-PLUS if RMSRST is being run on a VAX/VMS system, or from VAX/VMS if RMSRST is being run on an RSX-11M/M-PLUS system) and convert the file version number to the correct radix for the system on which the files are being restored.

If the file was originally created and backed up on a RSTS/E system, it will have no version number, and the /CV switch will have no effect.

If the back-up file is not from a foreign system and you specify /CV, the file version numbers may be converted to incorrect values.

For back-up files created using RMSBCK Version 2.0, the default is /CV. The file attributes preserved by RMSBCK include the radix of the version number. Version numbers will be converted automatically, if necessary, to the correct radix for the system on which the files are being restored. You can override the conversion with /NOCV.

/ID

Displays the current version of the RMSRST utility. For example:

```
RST>OUTFIL.DAT/ID=INFILE.DAT
%RST-I-ID, Version 2.000
```

The last two digits of the version number (00 in this example) are the patch level of the utility. The /ID switch can also be specified by itself on the command line:

```
RST>/ID
%RST-I-ID, Version 2.000
```

/[NO]QU

Enables or disables query mode. If you specify /QU, RMSRST will allow you to chose to continue or to terminate processing when one of the following occurs:

- A read error occurs on an input file.
- A read-after-write error occurs on an output file when the /RA switch is specified (see Section 7.3.2).
- A read-after-write error occurs on an output file when the /RC switch is specified (see Section 7.3.2).
- The table allocated internally for data integrity checks or for automatic retry of read errors is full

When one of these errors occurs, RMSRST prints a diagnostic message specifying the type of error and the name of the file being processed. It also prompts you, asking whether you want to continue processing. If you answer Y, RMSRST will continue processing the current file and command string. If you answer N, RMSRST will terminate processing immediately, without processing the rest of the command string.

For example:

```
RST>OUTFIL.DAT/QU=INFILE.DAT
?RST-F-WCE, Check after write error on output file OUTFIL.DAT
            VBN 1 to VB2.  Continue (Yes or No)?  Y
?RST-F-TER, Integrity check table full.  Continue (Yes or No)?  N
```

You can disable query mode with the /NOQU switch. RMSRST will print a diagnostic message when it encounters an error and terminate processing.

By default, if you do not specify /QU or /NOQU, query mode will be enabled.

/SL[:file-spec]

Directs RMSRST to produce a summary of file restoration processing. The summary listing consists of the following:

- The command string
- The names of files successfully restored
- The names of files that were not restored, with associated error messages
- The diagnostic messages produced in query mode (if query mode was enabled)
- A summary of any input errors and a summary of any errors remaining in restored output files after automatic retry of data integrity checks (if you specified /RA or /RC)

If you specify a file as an argument to /SL, and a file of the same name and type already exists, RMSRST will append the summary to that file. You can thus use one file to keep a record of all file conversions.

If you specify a file, and a file of the same name and type does not exist, RMSRST will create a new file and write the summary to that file.

If you do not specify a file, the summary will be written to your terminal.

Do not specify either the input file or the output file as the argument to /SL.

By default, if you do not specify /SL, RMSRST will write messages to your terminal if it encounters errors (see Appendix C).

7.3.2 Output File Switches

The output file switches are /FR, /RA, /RC, and /SU.

/FR

Directs RMSRST to change the protection code of a file when it restores it to disk. If the file is restored to the operating system from which it was backed up, RMSRST changes the protection code so that the default account (the account under which RMSRST is running) becomes the owner of the file.

If the file being restored was created and backed up on another operating system, RMSRST will set the protection code to the system default. For more information on intersystem file transfer, see Section 7.4.3.

If the file is restored to an account other than the one from which it was backed up and /FR is not specified, RMSRST will interpret the account field as described in the definition of the outfile specification in Section 7.1. In this case, the protection code will not be changed: the original owner will remain the owner, and so on.

By default, if you do not specify /FR, RMSRST will not change the protection code from the value copied when the file was backed up.

/RA

Directs RMSRST to read the restored file after writing it. RMSRST reads back each block of the file after it has been completely restored. If the device hardware detects a read error, RSMRST will respond according to whether query mode is enabled (see the description of the /[NO]QU switch in Section 7.3.1).

Note that you cannot specify /RA and /RC in the same command string. RMSRST terminates with the fatal error message:

```
?RST-F-CSW, Conflicting switch specified - /sw
```

The value /sw is /RA or /RC depending on which appears second in the command string.

By default, if you do not specify /RA, RMSRST will not perform read-after-writing data integrity checking.

NOTE

For multivolume backup sets, the /RA and /RC switches will not function properly since tape positioning cannot be accomplished on those files which span tape volumes.

/RC

Directs RMSRST to check the restored file after writing it. After the file has been completely restored, RMSRST reads it back into memory block by block. At the same time it reads the input (back-up) file back into memory block by block. RMSRST compares corresponding blocks of the files as it reads them. If an error occurs during the comparison (a hardware read error or a mismatch between the contents of the blocks), RMSRST will respond according to whether query mode is enabled (see the description of the /[NO]QU switch in Section 7.3.1).

Note that you cannot specify /RC and /RA in the same command string. RMSRST terminates with the fatal error message:

```
?RST-F-CSW, Conflicting switch specified - /sw
```

The value /sw is /RA or /RC depending on which appears second in the command line.

By default, if you do not specify /RC, RMSRST will not perform check-after-writing data integrity checking.

/SU

Directs RMSRST to supersede an existing output file with the records of the input file. RMSRST deletes the existing file, creates one of the same name, type, and version, and writes the records of the input file to it.

By default, if you do not specify /SU and an output file already exists with the same name, type, and version, RMSRST will print the following error message and continue processing (RMSRST does not restore duplicate files):

```
?RST-F-ER$FEX, File <file-spec> already exists
```

7.3.3 Input File Switches

The input file switches are /BD, /OA, and /SE.

/BD:dd-mmm-yy

Directs RMSRST to restore disk files based on their back-up date. If you use wild card characters in the input file specification and specify this switch, you can restore specific files within a group of disk files. RMSRST will restore only the files that satisfy the wild card specification and were backed up on the specified date.

Note that you cannot use the /BD switch if the input files are stored on magnetic tape; RMSRST cannot access the creation date of a file on magnetic tape. If the input file is on magnetic tape and you specify /BD, RMSRST will terminate with the fatal error message:

```
?RST-F-LSW, Illogical switch in context of command - /BD
```

By default, if you do not specify /BD, RMSRST will apply no date criterion in restoring files: all files that satisfy the input file specification will be restored.

/OA:[act nbr]

Directs RMSRST to restore files based on the account from which the original files were backed up. The **[act nbr]** argument is the User Identification Code (UIC). The square brackets are required with the UIC.

If you use wild card characters in the input file specification and specify this switch, you can restore specific files within a group of disk or magnetic tape files. RMSRST will restore only the files that satisfy the wild card specification and were backed up from the specified account.

By default, if you do not specify /OA, RMSRST will apply no account criterion in restoring files: all files that satisfy the input file specification will be restored.

/SE:file-spec
/SE:(file-spec1,file-spec2[,...,file-spec10])

Direct RMSRST to restore the specified data files from magnetic tape container files. You can specify from 1 to 10 files; multiple file specifications must be enclosed in parentheses and separated by commas.

The file specification can contain only a file name, type, and version. Wild card characters are permitted. If you omit the version, RMSRST will apply the following defaults:

- All versions of the file will be restored if any one of the following conditions exists:
  - The file name or type is a wild card character.
  - The /OA switch is not specified for the container file.
  - The /OA switch is specified for the container file but contains wild card characters.
- The first version of the file that RMSRST encounters will be restored if either of the following conditions exists:
  - Neither the file name nor type is a wild card character.
  - The /OA switch is specified and contains no wild card characters in the file specification.

By default, if you do not specify /SE, RMSRST will restore all the files in the specified container file.

7.4 RMSRST PROCESSING

RMSRST restores data files that were backed up on disk or magnetic tape (container files) using the RMSBCK utility (see Chapter 6). RMSRST translates the container files into standard RMS-11 data files with the form, structure, contents, and attributes (with the exception of file placement) of the original data files when they were backed up.

RMSRST provides six processing features, as described in the following sections.

7.4.1 Explicit and Implicit Input File Specification

The RMSRST command string allows you to specify input files either explicitly or implicitly. File specifications with wild card characters (*) implicitly identify a collection of files. When you use wild card characters, you can restrict the number of files selected for processing by including the /BD input file switch (see Section 7.3.3), which selects disk files on the basis of their back-up date.

7.4.2 Choice of Restoration Account or Volume

You can use the /OA input file switch to select specific data files from the back-up files based on the account in which the data file originally resided (see Section 7.3.3).

7.4.3 Intersystem File Transfer

You can use RMSBCK and RMSRST to transfer files between different operating systems, for example, between RSX-11M/M-PLUS and VAX/VMS systems, and between RSTS/E and RSX-11M/M-PLUS or VAX/VMS systems.

Note, however, that VAX/VMS file versions are numbered in decimal and RSX-11M/M-PLUS file versions are numbered in octal. RMSBCK preserves the radix of the file's version number: a file that was created and backed up on an RSX-11M/M-PLUS system has an octal version number; a file that was created and backed up on a VAX/VMS system has a decimal version number. RMSRST uses this information to determine whether the version number should be converted to decimal or octal.

If the files you are restoring (with RMSRST Version 2.0) were backed up using RMSBCK Version 2.0, the version numbers will be converted automatically to the proper radix. RMSBCK Version 2.0 preserves the attributes of the files backed up, including the radix of the version number. You can override the conversion by specifying /NOCV.

If, however, the files were backed up using RMSBCK Version 1.8 or earlier, the version numbers will not be converted automatically. You can specify the /CV switch to ensure that the version numbers are converted to the proper radix. For example, if you are restoring files to RSX-11M/M-PLUS that were backed up using RMSBCK Version 1.8 or earlier on VAX/VMS, you can specify /CV to convert the version numbers from decimal to octal.

RSTS/E does not support file version numbers. No conversion is necessary, therefore, for files created and backed up on RSTS/E and restored to RSX-11M/M-PLUS or VAX/VMS. The /CV switch has no effect for these files.

See Section 7.3.1 for more information on the /CV switch.

You can use RMSDSP (with the /FU switch) to display the attributes of files backed up by RMSBCK; see Chapter 5, Example 5-7. The display will indicate the radix of the version number for RSX-11M/M-PLUS and VAX/VMS files. If no version number appears in the file specification, the file was created on a RSTS/E system.

NOTE

Be careful when restoring files created on RSTS/E systems to RSX-11M/M-PLUS and VAX/VMS systems. Files marked "Record format=Stream (RSTS/E)" in the RMSDSP display were not created using RMS-11 and have no attributes. RMS-11 can process these files on RSX-11M/M-PLUS and VAX/VMS systems; however, because RSTS/E does not maintain information on the largest record written in a file, these files may not be restored properly to other operating systems.

For information on restoring files created and backed up on RSX-11M/M-PLUS and VAX/VMS systems to RSTS/E systems, see the RSTS/E RMS-11 Utilities manual.

7.4.4 Data Integrity Checking

You can direct RMSRST to check data integrity as each file is restored. If you specify the /RA output file switch, RMSRST will read back the contents of the restored file. If you specify the /RC output file switch, RMSRST will read back, block by block, the contents of the output (restored) file and the contents of the input file and compare them. RMSRST automatically retries any read errors if processing continues to normal termination.

Thus, you have three choices in data integrity checking:

- You can specify neither /RA nor /RC and rely on software and hardware accuracy, backing up your files in minimum time.
- You can specify /RA and verify that the restored files can be read, adding reading time to the minimum time.
- You can specify /RC and verify that the restored files can be read and that they exactly match the back-up files, adding reading time and comparison time to the minimum time.

See Section 7.3.2 for descriptions of the /RA and /RC switches.

7.4.5 Extended Diagnostic Messages

Query mode allows you to specify that processing continues or terminates when RMSRST encounters an error condition. By default, query mode is enabled. When RMSRST encounters an error, it issues a diagnostic message to your terminal and prompts you:

```
Continue (Yes or No)?
```

Depending on your response, RMSRST continues or terminates processing. With this feature, you can ensure that RMSRST does not terminate processing of a group of files if errors are encountered in only one file.

You can disable query mode with the /NOQU output file switch. If RMSRST encounters an error, it will issue a diagnostic message to your terminal and immediately terminate processing.

See Section 7.3.1 for a description of the /[NO]QU switch.

7.4.6 Summary Listing

You can use the /SL switch to produce a summary listing of RMSRST processing, including data integrity checks and error messages. RMSRST will either write the summary to your terminal or into a file that you specify, creating a permanent record of restoration processing.

See the description of the /SL switch in Section 7.3.1, and the example summary listing in Section 7.5.

7.5 RMSRST EXAMPLES

```
>MOU MM0:PAYROL
>RST *.*;*/SU/SL=MM0:PAYROL.CON
%RST-I-FPC, Processing complete for file DB0:[70,5]PAYROL.DAT;2
%RST-I-FPC, Processing complete for file DB0:[70,5]PAYROL.LOG;1
%RST-I-FPC, Processing complete for file DB0:[70,5]PAYROL.DES;2
%RST-I-FPC, Processing complete for file DB0:[70,5]PAYROL.SEQ;2
%RST-I-FPC, Processing complete for file DB0:[70,5]PAYROL.MST;1
```

RMSRST searches the magnetic tape volume mounted on MM0: for the container file, PAYROL.CON. The files in the container file are restored to their original form in the default (user's) account on SY:.

Because the /SU switch is specified, if the current account contains a file with the same file name and extension, RMSRST will supersede the existing file with the file being restored.

```
>RST *.*;*=MM0:MASTER.BCK/SE:(STAT.DAT,ACCNT.DAT)
%RST-W-ER$FEX, File DB0:[70,5]STAT.DAT;1 already exists
%RST-W-ER$FEX, File DB0:[70,5]ACCNT.DAT;1 already exists
>RST *.*;*/SU=MM0:MASTER.BCK/SE:(STAT.DAT,ACCNT.DAT)
```

In both examples, RMSRST searches the volume mounted on MM0: for the first occurrence of the container file MASTER.BCK. RMSRST searches this file for the files STAT.DAT and ACCNT.DAT. RMSRST then attempts to restore these files to their original form in the default account on SY:.

In the first example, the files cannot be restored because files with the same name already exist in the default. In the second example, the /SU switch is specified so that the files in the default account (on SY:) are superseded by the files restored from tape.

```
>RST [70,2]*.*;*/SU/SL:RESTOR.LOG=DB0:[70,1]*.*;*/BD:02-APR-83,MM0:*.*;*
>TYP RESTOR.LOG
RMSRST VERSION  2.000 02-APR-1983 01:49:43
[70,2]*.*;*/SU/SL:RESTOR.LOG=DB0:[70,1]*.*;*/BD:02-APR-83,MM0:*.*;*

RST-I-FPC, Processing complete for file DB0:[70,2]CUST.FIL;1
RST-I-FPC, Processing complete for file DB0:[70,2]CUST.SEQ;1
RST-I-FPC, Processing complete for file DB0:[70,2]PAYROL.DAT;2
RST-I-FPC, Processing complete for file DB0:[70,2]STAT.DAT;1
RST-I-FPC, Processing complete for file DB0:[70,2]EMPLY.DAT;1
RST-I-FPC, Processing complete for file DB0:[70,2]ACCNT.DAT;1
RST-I-FPC, Processing complete for file DB0:[70,2]PAYROL.DAT;2
RST-I-FPC, Processing complete for file DB0:[70,2]PAYROL.LOG;1
RST-I-FPC, Processing complete for file DB0:[70,2]PAYROL.DES;2
RST-I-FPC, Processing complete for file DB0:[70,2]PAYROL.SEQ;2
RST-I-FPC, Processing complete for file DB0:[70,2]PAYROL.MST;1
```

RMSRST reads all files in the account [70,1] on DB0: that were backed up on April 2, 1983, and restores them to account [70,2] on SY:. Then, RMSRST restores the files within all container files on MM0:.

As it writes the files to SY:, RMSRST supersedes any existing files having the same file name and extension with the file being restored. RMSRST also writes a summary listing to RESTOR.LOG; the contents of this file are displayed.

APPENDIX A

RMS-11 FILE DEFINITION UTILITY: RMSDEF

A.1 PURPOSE

The interactive RMS-11 File Definition Utility (RMSDEF) creates RMS-11 files, allowing you to control all attributes of the files being created.

A.2 EFFECT

You specify file attributes by responding to requests for data and questions from RMSDEF. The method of questioning is outlined in Figure A-1. The figure shows the general flow of processing while Section A.4 shows actual messages and valid responses as well as the meaning of each attribute. You can also get help from the utility by typing a question mark (?) in response to any question or request for data.

RMSDEF can also build an indirect command file while you operate the utility. This command file can be used thereafter to (re)create file(s) and can be modified to create other similar files. See Sections A.3.1 and A.4.1.

NOTE

Command files generated by RMSDEF are not compatible across operating systems.

RMSDEF, however, does not write records into the file. The actual data contents of the file must be loaded after the RMSDEF utility creates the file. You can use either an application program, the RMSCNV utility, or the RMSIFL utility to write records into the file.

NOTE

RMSDEF interprets all quantities you enter as decimal numbers.

The following list indicates the information that RMSDEF always requests as well as the requests that depend on specifications typed.

1. Command File?
    a. If yes, file specification
    b. If yes, create only command file or create both RMS-11 and command files?
2. File Specification and Supercession Information
3. Data Structure
    a. Minimum
        1. File organization
        2. Record format
        3. Maximum record size
        4. CARRIAGE RETURN control?
    b. Possible
        1. Size of fixed control area for VFC records
        2. Maximum number of records in a relative file
        3. Block-spanning records in a sequential file?
        4. FORTRAN character control if no CARRIAGE RETURN control?
4. Key Definition (indexed files only)
    a. Minimum
        1. Position of key
        2. Size of key
        3. Data type
        4. Name of key
        5. Duplicate keys?
    b. Possible
        1. Change keys if duplicatable? if alternate key
        2. Null key value? if alternate key
        3. Null key character if null key value
5. File Structure
    a. Minimum
        1. Areas? (indexed files only)
        2. Placement control?

3. Initial allocation quantity
4. Default extension quantity
5. Contiguous?

b. Possible

1. Location if placement control
2. Exactly if placement control?
3. Type of alignment if placement control and areas

6. Data Allocation (indexed files only)

a. Minimum

1. Number of bytes in data buckets filled
2. Number of bytes in index buckets filled

b. Possible

1. Area containing index level 0 for each key if areas
2. Area containing index levels 2+ for each key if areas
3. Area containing index level 1 if areas

7. Protection

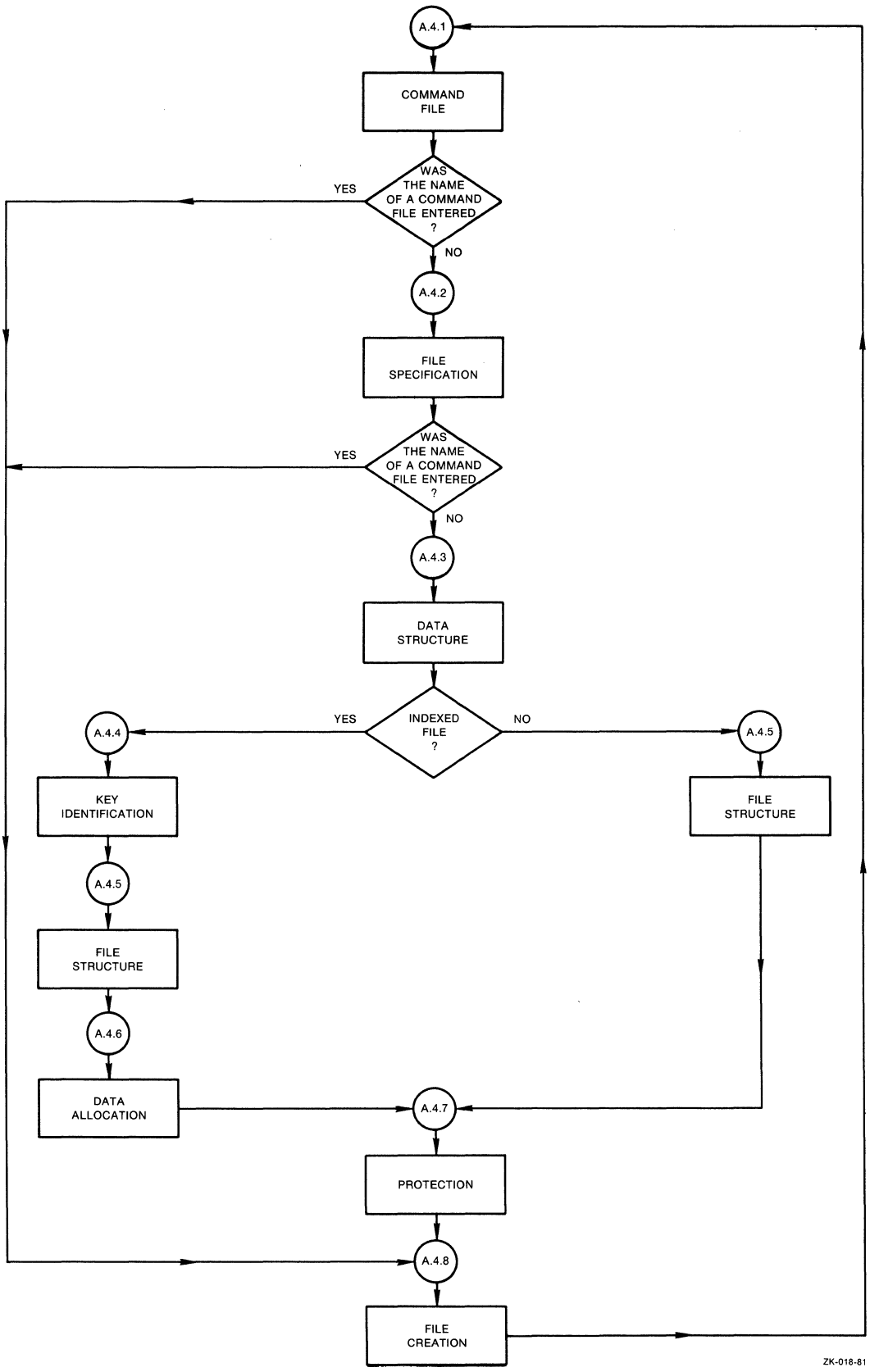

Figure A-1: RMSDEF Processing Flowchart

A.3 CALL AND TERMINATION

A.3.1 Calling the Utility

The RMSDEF utility can be either installed or uninstalled. If RMSDEF is installed, invoke it with the command:

```
DEF [command-string]
```

where command-string is one of the following:

@file-spec — if you want RMSDEF to read a command file named file-spec that was previously generated by RMSDEF (see Section A.4.1). The utility processes the file, printing messages only when it comes to the file creation phase (see Section A.4.8).

HELP or ? — if you want an introductory HELP message for the RMSDEF utility.

/ID — if you want the version number of the RMSDEF utility installed on your system. RMSDEF prints the following message:

```
THIS IS THE RMS DEFINE UTILITY, VERSION 2.000
```

where nn is the patch level of the utility.

If you do not include a command string, the utility prints:

```
DO YOU WANT TO GENERATE A COMMAND FILE FOR FUTURE USE (NO)?
```

See Section A.4 for the complete dialog sequence.

If RMSDEF is uninstalled on RSX-11M/M-PLUS systems, type the MCR command:

```
RUN $RMSDEF
```

The utility prints:

```
DO YOU WANT TO GENERATE A COMMAND FILE FOR FUTURE USE (NO)?
```

See Section A.4 for the complete dialog.

A.3.2 Terminating the Utility

You may terminate RMSDEF at any time by typing a CTRL/Z. The current keyboard monitor or command interpreter resumes control of your interface.

A.4 PROCESS

A.4.1 Command File

1. The terminal prints:

```
DO YOU WANT TO GENERATE A COMMAND FILE FOR FUTURE USE (NO)?
```

Type one of the following:

Y	if you want to enter a file-spec for an indirect command file and RMSDEF to write entries into the file as you move through the utility. Go to step 2.
N or (RET)	if you do not want to build a command file. Go to Section A.4.2.
@file-spec	if you have already built a command file with RMSDEF and want RMSDEF to read it now and create the specified file. Go to Section A.4.8.
/ID	if you want the utility to identify itself with a version number. RMSDEF prints the following message: THIS IS THE RMS DEFINE UTILITY, VERSION 2.0 where: nn is the patch level of the utility.

2. The terminal prints:

   ENTER FILE SPECIFICATION FOR COMMAND FILE:

Type a file specification. See the RSX-11M/M-PLUS RMS-11 User's Guide restrictions on file specifications defined by your operating system documentation.

3. The terminal prints:

   DO YOU WANT TO CREATE THE FILE YOU WILL BE DESCRIBING (NO)?

Type one of the following:

Y	if you want RMSDEF to create the RMS-11 file being specified as well as the command file.
N or (RET)	if you do not want to create the RMS-11 file, only the command file that can be used to create the file later.

NOTE

The command file created by RMSDEF takes the following form shown. The utility follows each comment with the appropriate sequence of your entries, each on a separate line. Where you press RETURN to accept the default value, RMSDEF places the CR/LF character sequence on a separate line in the command file.

```
;THE FIRST QUESTION ASKS FOR THE FILE SPECIFICATION.

;THE NEXT QUESTIONS DEAL WITH FILE ORGANIZATION & RECORD
ATTRIBUTES

;THE FOLLOWING QUESTIONS DEAL WITH KEYS (for indexed files only)
```

;THE NEXT QUESTIONS DEAL WITH ALLOCATION AND PLACEMENT ATTRIBUTES

;THE NEXT QUESTIONS ASK ABOUT FILL SIZES FOR KEYS (for indexed files only)

;THE FOLLOWING QUESTIONS DEAL WITH FILE PROTECTION

A.4.2 File Specification

1. The terminal prints:

   ENTER FILE SPECIFICATION:

   Type one of the following:

   file-spec — if you want to create (or simulate the creation of; see Section A.4.1, step 3) an RMS-11 file. Type a file specification as defined by your operating system documentation; see also the RSX-11M/M-PLUS RMS-11 User's Guide for RMS-11 restrictions on file specifications. Go to Section A.4.3.

   @file-spec — if you have built a command file (see "Command File," Section A.4.1) and want RMSDEF to use it and create the file specified. Go to Section A.4.8.

2. The terminal prints:

   IF THE FILE ALREADY EXISTS, DO YOU WANT TO SUPERSEDE IT (NO)?

   It is possible that a file already exists with the file specification you entered in step 1. Note that a conflict will result if you specified a version number in that specifiction. You must tell RMSDEF what to do if such a conflict occurs when it tries to create the file you describe during this interactive session.

   Type one of the following:

   Y — if you want RMSDEF to supersede an existing file.

   N or (RET) — if you do not want RMSDEF to create the file you will describe by superseding a file that exists.

   CAUTION

   1. If you say "N" here, and
   2. If you are generating a command file, and
   3. If you describe other files after this one in that command file, and
   4. If, when you use the command file, a file with the file specification entered in step 1 exists:

   RMSDEF prints the following error message and terminates processing of the command file.

A FILE WITH THE FILE SPECIFICATION YOU ENTERED ALREADY EXISTS

A.4.3 Data Structure

1. The terminal prints:

   ENTER FILE ORGANIZATION (SEQ):

   Type one of the following:

SEQ or (RET)	for sequential organization
REL	for relative organization
IDX	for indexed organization

   NOTE

   If you indicated a magnetic tape device in the file specification, RMSDEF does not request a file organization. Since a magnetic tape file requires sequential organization, the utility prints:

   SINCE YOU SPECIFIED A NON-DISK DEVICE, YOUR FILE ORGANIZATION MUST BE SEQUENTIAL!

2. The terminal prints:

   ENTER RECORD FORMAT (VAR):

   Type one of the following:

VAR or (RET)	if the records in the file have differing, or variable, lengths.
FIX	if the records in the file have the same, or a fixed, length.
VFC	if each record in the file has a control area with a fixed length and a data area of no standard length, that is, variable with fixed control.
STM	if the records in the file have no specific format, but are delimited only by record terminator characters. The stream format is permitted for sequential disk files only.
UDF	if there are no records (or you do not want RMS-11 to recognize records) in the file; this format is used only for block I/O files, such as RMS-11 back-up files. The undefined format is permitted for sequential disk or tape files only.

   RMSDEF rejects a format that conflicts with the file organization specified, for instance, STM for indexed files.

3. If you specified VFC in step 2, the terminal prints the following message; otherwise, go to step 4.

   ENTER SIZE OF FIXED CONTROL AREA (2):

   Type the decimal number of bytes in the fixed control area of each record in the file. The minimum size is 1 byte; the maximum size is 255 bytes; the default is 2 bytes.

4. The terminal prints:

```
ENTER MAXIMUM RECORD SIZE:
```

or

```
ENTER MAXIMUM RECORD SIZE (0):
```

Type a decimal number indicating the maximum number of bytes in any record in the file. RMS-11 checks this value whenever a record access operation is requested for this file: if the record specified exceeds the maximum size, RMS-11 returns an error. A size of zero disables the RMS-11 check, but a nonzero value is required for all relative files and all files with fixed-length records.

5. If you specified REL in step 1, the terminal prints the following message; otherwise, go to step 6.

```
ENTER MAXIMUM NUMBER OF RECORDS (0):
```

Type a decimal number indicating the maximum number of records that this relative file will contain. RMS-11 checks this value whenever a record access operation is requested for this file: if the relative record number specified exceeds the maximum record number, RMS-11 returns an error. Pressing RETURN sets the number to zero, which disables the RMS-11 check. The zero allows the file to contain as many records as is physically possible (the technical maximum is 2.14748 x 10**9 or 2,147,800,000).

6. If you specified SEQ in step 1, the terminal prints the following message; otherwise, go to step 7.

```
WILL YOU ALLOW RECORDS TO CROSS BLOCK BOUNDARIES (YES)?
```

Type one of the following:

Y or RET — if you want records to cross block boundaries.

N — if you do not want records to span blocks. If you specified FIX in step 2 and a maximum record size greater than 512 in step 4, the terminal prints:

```
SINCE YOU SPECIFIED FIXED LENGTH RECORDS, YOU MUST HAVE A
MAXIMUM RECORD SIZE LESS THAN 512 BYTES (THE SIZE OF ONE
BLOCK) OR YOU MUST ALLOW RECORDS TO CROSS BLOCK BOUNDARIES.
PLEASE CHANGE ONE OF THESE.
```

RMSDEF repeats steps 4 and 6. Change either your maximum record size or the answer to crossing block boundaries.

7. The terminal prints:

```
DO YOU WANT CARRIAGE RETURN CONTROL (YES)?
```

Type one of the following:

Y or RET — if you want each record to be preceded by a line feed character and followed by a carriage return character when it is written to a unit record device (printer, terminal, and so on). See the following note and go to appropriate section.

N — if you do not want CARRIAGE RETURN control and/or you do want FORTRAN character control. Go to step 8.

8. The terminal prints:

   DO YOU WANT FORTRAN CHARACTER CONTROL (NO)?

   Type one of the following:

Y	if you want the first byte of each record to be interpreted as a FORTRAN forms control character when the record is written to a unit record device.
N or (RET)	if you do not want FORTRAN character control.

   NOTE

   If you indicated a magnetic tape device in the file specification, at this point RMSDEF requests:

   ENTER MAGTAPE BLOCK SIZE (512):

   Type a decimal number between 18 and 8192 representing the number of bytes in each tape block. The number should be a multiple of 4; if it is not, RMS-11 rounds the number up to the next multiple of 4 before writing it as an attribute. Pressing RETURN sets the size to the default of 512 bytes.

   The utility then bypasses other processing and immediately requests protection information (see Section A.4.7).

A.4.4 Key Definition

As indicated by Figure A-1, this section applies only to indexed files. RMSDEF begins this portion of dialog with the message:

IT'S TIME TO DEFINE THE PRIMARY KEY

1. The terminal prints:

   ENTER DATA TYPE (STR):

   Type one of the following:

STR or (RET)	if your key value is a string of alphanumeric characters.
INT	if your key value is a 2- or 4-byte signed integer.
BIN	if your key value is a 2- or 4-byte unsigned binary number.
PAC	if your key value is a packed decimal number.

   See the RSX-11M/M-PLUS RMS-11 User's Guide for a discussion of key data types.

2. The terminal prints:

   ENTER POSITION OF KEY:

   Type a decimal number indicating the position of the first byte of the key within each record. For instance, if the key starts with the first byte of the record, its position is 0. The second byte has position 1 and so on.

   A position number must be specified for each segment of a segmented string key; the numbers are separated by commas and enclosed in parentheses. Integer, binary, and packed decimal keys cannot be segmented. For example, if a key has three segments that start at the first, fourth, and sixteenth positions, they are described as follows:

   (0,3,15)

3. The terminal prints:

   ENTER SIZE OF KEY:

   Type the decimal number of bytes in the key, that is, its length. Valid lengths depend on the data type entered in step 1:

Data Type	Length Restrictions
STR	Minimum = 1 byte Maximum = 255 bytes
INT	2 or 4 bytes
BIN	2 or 4 bytes
PAC	Minimum = 1 byte Maximum = 16 bytes

   A length must be specified for each segment of a segmented string key; the numbers are separated by commas and enclosed in parentheses. A length must be typed for each position number specified in step 1, but the sum of all lengths cannot exceed 255. Integer, binary, and packed decimal keys cannot be segmented. For example, the lengths of the keys in the step 2 example would be entered as follows:

   (2,2,16)

4. The terminal prints:

   ENTER NAME OF KEY (NONE):

   Type one of the following:

   RET — if you do not want to specify a key name.

   name — if you want to name the key being defined; up to 32 ASCII characters are allowed.

5. The terminal prints:

   WILL YOU ALLOW DUPLICATE KEYS (NO)?

Type one of the following:

Y if the file may contain more than one record with the same value for this key. Alternate keys must be specified as duplicatable before they can be specified as changeable (in step 6).

N if each record in the file must have a unique value for this key. RMS-11 returns an error if duplication is attempted; that is, put or update operations fail when the record has a value in this key field exactly like a record already in the file.

Defaults and the values of dflt are:

Primary Key No

Alternate Keys Yes

When defining the primary key, go to step 10. Otherwise, go to step 6.

6. If you specified Yes in step 5, the terminal prints the following message; otherwise, go to step 7.

WILL YOU ALLOW KEYS TO CHANGE (YES)?

Type one of the following:

Y or RET if this alternate key can change values during an update operation; that is, the record may be read with one value for the key and rewritten with another value for the same key.

N if this alternate key must not change after the record is originally created.

7. The terminal prints:

DO YOU WISH TO DEFINE A NULL KEY VALUE (NO)?

Type one of the following:

Y if you want the file to contain some records that cannot be accessed via this key. See the RSX-11M/M-PLUS RMS-11 User's Guide. Go to step 8 only if you specified a string key (step 1). Otherwise, go to step 9; the other key data types have a null key value of 0.

N or RET if you do not want to specify a null value for this key. Go to step 9.

8. The terminal prints:

ENTER NULL KEY VALUE CHARACTER:

Type one of the following:

c the single character itself, if it is not #, ?, @, or SPACE.

#40 for SPACE.

#43 for the reserved character #.

#77 for the reserved character ?.

#100 for the reserved character @.

#nnn any octal byte value (000-377), specified by nnn.

9. The terminal prints:

JUST FINISHED ALTERNATE KEY nnn

where:

nnn starts with 1 and is incremented when you answer Yes to the next question (step 10).

10. The terminal prints:

DO YOU WANT TO DEFINE MORE KEYS (NO)?

Type one of the following:

Y if you want to define more keys for the file. You may define up to 254 alternate keys; however:

- The RMSIFL and RMSCNV utilities do not read higher than the ninth alternate key; that is, the /KR switch must be less than or equal to nine (see Chapters 3 and 4).
- Your application language may not support that many keys. See your language manual.

RMSDEF then requests the information (in steps 1-9) for each alternate key indicated; the alternate keys are defined in order, beginning with the first alternate after the primary key is defined.

N or RET if all keys for this file are defined.

A.4.5 File Structure

1. If you specified IDX for indexed file organization, the terminal prints the following message; otherwise, go to step 2.

DO YOU WANT TO DEFINE AREAS (NO)?

Type one of the following:

Y if you want parts of this file to be logically different, with different attributes. See the RSX-11M/M-PLUS RMS-11 User's Guide. The following questions (steps 2-9) are asked for each area.

N or RET if you want this indexed file located in one area. RMSDEF requests the following information (steps 2-9) for the file once.

2. The terminal prints:

   DO YOU WANT PLACEMENT CONTROL (NO)?

   Type one of the following:

Y	if you want to specify an exact location on disk for this file or area. Go to step 3.
N or RET	if you do not want to specifically locate this file or area. Go to step 6.

3. On RSX-11M systems, the terminal prints the following message only for areas 1+ in indexed files.

   ENTER TYPE OF ALIGNMENT (LBN):

   Type one of the following:

LBN or RET	if the location you will specify in step 4 is a logical block number (LBN) on the disk volume.
VBN	valid only for areas 1+ in indexed files, if the location you will specify in step 4 is a virtual block number (VBN) already established within the file; that is, you are trying to closely align this area with a defined area.

4. The terminal prints:

   ENTER LOCATION:

   Type the decimal number location of the first block for this file area. There is no default. For the first area defined, this number is a logical block number.

5. The terminal prints:

   EXACTLY (NO)?

   Type one of the following:

Y	if this file or area must start in the exact LBN location specified in step 4. If this location is not available, RMSDEF prints an error message after it tries to create the file. Exact VBN locations are already taken, by definition.
N or RET	if you will accept the closest approximation of the LBN or VBN location specified in step 3, when the exact location is not available.

6. The terminal prints:

   ENTER INITIAL ALLOCATION IN BLOCKS (0):

   Type a decimal number indicating the initial size of the file or area in blocks. Pressing RETURN sets the value to zero: the area will be created, but it must be extended before any records can be written into it. Since automatic file extension is time consuming, the file should be fully allocated when it is created (see the RSX-11M/M-PLUS RMS-11 User's Guide).

NOTE

If you have specified placement control (step 2), you should type a nonzero allocation quantity. An initial allocation of zero blocks essentially nullifies placement control: the file processor does not allocate any blocks to the file, so it does not use the placement control information. For example, you can place multiple files at the same LBN location, even answering YES to EXACTLY? (step 5), as long as you use zero allocation quantities for all but one.

7. If you specified REL or IDX for file organization, the terminal prints the following message; otherwise, go to step 8.

   BUCKET SIZE (1):

   Type a decimal number indicating the number of blocks in a bucket for this file or area. The minimum is the number of blocks that can contain one record (according to the size specified; see Section A.4.3, step 4); the maximum is 32 blocks; the default is 1. This number determines the number of blocks read into memory during each file access operation and therefore affects processing speed and the amount of memory a program accessing this file requires. See the RSX-11M/M-PLUS RMS-11 User's Guide.

8. The terminal prints:

   ENTER DEFAULT EXTENSION QUANTITY IN BLOCKS (0)

   Type a decimal number indicating the number of blocks that should be added to the file or area each time RMS-11 extends it. The default extension quantity (DEQ) should be a multiple of the bucket size. RMS-11 requests this number of blocks from the operating system when it extends the file or area.

   Pressing RETURN sets the value to zero: RMS-11 adds only the minimum space required each time it expands the file or area. A definite, but reasonable extension quantity speeds processing.

9. The terminal prints one of the following messages:

   DO YOU WANT A CONTIGUOUS AREA (NO)?

   or

   DO YOU WANT A CONTIGUOUS FILE (NO)?

   Type one of the following:

   Y if you want the disk space for this file or area allocated in contiguous blocks. If the file processor cannot find that many contiguous blocks, it will not create the file, even if enough noncontiguous blocks are available.

   A contiguous file or area may be extended although the disk space added may not be contiguous with the original allocation.

10. If you answered Yes in step 1 (you have an indexed file), the terminal prints the following messages; otherwise, go to the next appropriate section.

    ```
    JUST FINISHED WITH AREA NUMBER nnn
    DO YOU WANT TO DEFINE MORE AREAS (NO)?
    ```

    Type one of the following:

    Y — if you want to specify attributes for another area of your file. Areas are numbered sequentially, starting with zero. The areas are associated with the index and data portions of the file in the next section of the utility. Go to step 2.

    N or RET — if you have defined enough areas for this file. Go to step 11.

11. If you defined one or more areas with a DEQ of zero, the terminal prints the following message; otherwise, go to the next section.

    ```
    ENTER FILE DEFAULT EXTENSION QUANTITY (0):
    ```

    Type a decimal number indicating the number of blocks that should be added to the file each time RMS-11 extends it. The DEQ should be a multiple of the bucket size. Pressing RETURN sets the value to zero: RMS-11 will add only the minimum space required each time it expands the file. A definite, but reasonable extension quantity speeds processing.

A.4.6 Data Allocation

As indicated by Figure A-1, this section applies only to indexed files. RMSDEF begins this portion of dialog with the message:

```
IT IS TIME FOR AREA NUMBERS AND FILL FACTORS FOR KEYS.
```

The questions are asked for each key defined (see Section A.4.4). RMSDEF begins the first session with the message:

```
THESE QUESTIONS ARE FOR THE PRIMARY KEY
```

Any further sessions begin with the message:

```
THESE QUESTIONS ARE FOR ALTERNATE KEY NUMBER nnn
```

where:

nnn starts with 1.

1. The terminal prints:

    ```
    ENTER AREA NUMBER FOR DATA BUCKETS FOR THIS KEY (0):
    ```

    Type an integer (0-n) indicating the area already defined (see Section A.4.4) that should contain the data portion (level 0) of this key's index.

2. The terminal prints:

```
THE BUCKET SIZE IS nn
ENTER FILL NUMBER FOR DATA BUCKETS (0):
```

Type a decimal number of bytes in each of this key's data buckets that should be used during the original population of the file. This number is honored by the RMSCNV and RMSIFL utilities and may be honored by MACRO-11 programs. See the RSX-11M/M-PLUS RMS-11 User's Guide.

Pressing RETURN sets the number to zero, indicating that buckets will be filled and that no free space will be available in existing buckets for records added after the file is populated.

3. The terminal prints:

```
ENTER AREA NUMBER FOR INDEX BUCKETS FOR THIS KEY (0):
```

Type an integer (0-n) indicating the area already defined (see Section A.4.4) that should contain the upper portions (levels 2+) of the index for this key.

4. The terminal prints:

```
THE BUCKET SIZE IS nn
ENTER FILL NUMBER FOR INDEX BUCKETS (0)
```

Type a decimal number of bytes in each of this key's index buckets that should be used during the original population of the file. This number is honored by the RMSCNV and RMSIFL utilities and may be honored by MACRO-11 programs.

Pressing RETURN sets the number to zero, indicating that buckets will be filled and that no free space will be available in existing buckets for records added after the file is populated.

5. The terminal prints:

```
ENTER AREA NUMBER FOR THE LOWEST INDEX LEVEL FOR THIS KEY (0):
```

Type an integer (0-n) indicating the area already defined (see Section A.4.4) that should contain the lowest index level (level 1) portion of this key's index.

If the area you specified for the upper portions of the index (levels 2+) and the area you specified for level 1 have different bucket sizes, RMSDEF prints the following message and returns to step 3:

```
THE AREA ASSOCIATED WITH THE LOWEST LEVEL INDEX BUCKET
HAS A DIFFERENT BUCKET SIZE THAN THE AREA ASSOCIATED WITH
THE HIGHER INDEX BUCKET. TRY BOTH AGAIN.
```

A.4.7 Protection

1. The terminal prints:

   ```
   SPECIFY PROTECTION BY CLASS:
   OWNER (RWED ALLOWED):
   ```

   Type one of the following:

(RET)	if you want this file available to access (Read, Write, Extend or Execute, Delete) by the account current when the file is created.
NONE	if you do not want the file owner to have access to this file after it is created.
R, W, E, and/or D	if you want to specify a level of protection between none and all. You may specify one or more of the letters representing Read, Write, Extend, and Delete without separation.

2. The terminal prints:

   ```
   GROUP (RWED ALLOWED):
   ```

   Type one of the following:

(RET)	if you want this file available to access (Read, Write, Extend or Execute, Delete) by accounts with the same group or project number as the owner's account.
NONE	if you do not want the group members to have access to this file after it is created.
R, W, E, and/or D	if you want to specify a level of protection between none and all. You may specify one or more of the letters representing Read, Write, Extend, and Delete without separation.

3. The terminal prints:

   ```
   SYSTEM (RWED ALLOWED):
   ```

   Type one of the following:

(RET)	if you want this file available to access (Read, Write, Extend or Execute, Delete) by system privileged accounts.
NONE	if you do not want privileged accounts to have access to this file after it is created.
R, W, E, and/or D	if you want to specify a level of protection between none and all. You may specify one or more of the letters representing Read, Write, Extend, and Delete without separation.

4. The terminal prints:

   WORLD (R ALLOWED):

   Type one of the following:

RET	if you want this file available for Read access only by all accounts outside the owner, group, and privileged accounts.
NONE	if you do not want other accounts to have access to this file after it is created.
R, W, E, and/or D	if you want to specify a level of protection other than none and Read only. You may specify one or more of the letters representing Read, Write, Extend, and Delete without separation.

A.4.8 File Creation

The RMSDEF utility attempts to create the file.

Success -- If RMS-11 does not return an error, the utility prints:

   YOUR FILE HAS BEEN CREATED -- file-spec

If you chose to create a command file (See Section A.4.1), RMSDEF also prints the following message. Otherwise, the utility requests you to enter another file specification (Section A.4.2).

   YOUR FILE HAS BEEN PROCESSED AND A COMMAND FILE GENERATED -- file-spec
   DO YOU WANT TO CLOSE THE COMMAND FILE (NO)?

Type one of the following:

Y	if you are finished specifying files for the command file. The utility returns to the question about command file generation (see Section A.4.1).
N or RET	if you want to specify another RMS-11 file and you want the command file to include your input. RMSDEF continues to use the command file originally specified and to obey your answer to the DO YOU WANT TO CREATE THE FILE YOU WILL BE DESCRIBING? question (see Section A.4.1, step 3). The utility returns to the request for a file specification (see Section A.4.2).

You may start the process to create another file or type CTRL/Z to terminate the utility.

Error -- RMSDEF allows you to recover from one type of creation error. The terminal prints an error description, followed by:

   DO YOU WISH TO REENTER THE ENTIRE FILE SPECIFICATION (YES)?

Type one of the following:

Y or (RET) — if you know how to correct the error and/or want to enter another file specification. The utility requests the file specification and attempts to create the file using it.

Y — if you do not know how to correct the error and/or want to start again. RMSDEF returns to the file specification request (Section A.4.2).

Other errors result in one of the following messages:

- THIS FILE CANNOT BE CREATED
  RMS STATUS CODE = nnnnnn SYSTEM CODE = nnnnnn
- THIS FILE CANNOT BE CREATED SINCE THE FILE ALREADY EXISTS AND YOU DID NOT SPECIFY SUPERSEDE.

The RSX-11M/M-PLUS RMS-11 User's Guide defines the codes represented by nnnnnn.

The utility returns to the file specification request to let you restart the definition process (see Section A.4.2).

A.5 RMSDEF MESSAGES

BAD UIC, TRY AGAIN

Explanation: The account number in the file specification is not valid for your operating system for one of the following reasons:

- Either one or both parts of the account number exceeds the maximum value allowed; the range is 1 through 377 (octal).
- The specified account has not been set up by the system manager.

User Action: Type the file specification with a valid account number.

THE DEFAULT EXTENSION QUANTITY MUST BE A MULTIPLE OF THE BUCKET SIZE

Explanation: You typed a decimal number of blocks for default extension quantity, but that number is not an integral multiple of the bucket size you specified.

User Action: Type the next higher multiple of bucket size for default extension quantity.

** DEVICE IS FULL **

Explanation: RMSDEF attempted to create the file you defined, but could not allocate the number of blocks you specified in the way you specified for one of the following reasons:

- There exist insufficient free clusters to allocate the entire file.
- There exist insufficient free contiguous clusters to allocate.
- There is insufficient free space in the User File Directory (UFD) to map all allocated clusters of the file.
- The allocation is too large for the device.

User Action: Determine the cause of the error. The parameters can be altered to compensate.

** DEVICE NOT AVAILABLE **

Explanation: RMSDEF attempted to create the file you defined, but could not access the specified device because it was not ready or online.

User Action: Make the specified device ready or specify another device; and try again.

DIRECTORY NOT FOUND

Explanation: RMSDEF attempted to create the file you defined, but the file processor indicates that the specified account has not been set up.

User Action: Specify another account number or request the system manager to create the specified account; and try again.

ERROR IN LUN ASSIGNMENT

Explanation: RMSDEF assigned a logical channel erroneously.

User Action: Recall the utility and try the dialog again.

ERROR WITH INDIRECT FILE

Explanation: You directed RMSDEF to read an indirect file, but the utility detected an error in the file that is not covered by other error messages.

User Action: Recall the utility and respecify the indirect file.

FILE SPECIFICATION IS TOO LONG

Explanation: You directed RMSDEF to read an indirect file, but the specified file contains either a file specification longer than 120 bytes or a line (record) longer than 80 bytes.

User Action: Edit the indirect file, shortening the file specification or line. Recall the utility and specify the indirect file.

I/O ERROR IN PROCESSING INDIRECT FILE

Explanation: You directed RMSDEF to read an indirect file, but the utility detected an I/O error during its processing.

User Action: Recall the utility and respecify the indirect file. If this error recurs, re-create the indirect file.

ILLEGAL DEVICE, TRY AGAIN

Explanation: RMSDEF attempted to create the file you defined, but encountered an error: either the specified device is not on your computer system or the utility has erroneously assigned its logical channel to another device.

User Action: If the device is not on your system and is not a logical name, specify another device and try again.

ILLEGAL FILE SPECIFICATION, TRY AGAIN

Explanation: RMSDEF detected a syntax error in the file specification.

User Action: Retype the file specification correctly.

ILLEGAL KEY SIZE. TYPE ? FOR HELP.

Explanation: You typed a key size that is not valid for the key data type you specified.

User Action: Type a question mark (?) to print a HELP message showing the size restrictions for the different data types. Then, type a valid key size.

ILLEGAL VALUE. TYPE ? FOR HELP.

Explanation: You typed a value that is not a proper reply to the current question or request.

User Action: Type a question mark (?) to print a HELP message that should help you determine the proper values. Then type a proper value for the question or request.

INVALID DATA IN COMMAND FILE

Explanation: You directed RMSDEF to read an indirect file, but the a line in that file did not satisfy the utility's data requirements. If you specified the indirect file at command level, RMSDEF returns to that level. However, if you specified the indirect file after the utility printed its prompt, RMSDEF continues its dialog with you rather than the indirect file.

User Action: Correct or re-create the indirect file and try again, or continue the interactive dialog.

KEY EXCEEDS MAXIMUM RECORD SIZE.

Explanation: You typed a size for the key being defined which extends the key beyond the end of the record you defined.

User Action: Type a smaller size or terminate the current session with a CTRL/Z and start again, defining a larger maximum record size.

MAXIMUM NESTING DEPTH EXCEEDED IN INDIRECT FILE

Explanation: An indirect command file contains an indirect command. RMSDEF does not permit nesting of indirect commands.

User Action: Type the contents of one or both of the indirect command files as a sequence of individual command strings.

OPEN ERROR IN INDIRECT FILE

Explanation: You directed RMSDEF to read an indirect file, but the utility could not open the file. The file may not exist as specified; it may be locked; or there is some other problem with the file.

User Action: Correct the problem and try again.

** PRIVILEGE VIOLATION **

Explanation: RMSDEF attempted to create the file you defined, but the operating system denied the utility access.

User Action: Correct the problem and try again.

SEGMENTS ARE NOT ALLOWED WHEN THE KEY IS NUMERIC

Explanation: You attempted to indicate segments for a key you defined as numeric (integer, binary, or packed decimal).

User Action: Enter one size or position.

SUM OF SEGMENTED KEY SIZES IS GREATER THAN 255.

Explanation: You typed a series of sizes for a segmented key. However, the sum of those sizes is greater than 255 bytes, the maximum size for one key.

User Action: Type the series of sizes again, making one or more of the segments smaller so that their total is 255 or less. You could make two keys out of the one you are attempting to define: reduce the segments in this key and when RMSDEF asks if you have more keys to define, type Y and specify another key containing the rest of the segments.

SYNTAX ERROR IN INDIRECT FILE SPECIFICATION

Explanation: You typed a file specification for an indirect command file, but there was a syntax error in it.

User Action: Type the file specification correctly.

THE BUCKET FILL SIZE IS GREATER THAN THE ACTUAL BUCKET SIZE.

Explanation: You typed a decimal number of bytes as a fill number. However, that number is greater than the total bytes in the bucket you specified for the current area.

User Action: Type a fill number smaller than the bucket size.

THE NUMBER OF VIRTUAL BLOCKS PER BUCKET IS NOT BIG ENOUGH GIVEN THE MAXIMUM RECORD SIZE YOU SPECIFIED.

Explanation: You typed a bucket size that would not contain the largest record you indicated in the maximum record size question.

User Action: Enter a larger bucket size; or terminate the utility, recall it, and redefine the file with no or a larger maximum record size.

TOO MANY AREAS

Explanation: You typed Y in response to the question:

DO YOU WANT TO DEFINE MORE AREAS (NO)?

However, you have defined 255 areas, the maximum for one file.

User Action: Continue with the RMSDEF interactive session and complete the file definition.

TOO MANY KEYS

Explanation: You typed Y in response to the question:

DO YOU WANT TO DEFINE MORE KEYS (NO)?

However, you have defined 255 keys, the maximum for one file.

User Action: Continue with the RMSDEF interactive session and complete the file definition.

TOO MANY SEGMENTS

Explanation: You are defining a segmented key. However, the series of positions or the series of sizes you typed contained more than eight numbers, the maximum segments for one key.

User Action: Type the number series again, using no more than eight values.

WARNING - BE AWARE THAT THE FILENAME HAS BEEN TRUNCATED.

Explanation: In the file specification you just typed, you specified a file name longer than nine characters. RMSDEF continues the interactive session, but the utility uses only the maximum number of characters allowed in the file name when it tries to create the file. For example, you type the file specification:

ACCOUNTING.DAT

RMSDEF uses only ACCOUNTIN.DAT when it tries to create the file.

User Action: Continue the session unless you cannot accept the file name truncation.

WARNING - BE AWARE THAT THE FILE TYPE HAS BEEN TRUNCATED.

Explanation: In the file specification you just typed, you specified a file type longer than three characters. RMSDEF continues the interactive session, but the utility uses only the first three characters in the file type when it tries to create the file. For example, you type the file specification:

PAYDAY.FILE

RMSDEF uses only FIL when it tries to create the file.

User Action: Continue the session unless you cannot accept the shorter type.

YOU ENTERED A DIFFERENT NUMBER OF SEGMENTS IN THE KEY POSITION QUESTION.

Explanation: You are defining a segmented key. However, the series of key sizes you typed does not contain the same number of segments as the series of key positions you typed for the last question.

User Action: Type the key size series again, specifying the same number of segments that you used in the key position series.

YOU HAVE SPECIFIED A VIRTUAL BLOCK NUMBER WHICH IS NOT WITHIN THE BOUNDARIES OF THE FILE.

Explanation: You are using virtual block number (VBN) alignment to place a file area other than area 0. However, the number you typed in response to the LOCATION: request is not a valid VBN for the file you have defined to this point.

User Action: Recalculate the extent of the file you have defined to this point by adding the initial allocation quantities for all the areas defined, starting with area 0. Use the sum to help you type a valid VBN.

YOU ORIGINALLY GAVE A BAD FILE SPECIFICATION WHICH WAS OUTPUT TO YOUR COMMAND FILE. YOU MUST EDIT THE CHANGE IN YOURSELF.

Explanation: When you typed a file specification for the file you were defining, you made an error. RMSDEF allowed you to retype the file specification at the end of the interactive session, after it failed to create the file. However, you also directed the utility to create an indirect file during the session. That indirect file now has an incorrect file specification at its beginning.

User Action: Before you can use the indirect file, you must change the invalid file specification with a file editor.

APPENDIX B

RMSDES AREA DEFINITION

One option in using RMSDES is to allow the utility to define areas by default. In most cases, this is the best option and will provide optimal area definitions.

RMSDES bases its area calculations on the number of keys specified in the key section:

- If you specify 1 key (the primary key), RMSDES will define 3 areas:

  Area 0 -- data level for primary key

  Area 1 -- level-1 index

  Area 2 -- level-2 index

- If you specify 2 keys, RMSDES defines areas 0, 1, and 2, as above, plus 2 additional areas:

  Area 3 -- all index levels

  Area 4 -- data level for first alternate key

- If you specify 3 or more keys, RMSDES defines areas 0 through 4, as above, plus 1 new data-level area for each additional key specified. For example, if you specify 4 keys, RMSDES will define areas 0 through 4, as above, plus 2 new data-level areas for keys 3 and 4 (or 7 areas total). All index levels will be in area 3.

Note that the data area for alternate keys contains a set of pointers from the alternate index to the user data records.

In defining areas, RMSDES automatically calculates default allocation, bucket size, and extension values for each area.

RMSDES calculates allocation using the maximum record number (see Section 2.6.2) and the record size (see Section 2.6.3). If you are allowing RMSDES to define areas by default, you must set these attributes in the file and record sections of the design buffer. Otherwise, RMSDES will calculate an allocation that may not be appropriate for your file.

RMSDES allocation calculations also take into account whether duplicate key values are allowed, based on your response to the prompt for the average number of duplicates (see Section 2.6.5).

You will be prompted for bucket size when you allow RMSDES to define areas by default (see Section 2.6.5). You can respond with an explicit bucket size for RMSDES to use in defining areas, or you can take the default. By default, RMSDES sets bucket size to 2 blocks, or to a size great enough to contain 1 record if record size is greater than 2 blocks.

RMSDES calculates the default extension as 5 percent of the area calculation, rounded to a multiple of bucket size.

APPENDIX C

RMS-11 COMMAND UTILITIES MESSAGES

Section C.1 of this appendix documents error messages issued by the RMSBCK, RMSRST, RMSCNV, and RMSDSP command utilities. RMSIFL command utility messages are documented in Section C.2. Messages issued by the interactive utilities RMSDES and RMSDEF are documented in Chapter 2 and Appendix A, respectively.

The following conventions are observed in the messages documented in this appendix.

Convention	Meaning
?	A question mark, appearing as the first character of a message, indicates a fatal error condition. When a fatal error condition occurs, the utility terminates processing at the point at which the error was encountered.
%	A percent sign, appearing as the first character of a message, indicates a nonfatal error, warning, or an informational message. The utility terminates processing of only the file causing the error. The utility continues processing any other files specified in the command string.
s	A character representing the severity level for the error. A severity code of F is fatal, a severity code of I is informational, and a severity code of W is a warning.
utl	The 3-character name of the utility encountering the error.
<date>	A date you specified within the command string.
<device>	A device specification.
<file-spec>	A file specification.
<fipcode>	A numeric value representing a file processor error code. When an error message contains a fipcode, refer to the description of the code in the error code appendix of the IAS/RSX-11 I/O Operations Reference Manual.
<nn>	A numeric value.
<rmscode>	A numeric value representing an RMS-11 error code. When an error message contains a code, refer to the description of such codes in the RMS-11 Macro Programmer's Guide.

<status-val> A status value representing either a fipcode or detailed information about an operation (for example, a status subcode).

<string> A command or fragment of a command.

<switch> A switch appearing within the command string.

<value> A switch value appearing within the command string.

<vbn> A virtual block number.

C.1 RMSCNV, RMSDSP, RMSBCK, AND RMSRST ERROR MESSAGES

This section lists all RMSCNV, RMSDSP, RMSBCK, and RMSRST error messages. An explanation and the appropriate user action is indicated for each message.

?utl-s-ANA, Argument not allowed - <switch>

Explanation: The specified switch does not accept arguments.

User Action: Refer to the appropriate chapter and verify the usage for the switch you specified. Retype the command line.

?utl-s-APR, File <file-spec> is not empty, /AP switch is required

Explanation: The output file exists, the file was not empty, and the /SU switch was not specified.

User Action: To supersede, specify the /SU switch. To append, specify the /AP switch.

?utl-s-BFS, Too many selective file specifications

Explanation: More than 10 file specifications appear as values of the /SE switch.

User Action: Type the command string specifying no more than 10 file specifications as values of the /SE switch. Use multiple command strings to select additional files.

?utl-s-CDE, Unable to restore special attributes for file <file-spec>

Explanation: When RMSRST created the output file, the write attributes QIO failed. The utility cannot restore the file with one or more of the following attributes from the original file:

- One or more of its original data attributes
- Its original protection specification
- Contiguity
- Placement control
- Areas

User Action: Use the RMSDSP utility to determine which attributes of the file were not restored.

?utl-s-COR, Not enough memory available

Explanation: Not enough memory is available in the image to process the files specified in the command string. This could be caused by one or more of the following:

- Too many keys were specified in the output file.
- Too many keys were specified in the input file.
- The bucket size in either the input or output file was too large.

User Action: Use a smaller bucket size or fewer keys in the output file. If this is not possible, copy the input file to a sequential file using RMSCNV and use this sequential file as input to RMSIFL.

?utl-s-CRP, Output volume corrupt - <file-spec>

Explanation: The internal structure of an RMS-11 file is corrupted, and RMS-11 cannot read or write records in the file.

User Action: Move the disk pack to another device and try the command string again. If the utility processes the file normally, there was a read error on the first attempt; you should notify your system manager. However, if the utility prints this error message again, you must recreate the indicated file:

1. Use RMSDES to define and create a file, then use RMSIFL or RMSCNV to populate the new file from the corrupted one.
2. Use the last back-up copy of the file with the RMSRST utility.

?utl-s-CSE, Syntax error <string>

Explanation: The format of the command string does not conform to the syntax rules. The utility prints the command string from the point of the error to the end of the line.

User Action: Refer to the appropriate utility description in Chapters 4 through 7. Type the command string with the correct syntax.

?utl-s-CSW, Conflicting switch specified - <switch>

Explanation: The indicated switch contradicts a previous switch within the same command string (scanning from left to right).

User Action: Type the command string with the correct switch.

?utl-s-DOL, Device off line - <device>

Explanation: The indicated device is on the system, but access is prohibited for one of the following reasons:

- The device is not ready.
- No volume is mounted on the device.
- The device requires privileges for use and you do not have those privileges.
- The device is disabled.
- The device on which the file is contained is off line.

User Action: Correct the problem and retype the command string.

?utl-s-DWP, Device write protected - <device>

Explanation: The utility cannot access the device for write operations for one of the following reasons:

- The device is characteristically read-only.
- Read-only access (by file protection code or hardware/software write locking) is allowed.
- On a write operation, the error IE.WER or IE.WLK may have been returned.

User Action: Write-enable the device.

%utl-s-EOE, EOF (Control/Z) established at VBN <vbn>, byte <nn>

Explanation: When using the RMSCNV /EO switch, if the EOF character (CTRL/Z) has been found, the utility displays its location.

User Action: None.

?utl-s-ESA, Incomplete output file remains - <file-spec>

Explanation: An error has occurred which resulted in the suspension of a file backup (RMSBCK) or restore (RMSRST). This message appears after the appropriate error message has been displayed, and indicates that although the output file has been created, the contents are not complete.

User Action: If possible, correct the problem and try again.

%utl-s-EXC, Exception record: --reason code-- <notes>

Explanation: An exception record has been produced by RMSCNV. This message will appear only if you specified the /ER switch, and your target output file is a record unit (terminal or printer) device. One of the following reason codes is listed:

- Record out of sequence
- Duplicate key where not allowed
- Record too short for primary key
- Record too long for bucket size
- Record doesn't fit fixed length
- Record too long for maximum record size
- Record contains an illegal data type

In addition, one of the following notes may also be listed:

- Truncated - indicating that the the key was too large to display
- Nonprinting - indicating that the key contained nonprinting characters, which RMSCNV changed to a "?"

User Action: You may fix the exception records and re-run RMSCNV to load them.

?utl-s-EXF, Multiple output files not allowed

Explanation: Multiple output files are not allowed for this utility.

User Action: Break the command line into several commands each having single output files.

?utl-s-FNA, File <file-spec> not available

Explanation: Either the file could not be shared (for RMSDSP), or write access could not be obtained to process the file.

User Action: Retry the command string until the file is released.

%utl-s-FPC, Processing complete for file <file-spec>

Explanation: The utility displayed an informational message indicating that the backup (RMSBCK) or restore (RMSRST) for a file was completed successfully.

User Action: None.

?utl-s-FPL, File position lost - <file-spec>

Explanation: An internal inconsistency was detected in the back-up set, and processing cannot be continued. Possible reasons include:

- On /RC (read check), the internal EOF was not processed properly.
- A rewritten VBN is out of place within the back-up file.
- A preamble (internal header) check failed, indicating that internal structures are in error.

User Action: Determine from the output account or summary listing file the extent of the processing that was completed before the error occurred. Type the command string, eliminating specifications of files successfully processed. Use a new tape volume and/or a different tape drive.

?utl-s-FQL, Output file fill quantities will not be preserved

Explanation: When using the RMSCNV /CA (copy attributes) switch with multiarea indexed files that are not empty, the fill quantities will not be copied to the new output file.

User Action: To preserve multiareas, use RMSDES to create them. Use RMSCNV or RMSIFL to populate the indexed file.

?utl-s-FTL, Fatal RMS error; PC=<nn>, STS=<rmscode>, STV=<status-val>, file=<file-spec>

Explanation: An unexpected fatal error return occurred on an RMS-11 operation. The error returns are octal values.

For RMSCNV and RMSDES, this indicates that the internal error processor cannot identify the error code. This may indicate a mismatch between the version of the utility and the RMS-11 access methods.

User Action: See the RMS-11 Macro Programmer's Guide for descriptions of the STS and STV codes.

?utl-s-FTO, Illegal file specification - <file-spec>

Explanation: The file specification does not conform to the syntax rules or the expanded string cannot fit within the internal buffer allocated.

User Action: Type the command string with the correct syntax.

?utl-s-IAC, File <file-spec> is locked

Explanation: The specified file was improperly closed the last time it was accessed and remains in a locked state.

User Action: Use the /UN switch or the /EOF switch (if appropriate) with PIP to unlock the file (see the RSX-11M/M-PLUS Utilities Manual).

?utl-s-IAS, Illegal use of /AP, /BL, or /SU with file organization

Explanation: You combined one of the switches illogically with a file organization, as follows:

- /AP or /SU with a nonsequential file
- /BL with a disk file

User Action: Type the command string without the switch.

?utl-s-IAX, Illegal use of /KN, /KR, or /LO with file organization

Explanation: The RMSCNV /KN, /KR, and /LO switches require output indexed files. You specified an output file that is not indexed.

User Action: You cannot use any of the switches (/KN, /KR, or /LO) unless the output file is indexed.

%utl-s-ID, Version <nn>

Explanation: This message displays the version number and patch level of the utility.

User Action: None.

?utl-s-IFL, Incorrect file organization for file <file-spec>

Explanation: The RMSCNV /FO switch argument value does not match the real organization for the output file.

User Action: Type the command string with the correct file organization.

?utl-s-IKN, Missing or illegal key name argument

Explanation: There is an error in the key name argument to the RMSCNV /KN switch. You specified either a key that is null, or a key that contains illegal characters.

User Action: Specify RMSDSP using the /FU switch to determine what the key names are.

?utl-s-IKR, Invalid /KN or /KR value

Explanation: The key specified as the argument to the RMSCNV /KN or /KR switch is not defined for this input indexed file.

User Action: Enclose the name in quotation marks if the name is lower case or contains spaces.

?utl-s-IKY, Input key of reference is <nn> <key-name>

Explanation: The utility displays this message to identify the key of reference for input indexed files.

User Action: None.

?utl-s-ILF, Error insert too big, code= <rmscode>

Explanation: This message indicates that either a command line from an indirect command file is too long, or that the parameters passed to the error processor exceed the provided buffer space. The code field contains the octal error code.

User Action: Notify your system manager who should submit a Software Performance Report (SPR) documenting the conditions under which the error occurred.

?utl-s-ILS, Illegal switch on input file - <switch>

Explanation: The indicated switch is not a valid input file switch for the utility.

User Action: Type the command string with the correct switch or without the indicated switch.

?utl-s-INP, Multiple input files not allowed

Explanation: You specified multiple input files for a utility that accepts only a single input file.

User Action: Type the command string specifying a single input file.

%utl-s-IOI, I/O error encountered on input file <file-spec>
VBN <vbn> to <vbn>. Continue (Yes or No)?

Explanation: The virtual blocks cannot be read from the file. If the file is input to RMSBCK, the utility cannot back up the data records within the blocks. If the file is input to RMSRST, the utility cannot restore the data within the blocks.

User Action: Type "Y" if you want the utility to continue despite the error. Type "N" if you want the utility to terminate.

?utl-s-IOO, I/O error encountered on output file - <file-spec>

Explanation: One of the following conditions exists:

- The device is not on line.
- The device is not mounted.
- The hardware failed.
- The volume is full.

User Action: Correct the problem and retype the command string.

?utl-s-IOR, I/O error on command line input

Explanation: The utility cannot read:

- The command string you typed at the terminal
- A line within an indirect file

User Action: Retype the command string or recreate the indirect file and retry the command.

?utl-s-IOS, Illegal output option - <switch>

Explanation: The switch is not a valid output file switch.

User Action: Type the command string with the correct switch or without the indicated switch.

?utl-s-ISP, Illegal argument value - <value>

Explanation: The value specified in the switch is either null, the incorrect data type, or inappropriate for the specific file.

User Action: Type the command string with the correct switch value.

?utl-s-ISS, Illegal switch specified - <switch>

Explanation: You specified a switch that the utility does not recognize.

User Action: Retype the command string correctly.

?utl-s-IWC, Illegal use of wildcards on input file

Explanation: The utility does not permit wildcard characters (asterisks) in an input file specification.

User Action: Type the command string without wildcard characters in the input file specification.

?utl-s-IWF, Illegal use of /WF with record format

Explanation: You specified the /WF switch in an RMSCNV command string, but neither the input nor the output file contains variable-with-fixed-control (VFC) records.

User Action: Type the command string without the /WF switch.

?utl-s-LSW, Illogical switch in context of command - <switch>

Explanation: Either the switch appears multiple times on the command line, or the specified switch does not have any meaning with the files being processed.

User Action: Type the command string with appropriate switches.

?utl-s-LUN, TI: LUN assignment error

Explanation: An attempt to assign the terminal LUN for command line and error text processing has failed.

User Action: Notify your system manager who should submit a Software Performance Report (SPR) documenting the conditions under which the error occurred.

?utl-s-MDE, Command file nesting depth exceeded

Explanation: An indirect command file contains an indirect command. RMS-11 does not permit nesting of indirect commands.

User Action: Type the contents of one or both of the indirect command files as a sequence of command strings.

?utl-s-MLE, Block value of /ML switch exceeds limit or available memory

Explanation: The RMSCNV /ML (memory limitations) switch specifies a numeric argument which is not in the allowable limit (0-63 for image copy), or specifies allocation of more memory than is available to the task image.

User Action: Reenter the command line specifying a smaller value for the /ML switch.

?utl-s-MLF, Read error or inconsistent data. May have lost files.

Explanation: The utility encountered an error while reading a back-up or container file.

User Action: Retry the command string, specifying the /RC switch. If the error recurs, determine, from the summary listing produced when the back-up or container file was created, which files cannot be completely restored. If you have other back-up copies of these files, use them with the command.

?utl-s-MMQ, Mismatched quotes <string>

Explanation: An odd number of quotation characters is present on the command line. Quotation characters must be used in pairs.

User Action: Determine where the unpaired quote is in the command line and either remove it or place the second quote in the appropriate place. Reenter the command line.

?utl-s-MOV, Missing option value - <switch>

Explanation: You did not supply a value with the indicated switch.

User Action: Type the command string specifying a value for the indicated switch.

?utl-s-MRS, Input and output record sizes do not correspond

Explanation: The input and output record sizes do not correspond. Typically, this is due to the output file being fixed format, and the input MRS (fixed or variable size) not matching the output MRS (fixed size).

User Action: You may need to specify the /PD and/or /TR switches. This will allow RMSCNV to pad short records and/or truncate long records to conform to the output MRS.

?utl-s-MSF, Output file specification required

Explanation: You did not provide an output file specification on the command line, when one was expected and required.

User Action: Supply an output file specification that output records can be written to. The device is utility-specific (see the appropriate chapter) and may be restricted to a certain category.

?utl-s-NBK, Input file <file-spec> is not container file

Explanation: The utility requires the input file to be a back-up file. You have specified a file that is not in back-up format.

User Action: Type the command string with the correct file specification.

?utl-s-NDM, Explicit container filename necessary

Explanation: The output file must be an explicit file specification. No wildcard file name and type fields are allowed.

User Action: Retype the command line providing an explicit file specification.

?utl-s-NRN, No rename allowed

Explanation: During backup (RMSBCK) or restore (RMSRST) of disk-based back-up files, the file specifications of the output files were not wildcarded. RMSBCK and RMSRST output file name and type fields must be wildcarded if the medium is disk.

User Action: Type the command string specifying wildcards for the name and extension components of the output file.

?utl-s-NSD, Not a shareable device - <device>

Explanation: The command string you typed requires the sharing of a nonshareable device.

User Action: Type the command string with a correct device specification.

?utl-s-OOA, Output file will contain only one area

Explanation: The file used as input to the RMSCNV /CA switch (or the input file if no argument was specified to the /CA switch) contained multiple areas.

User Action: If the output indexed file is to contain multiple areas, use RMSDES to define and create the file. Then, use RMSCNV or RMSIFL to populate it.

?utl-s-OPR, Failure to open command file

Explanation: An I/O or file system error, such as a privilege violation, an attempted write operation to a locked unit, or a parity error occurred when the utility attempted to open an indirect command file.

User Action: Correct the problem and retry the command string.

?utl-s-OWC, Illegal use of wildcards on output file

Explanation: The utility does not permit wildcard characters (asterisks) in the output file specification.

User Action: Type the command string without wildcard characters in the output file specification.

?utl-s-PRV, Privilege violation on file <file-spec>

Explanation: You do not have the privileges necessary to access the indicated file.

User Action: Ask the owner of the file to change the protection code, or use the utility from a more privileged account.

?utl-s-RAE, Error reading attributes for file <file-spec>

Explanation: This message results from one of sveral possible errors:

- An inconsistency in the preamble of a back-up file
- An error attempting to read the attributes of a back-up file
- An error reading the preamble of a new back-up file on the RMSBCK /RC (read check) pass
- A mismatch between the copied data and the original file on the RMSBCK /RC (read check) pass

User Action: Use RMSDSP to determine the file's protection:

- If you are forbidden access to the file by its protection code, request that its protection be changed or use a more privileged account.
- If you are allowed access to the file by its protection code, use the VFY utility to check the volume for corruption.

?utl-s-RAM, Required argument missing - <switch>

Explanation: You did not provide an argument value for the switch you specified.

User Action: Retype the command line and provide an argument value for the switch.

%utl-s-RCE, Read after write error on output file <file-spec>, VBN <vbn> to <vbn>. Continue (Yes or No)?

Explanation: During data integrity checking, the utility encountered a read error while attempting to read virtual blocks vbn1 through vbn2 of the indicated file.

User Action: Type "Y" if you want the utility to stop processing the indicated file, but process the rest of the command string. Type "N" if you want the utility to stop processing the command string.

?utl-s-RCR, Release core error

Explanation: The utility's internal dynamic memory has been corrupted.

User Action: The utility finished processing the current command string. If you want to continue using the utility, you should terminate it with CTRL/Z (if necessary) and then reinvoke it.

%utl-s-REC, <string>

Explanation: This message is displayed after the EXC message and when the RMSCNV /ER switch is selected and the exception record device is unit oriented (terminal or printer). The message displays the contents of the key section of the record that caused the error. If the minimum record size (based upon the key positions and sizes) is larger than 68 characters for terminals or 120 characters for line printers, the EXC message will indicate that the record fragment was truncated. If nonprinting characters are present, the EXC message will also indicate this, and the nonprinting characters will be replaced by "?" characters.

User Action: None.

?utl-s-RET, Read error, integrity check table and rewrite data may have been lost

Explanation: The RMSBCK or RMSRST utility was unable to read the internal rewrite table of the current back-up file. This can occur on a RMSBCK verification of the output file, or a RMSRST read of the rewrite table.

User Action: Retry the command string, specifying the /RC switch. If the error recurs, determine, from the summary listing produced when the back-up or container file was created, which files cannot be completely restored. If you have other back-up copies of these files, use them with the command.

?utl-s-RLG, Error in prologue of file <file-spec>

Explanation: The utility detected an error in the prologue of the relative or indexed file.

User Action: This file may be corrupt; the prologue may contain invalid data. This implies that the file must be recreated.

?utl-s-RPL, Error reading prologue of file <file-spec>

Explanation: An error occurred when RMS-11 attempted to read and verify the prologue of a relative or indexed file.

User Action: Determine if the hardware is at fault (for example, if there are read errors). If the hardware is not at fault, the prologue may be corrupt and the file may have to be recreated.

?utl-s-RSM, Required file specification(s) missing

Explanation: You entered a command string with either an input or output file specification missing.

User Action: Retype the command string with both input and output file specifications.

?utl-s-RWC, Error during wildcard lookup <file-spec>. Code=<fipcode>

Explanation: A wildcarding directive failed. The code field contains the octal error code returned by the executive.

User Action: Determine if the device is at fault (hardware errors), or verify that the directory is intact. Retry the command line.

?utl-s-RWE, Rewind or space error on file <file-spec>

Explanation: While processing a magnetic tape, either an illegal operation was attempted or the device returned a device positioning error.

User Action: Retry the command string. If the condition recurs, mount the volume on another drive and retry the command string.

?utl-s-SLT, Select error - <device>

Explanation: The magtape specified in the device field of the file specification is not accessible. It is probably off line.

User Action: Correct the problem and retype the command string.

?utl-s-SLW, Wildcards not permitted in summary listing file specification

Explanation: You used asterisks in a file specification as the argument of an /SL switch. RMS-11 does not permit wildcard characters in this context.

User Action: Type the command string without wildcard characters in the summary listing file specification.

?utl-s-SSO, Illegal use of /EO with organization or record format

Explanation: You attempted to use the /EO (end-of-file) switch on a file that was not sequential organization with stream record format. The file is not acceptable input for /EO processing.

User Action: None.

%utl-s-TER, Integrity check table full. Continue (Yes or No)?

Explanation: The table, allocated internally by the utility to monitor input read errors, exceeded its capacity. The file or volume can be assumed to be either corrupt, or produce too many errors for the back-up data to be considered intact. This suggests hardware faults.

User Action: Type "Y" to continue processing. Type "N" to terminate processing. Also, check the system error log, or clean the (tape) device or unit.

?utl-s-TMP, Error processing temporary file. Code= <fipcode>

Explanation: The wildcard processor called by the utility encountered an error while creating a temporary file for resolution of wildcards in a file specification. The code field contains the octal error code.

User Action: Type successive command strings to achieve the result wildcards would produce.

?utl-s-TNF, No EOF (Control/Z) found in file

Explanation: The RMSCNV utility could not find the CTRL/Z character in the sequential stream file specified with the /EO switch. The data within the file remains unaltered.

User Action: None.

?utl-s-TOO, Command line too long

Explanation: The command string you typed is longer than the utility's command string buffer (158 characters). Therefore, you have typed too many continuation lines.

User Action: Use several command strings to perform the functions of the original command string. If the condition occurs repeatedly, notify your system manager who should submit a Software Performance Report (SPR).

?utl-s-TPM, Switch /TR or /PD or both are needed for this CONVERT

Explanation: The record sizes of the files being processed by RMSCNV are not compatible.

User Action: Refer to Chapter 4 for the use of the /TR and /PD switches. Then, retype the command string using one or both of the switches.

?utl-s-UER, Uncorrectable error, VBN <vbn> to <vbn>, reason= <reason>

Explanation: Uncorrectable errors have been detected by the RMSBCK or RMSRST utility. This message indicates the cause of the problem, and the range of virtual blocks involved. The message is one of the following:

Read check error

Write check error

Input error

Output uncheckable

User Action: Record the ranges of erroneous virtual blocks in the file and notify your system manager.

?utl-s-UIC, Destination UICs may only be [*,*] or [X,Y]

Explanation: You typed an output file specification with an invalid account number.

User Action: When specifying wildcarded UICs, both the group and member fields must be wildcards. Either both group and member fields are wildcarded, or neither one can be.

?utl-s-UIF, CNV internal fault

Explanation: An internal error was detected by the RMSCNV utility.

User Action: Notify your system manager who should submit a Software Performance Report (SPR) documenting the conditions under which the error occurred.

?utl-s-UTL, Utility input error

Explanation: An internal initialization error occurred during the execution of the indicated utility.

User Action: Retype the command string. If the condition recurs, notify your system manager who should submit a Software Performance Report (SPR) documenting the conditions under which the error occurred.

?utl-s-VFC, Input and output fixed control header sizes do not correspond

Explanation: You attempted to write records from one file to another. Both files contain variable-with-fixed-control (VFC) records, but the sizes of the fixed areas are different. This is not allowed when using the /WF (write fixed headers) switch.

User Action: Redefine the output file and retype the command string.

?utl-s-WAE, Error writing attributes of file <file-spec>

Explanation: The volume is corrupted or you do not have the necessary privileges to write to the file.

User Action: Verify access to file.

?utl-s-WCD, Error with wildcards

Explanation: The wildcard processor returned an error to the utility during resolution of wildcards in a file specification.

User Action: Type the command string again. If the condition recurs, use successive command strings to achieve the result wildcards would produce.

%utl-s-WCE, Check after write error on output file <file-spec>, VBN <vbn> to <vbn>. Continue (Yes or No)?

Explanation: This diagnostic error message is displayed on your terminal if query mode is enabled and the /RC switch (check after writing) is specified. During data integrity checking, the utility found that the contents of virtual blocks vbn1 through vbn2 in the indicated output file are not identical to the associated virtual blocks of the input file.

User Action: Type "Y" to continue processing despite the error. Type "N" to terminate processing of the indicated output file and bypass the processing of any remaining files.

?utl-s-WET, Write error on integrity check table on output file <file-spec>

Explanation: The utility cannot write internal data integrity checking tables in the output back-up file.

User Action: If the output medium is magnetic tape, use a different tape volume and retry the command string. If the output medium is disk, rename the output file so that the utility will not attempt to use the space, and retry the command string.

C.2 RMSIFL ERROR MESSAGES

This section lists all RMSIFL error messages. An explanation and the appropriate user action is indicated for each message.

?IFL -- Block I/O read error on <file-spec>

Explanation: RMSIFL could not read the index bucket from the output file.

User Action: Retype the command string. If the error recurs, check with your system manager to see if there have been other errors reported for that device. If not, move the file to a different disk or specify a different device for the file and retry the operation.

?IFL -- Block I/O write error on file <file-spec>

Explanation: RMSIFL could not write a block to the output file.

User Action: Retype the command string. If the error recurs, check with your system manager to see if there have been other errors reported for that device. If not, move the file to a different disk or specify a different device for the file and retry the operation.

?IFL -- Close error on <file-spec>

Explanation: The specified input or output file could not be properly closed.

User Action: Retype the command string. If the error recurs, run a validity check of the file structure using the VFY utility on the volume to determine whether it is corrupted.

?IFL -- Command file nesting depth exceeded

Explanation: An indirect command file contains an indirect command. RMS-11 does not permit nesting of indirect commands.

User Action: Type the contents of one or both of the indirect command files as a sequence of command strings.

?IFL -- Create error on <file-spec>

Explanation: The specified file could not be created by RMSIFL. This condition can occur for one of the following reasons:

- The /ER switch was specified on the command line and RMSIFL could not open the Exception Record file.
- RMSIFL could not create the alternate key temporary file.
- RMSIFL could not create the temporary file into which to write the sorted records from SORT.

User Action:

- Check the file specification that you supplied with the /ER switch to be sure that it is a legal file specification.
- Check the available space on the disk and make sure that there is enough space to create the files. Refer to Chapter 3 for file size calculations.
- If the error occurred on a work file used in RMSIFL sorting, retype the command line specifying the /DE switch to assign the temporary work files to another device.

?IFL -- Error terminated input before expected, but continuing

Explanation: RMSIFL encountered a read error in the input file. RMSIFL will finish processing with as many records as it was able to read from the input file.

User Action: The input file cannot be read. Retry the operation using the last back-up version of the file.

?IFL -- Error with logical unit device

Explanation: RMSIFL assigned a logical channel erroneously.

User Action: Invoke RMSIFL again and retype the command line. If this message recurs, submit a Software Performance Report (SPR).

?IFL -- Extend error on output file <file-spec>

Explanation: The output file you provided was not large enough to contain the input records, or a temporary file could not be extended. RMSIFL explicitly extended the file, but the operation failed.

User Action: Determine why the file could not be extended (such as, disk full). Allocate a larger output file or use a different output disk, and type the command string again.

?IFL -- Failure to open command file

Explanation: An I/O or file system error, such as a privilege violation, an attempted write operation to a locked unit, or a parity error occurred when RMSIFL attempted to open an indirect command file.

User Action: Correct the problem and retry the command string.

?IFL -- Fatal RMS error - STS = <rmscode>, STV = <rmscode>

Explanation: A fatal error occurred in the RMSIFL interface with RMS-11 file and record handling routines.

User Action: See the RMS-11 Macro Programmer's Guide for descriptions of the STS and STV codes.

?IFL -- File access error on <file-spec>

Explanation: The specified file could not be accessed.

User Action: Check to be sure that the file specifications that you supplied are legal file specifications and that the input and output files exist.

?IFL -- Illegal device <device>

Explanation: The device does not exist.

User Action: Type the command string with a correct device specification.

?IFL -- Input and output fixed length record sizes unequal

Explanation: The input and output files both have fixed-length records, but the sizes are different. One of the following conditions exists:

- The output file's maximum record size is less than the input file's maximum record size, and the /TR switch was not specified on the command line.
- The output file's maximum record size is greater than the input file's maximum record size, and the /PD switch was not specified on the command line.

User Action: You must specify either the /PD or /TR switch in the command string.

?IFL -- Input file is not sequential, relative, or indexed

Explanation: RMSIFL requires an input file of sequential, relative, or indexed file organization.

User Action: Retype the command string specifying a valid file.

?IFL -- Input file switch specified with output file switch

Explanation: The switch you specified on the output side of the command line is only valid as an input file switch.

User Action: Retype the command string correctly.

?IFL -- Invalid /KR value

Explanation: You specified a key of reference greater than or equal to the number of keys in the input file.

User Action: Retype the command line specifying a valid key of reference number.

?IFL -- I/O error on command line input

Explanation: RMSIFL cannot read:

- The command string you typed at the terminal
- A line within an indirect file

User Action: Retype the command string or recreate the indirect file and retry the command.

?IFL -- /KR not allowed for sequential or relative file

Explanation: The /KR switch is permitted for indexed files only. You either specified the wrong file or used the /KR switch in the wrong context.

User Action: Type the command string with the correct file specification or without the /KR switch.

?IFL -- Memory exhausted

Explanation: Not enough memory is available in the image to process the files specified in the command string.

User Action: Consult your system manager for details on increasing the available memory for RMSIFL.

?IFL -- Missing file specification

Explanation: You entered a command string with either an input or output file specification missing.

User Action: Retype the command string with both input and output file specifications.

?IFL -- Output file fixed record size too large for bucket size

Explanation: You specified a fixed-record size in an indexed output file that is larger than the bucket size.

User Action: Create a new output file with either a larger bucket size or a smaller record size.

?IFL -- Output file must be empty

Explanation: The output file you specified contains data.

User Action: Recreate the output file.

?IFL -- Output file must be an indexed file

Explanation: You specified a sequential or relative file as an output file.

User Action: Specify or create another indexed output file.

?IFL -- Output file switch specified with input file switch

Explanation: The switch you specified on the input side of the command line is only valid as an output file switch.

User Action: Retype the command string correctly.

?IFL -- Prologue version number too high

Explanation: RMSIFL opened an RMS-11 indexed file and checked the version number of the file's prologue. That number indicates that the file was created with attributes that RMS-11 cannot handle.

User Action: Recreate the file using a newer version of RMSIFL or RMSCNV or write an application program using the latest RMS-11 software.

?IFL -- Read error on filespec

Explanation: RMSIFL could not read the sorted records.

User Action: Retry the command string. If the error occurred on an input file and the error persists, check with your system manager to see if there have been other errors reported for that disk. If there have not been any errors, then move the disk to another device and retry the command string. If RMSIFL processes the file normally, a read error occurred in the first two attempts. However, if the error occurs again, you must recreate the indicated file as follows:

1. Use RMSDES to define and create a file, then use RMSIFL or RMSCNV to populate the new file from the corrupted one.

2. Use the last back-up copy of the file with the RMSRST utility.

If the error occurred on a work file used in RMSIFL sorting, use the /DE switch to assign the work file to another device.

?IFL -- Read error on file prologue <file-spec>

Explanation: RMSIFL cannot read the prologue of the indicated file. RMSIFL bypasses the file and continues processing.

User Action: Type the command string specifying only the indicated file. If the error recurs, move the disk pack to another device and try the command string again. If RMSIFL processes the file normally, a read error occurred on the first two attempts. However, if RMSIFL prints this error message again, you must recreate the indicated file as follows:

1. Use RMSDES to define and create a file, then use RMSIFL or RMSCNV to populate the new file from the corrupted one.

2. Use the last back-up copy of the file with the RMSRST utility.

?IFL -- SORTS error code in octal: <nn>

Explanation: The sort routines in RMSIFL encountered an error; nn is a sort error code.

User Action: Refer to Appendix D for a description of the indicated error code. Correct the problem and retype the command string.

?IFL -- Syntax error

Explanation: The format of the command string does not conform to the syntax rules.

User Action: Refer to the appropriate RMSIFL description in Chapter 3. Type the command string with the correct syntax.

?IFL -- /TR and/or /PD required with fixed length output records

Explanation: The record size for the input file is different from the record size for the output file and may require truncating (/TR) and/or padding (/PD).

User Action: Retype the command string specifying /PD and/or /TR.

?IFL -- Undefined GCML error

Explanation: An unknown error occurred during the processing of the command line.

User Action: Retype the command line. If the error recurs, submit a Software Performance Report (SPR).

If the error occurred while RMSIFL was processing an indirect command file, try the commands interactively.

?IFL -- Unknown switch <switch>

Explanation: You specified a switch that RMSIFL does not recognize.

User Action: Retype the command string correctly.

?IFL -- Write error on filespec

Explanation: RMSIFL could not write the sorted records to the file.

User Action: Retype the command string. If the error persists, use the /DE switch and assign the file to a different device.

APPENDIX D

SORT ROUTINE ERROR CODES

When the sort routines used by RMSIFL detect an error, they return one of the nonzero, octal codes listed below.

Error Code (Octal)	Meaning
00	No errors
01	Device input error
02	Device output error
03	OPEN (IN) failure
04	OPEN (OUT) failure
05	Size of current record is greater than maximum size
06	Not enough work area
12	Key size is not positive (the sort routines detected a zero or negative key size in the calling parameter)
13	Record size not positive
14	Key address is not even (the keys must start at an even address because the sort routines use word moves)
15	Record address is not even
16	Scratch records will be too large (the size of the keys plus the size of the largest record must be less then 37776 octal)
17	Too few scratch files are given (a minimum of 3 scratch files must be specified)
20	Too many scratch files are given (a maximum of 10 scratch files may be specified)
21	End-of-string record was detected where none was expected
22	Unexpected end-of-file
23	Record larger than expected

INDEX

/[NO]CV switch
  RMSRST utility, 7-5, 7-10
/[NO]QU switch
  RMSBCK utility, 6-5 to 6-7, 6-10
  RMSRST utility, 7-6 to 7-8, 7-12
/AP switch
  RMSCNV utility, 4-5, 4-7, 4-13, 4-15
/BD switch
  RMSRST utility, 7-8, 7-10
/BL switch
  RMSCNV utility, 4-8
/BP switch
  RMSDSP utility, 5-3 to 5-4
/CA switch
  RMSCNV utility, 4-8
/CD switch
  RMSBCK utility, 6-8 to 6-9
/DE switch
  RMSIFL utility, 3-7, 3-10, 3-14
/EO switch
  RMSCNV utility, 4-8
/ER switch
  RMSIFL utility, 3-5 to 3-6
/FO switch
  RMSCNV utility, 4-9, 4-15
/FR switch
  RMSRST utility, 7-7
/FU switch
  RMSDSP utility, 5-2, 5-4
/ID switch
  RMSBCK utility, 6-4 to 6-5
  RMSCNV utility, 4-5, 4-9
  RMSDEF utility, A-5
  RMSDSP utility, 5-3, 5-5
  RMSIFL utility, 3-3, 3-5
  RMSRST utility, 7-3, 7-5
/IM switch
  RMSCNV utility, 4-9
/KN switch
  RMSCNV utility, 4-10, 4-16
/KR switch
  RMSCNV utility, 4-10, 4-16
  RMSIFL utility, 3-8 to 3-9
/LO switch
  RMSCNV utility, 4-10
  RMSIFL utility, 3-6
/MA switch
  RMSCNV utility, 4-10
/ML switch
  RMSCNV utility, 4-11
/NOER switch
  RMSIFL utility, 3-6
/NOSO switch
  RMSIFL utility, 3-8 to 3-10
/OA switch
  RMSRST utility, 7-9 to 7-10
/PD switch
  RMSCNV utility, 4-12
  RMSIFL utility, 3-6 to 3-7, 3-11
/RA switch
  RMSBCK utility, 6-6 to 6-7, 6-9
  RMSRST utility, 7-7, 7-11
/RC switch
  RMSBCK utility, 6-7, 6-9
  RMSRST utility, 7-8, 7-11
/RD switch
  RMSBCK utility, 6-8 to 6-9
/RW switch
  RMSBCK utility, 6-7
/SE switch
  RMSRST utility, 7-3, 7-9
/SL switch
  RMSBCK utility, 6-6, 6-10
  RMSCNV utility, 4-12
  RMSRST utility, 7-6, 7-12
/SU switch
  RMSBCK utility, 6-7
  RMSCNV utility, 4-5, 4-7, 4-13, 4-15
  RMSRST utility, 7-8
/TR switch
  RMSCNV utility, 4-13
  RMSIFL utility, 3-7, 3-11
/WF switch
  RMSCNV utility, 4-14

Alignment
  RMSDEF utility, A-14
ALLOCATION
  RMSDES utility
    area section, 2-44
    file section, 2-31
Allocation
  RMSDEF utility
    data, A-16
    file, A-14
  RMSDES utility
    areas, 2-10, 2-25
  RMSDSP
    displaying, 5-4
Alternate index (RMSIFL), 3-11 to 3-12
Alternate key
  RMSDEF utility
    defining, A-11
  RMSIFL utility
    allocating file space, 3-13
    sorting, 3-11
Alternate key files (RMSIFL)
  device allocation, 3-7
ANSI-standard tape
  RMSBCK utility, 6-1, 6-8

RMSDSP utility, 5-3
Area information, displaying, 5-4
Area numbers (RMSDEF), A-16
Areas
  RMSDEF utility
    defining, A-13
  RMSDES utility, 2-10, 2-25
    defining, 2-1, 2-42
      and BUCKET_SIZE attribute, 2-32
      and DATA_AREA attribute, 2-40
      and INDEX_AREA attribute, 2-41
      and LEVEL1_INDEX_AREA attribute, 2-41
      and SIZE attribute, 2-36
      by default, 2-43
      explicitly, 2-43
    defining by default, 2-17, 2-38
      CREATE command, 2-8, 2-28
      SET ALL command, 2-14
    defining explicitly, 2-17
    numbering, 2-8, 2-28, 2-43
ASCII stream files
  RMSCNV utility, 4-3, 4-8
  RMSIFL utility, 3-1
Attributes
  RMSDES utility, 2-29
    ALLOCATION
      area section, 2-44
      file section, 2-31
    BLOCK_SPAN, 2-37
    BUCKET_SIZE
      area section, 2-43, 2-45
      file section, 2-32
    CARRIAGE_CONTROL, 2-37
    CHANGES, 2-40
    CLUSTER_SIZE, 2-31
    CONTIGUOUS
      area section, 2-46
      file section, 2-35
    CONTROL_FIELD_SIZE, 2-37
    DATA_AREA, 2-40
    DATA_FILL, 2-40
    DUPLICATES, 2-40, 2-42, 2-44
    EXACT_POSITIONING, 2-47
    EXTENSION
      area section, 2-45
      file section, 2-32
    FILE PLACEMENT
      system section, 2-30
    FORMAT, 2-36
    INDEX_AREA, 2-41
    INDEX_FILL, 2-41
    informational, 2-29
    LEVEL1_INDEX_AREA, 2-41
    MAGTAPE_BLOCK_SIZE, 2-34
    MAX_RECORD_NUMBER, 2-34
    NAME
      file section, 2-31
      key section, 2-38
    NULL_KEY, 2-39
    NULL_VALUE, 2-39
    ORGANIZATION, 2-31
    OWNER, 2-34
    POSITION, 2-46
    PROTECTION, 2-33
    REWIND_MAGTAPE, 2-34
    RMS-11, 2-29
    SEG0_POSITION, 2-42
    SEGn_LENGTH, 2-42
    SEGn_POSITION, 2-42
    SIZE, 2-36
    SOURCE, 2-30
    SUPERSEDE, 2-35
    TARGET, 2-30
    TYPE, 2-38
  RMSDSP, 5-1
Attributes (RMSDES)
  clearing, 2-6, 2-22, 2-25
    individual values, 2-26
    sections, 2-26
  displaying, 2-15, 2-26
    individual values, 2-26
    sections, 2-27
  informational, 2-1, 2-17
  resetting, 2-21
  RMS-11, 2-1, 2-17
  setting, 2-13
    from a data file, 2-25
    from a description file, 2-24
    individual values, 2-21
  setting sections, 2-21

Back-up date, file
  RMSRST utility, 7-8, 7-10
Block boundaries
  RMSDEF utility, A-9
Block mode
  RMSCNV utility, 4-1
BLOCK_SPAN
  RMSDES utility, 2-37
Bucket size
  RMSDEF utility, A-15
BUCKET_SIZE
  RMSDES utility
    area section, 2-43, 2-45
    file section, 2-32
Buckets
  RMSCNV utility, 4-10

Calculating file work space (RMSIFL), 3-12
Carriage control
  RMSDEF utility, A-9
CARRIAGE_CONTROL
  RMSDES utility, 2-37
CHANGES
  RMSDES utility, 2-40
CLEAR command
  RMSDES utility, 2-6, 2-18, 2-25 to 2-26

defining areas by default, 2-44
in indirect command file, 2-4
PROTECTION attribute, 2-33
with CREATE, 2-8, 2-28
with SAVE, 2-27
Clearing attributes (RMSDES)
See Attributes, clearing
CLUSTER_SIZE
RMSDES utility, 2-31
Command file
RMSDEF utility, A-5
Command string
RMSBCK utility, 6-2 to 6-3
RMSCNV utility, 4-3 to 4-4
RMSDEF utility, A-5
RMSDSP utility, 5-2 to 5-3
RMSIFL utility, 3-2 to 3-3
RMSRST utility, 7-2 to 7-3
Command string checking (RMSIFL), 3-9
Command utility
messages, C-1
Commands
RMSDES utility, 2-1, 2-4
See also individual command entries
See also RMSDES
Compression
RMSIFL utility, 3-11
Container file
RMSBCK utility, 6-1, 6-3, 6-7
RMSDSP
detailed display, 5-4
RMSDSP utility, 5-1, 5-3 to 5-4
RMSRST utility, 7-3, 7-9 to 7-10
Contiguity
RMSCNV utility, 4-2
RMSDEF utility, A-15
RMSIFL utility, 3-1, 3-6
CONTIGUOUS
RMSDES utility
area section, 2-46
file section, 2-35
CONTROL_FIELD_SIZE
RMSDES utility, 2-37
CREATE command
RMSDES utility, 2-7, 2-18, 2-28, 2-31
defining areas by default, 2-44
defining areas explicitly, 2-43
INDEX_AREA attribute, 2-41
key numbering, 2-38
LEVEL1_INDEX_AREA attribute, 2-41
setting attributes, 2-21
with GET, 2-10, 2-25
Creating files
RMSDEF utility, A-19
See also Data file (RMSDES)
Creation date, file
RMSBCK utility, 6-8
CTRL/Z
RMSBCK utility, 6-2
RMSCNV utility, 4-4
file terminator, 4-8
RMSDES utility, 2-2, 2-9, 2-18
RMSDSP utility, 5-2
RMSIFL utility, 3-2
RMSRST utility, 7-2

Data file
RMSDES utility, 2-3, 2-17, 2-24
creating, 2-3, 2-7, 2-28
GET command, 2-10
Data integrity checking
RMSBCK utility, 6-6 to 6-7, 6-9
RMSRST utility, 7-11
data integrity checking
RMSRST utility, 7-6, 7-8
DATA_AREA
RMSDES utility, 2-40
DATA_FILL
RMSDES utility, 2-40
Default extension quantity
RMSDEF utility, A-15
Defaults
RMSDES utility, 2-21
Description file
RMSDES utility, 2-3, 2-13, 2-17, 2-24
EXIT command, 2-10
GET command, 2-10
SAVE command, 2-27
Design buffer
RMSDES utility, 2-1
Design buffer (RMSDES)
using, 2-16
Design buffer sections (RMSDES)
See Sections
Design session
RMSDES utility, 2-1
resuming, 2-22
Display information (RMSDSP), 5-1
Displaying allocation information
RMSDSP utility, 5-4
Displaying area information
RMSDSP utility, 5-4
Displaying attributes (RMSDES)
See Attributes
Displaying indexed file attributes
RMSDSP utility, 5-4
Displaying key information
RMSDSP utility, 5-4
Displays (RMSDSP), 5-5
DUPLICATES
RMSDES utility, 2-40, 2-42
area section, 2-44

Error codes (RMSIFL)
  sort routine, D-1
ESC command
  RMSDES utility, 2-9, 2-21
EXACT_POSITIONING
  RMSDES utility, 2-47
Examples
  RMSBCK utility, 6-10
  RMSCNV utility, 4-16
  RMSDES utility, 2-47
  RMSIFL utility, 3-14
  RMSRST utility, 7-12
Exception codes
  RMSIFL utility, 3-6
Exception records
  RMSIFL utility, 3-5 to 3-6, 3-10
EXIT command
  RMSDES utility, 2-3, 2-9, 2-18, 2-27
EXTENSION
  RMSDES utility
    area section, 2-45
    file section, 2-32
External files (RMSDES)
  See Data file, Description file

File attributes (RMSDES)
  See Attributes
File attributes (RMSDSP)
  indexed file
    detailed display, 5-4
File back-up date
  See Back-up date
File contiguity
  See Contiguity
File creation (RMSDEF), A-19
File creation date
  See Creation date
File organization
  RMSDEF utility, A-8
FILE PLACEMENT
  RMSDES utility, 2-30
File revision date
  See Revision date
File space (RMSIFL)
  calculating, 3-12
File specification
  RMSBCK utility, 6-9
  RMSDEF utility, A-7
  RMSRST utility, 7-10
  See also Infile, Outfile
Fill numbers
  RMSCNV utility, 4-10
  RMSDEF utility, A-16
  RMSIFL utility, 3-6
Fixed control area
  RMSDEF utility, A-8
FORMAT
  RMSDES utility, 2-36

GET command
  RMSDES utility, 2-10, 2-17, 2-24
  PROTECTION attribute, 2-33
  with SAVE, 2-28
Global switches
  RMSBCK utility, 6-5
  RMSIFL utility, 3-5
  RMSRST utility, 7-5

HELP
  RMSBCK utility, 6-4
  RMSCNV utility, 4-5
  RMSDEF utility, A-5
  RMSDES utility, 2-2, 2-11
  RMSDSP utility, 5-3
  RMSIFL utility, 3-3
  RMSRST utility, 7-3

IFLA.TMP (RMSIFL), 3-11
  allocating space, 3-13
IFLA1.TMP (RMSIFL), 3-12
  allocating space, 3-14
IFLP.TMP (RMSIFL), 3-10
  allocating space, 3-13
INDEX_AREA
  RMSDES utility, 2-41
INDEX_FILL
  RMSDES utility, 2-41
Indirect command file
  RMSBCK utility, 6-2
  RMSCNV utility, 4-4
  RMSDES utility, 2-3
    comments, 2-4
  RMSDSP utility, 5-2
  RMSIFL utility, 3-2
  RMSRST utility, 7-2
Infile
  RMSBCK utility, 6-3
  RMSCNV utility, 4-5
  RMSDSP utility, 5-3
  RMSIFL utility, 3-3
  RMSRST utility, 7-3
Input file switches
  RMSBCK utility, 6-8
  RMSIFL utility, 3-7
  RMSRST utility, 7-8
Interactive utility
  RMSDES utility, 2-1
Intersystem file transfer, 1-2, 6-1
  RMSCNV utility, 4-1
  RMSRST
    /FR switch, 7-7
  RMSRST utility, 7-1, 7-10
    file-version number conversion, 7-5
Invoking
  RMSBCK utility, 6-2
  RMSCNV utility, 4-3
  RMSDEF utility, A-5
  RMSDES utility, 2-2
  RMSDSP utility, 5-2
  RMSIFL utility, 3-2
  RMSRST utility, 7-2

Key information
  displaying, 5-4
Key name
  RMSCNV utility, 4-10
Key of reference
  RMSCNV utility, 4-10, 4-16
  RMSIFL utility, 3-8 to 3-9
Keys
  RMSDES utility
    numbering, 2-28, 2-38
Keys (RMSDES)
  numbering, 2-8

LEVEL1_INDEX_AREA
  RMSDES utility, 2-41
Location
  RMSDEF utility, A-14

Magnetic tape
  RMSBCK utility, 6-7
    multivolume, 6-9
  RMSCNV utility, 4-8
  RMSDEF utility, A-8, A-10
  RMSDES utility, 2-34
  RMSRST utility, 7-8 to 7-9
MAGTAPE_BLOCK_SIZE
  RMSDES utility, 2-34
mass-insertion mode
  RMSCNV utility, 4-10
Maximum record number
  RMSDEF utility, A-9
Maximum record size
  RMSDEF utility, A-9
Messages
  command utilities, C-1
  RMSDEF utility, A-20
  RMSDES utility, 2-52
Mounting tapes (RMSDSP), 5-3
Multiple areas
  RMSDES utility, 2-7, 2-43
Multiple keys
  RMSDES utility, 2-7, 2-38

NAME
  RMSDES utility
    file section, 2-31
    key section, 2-38
Network facilities
  RMSCNV, 4-1
NULL_KEY
  RMSDES utility, 2-39
NULL_VALUE
  RMSDES utility, 2-39

ORGANIZATION
  RMSDES utility, 2-31
Outfile
  RMSBCK utility, 6-3
  RMSCNV utility, 4-4
  RMSDSP utility, 5-3
  RMSIFL utility, 3-3
  RMSRST utility, 7-3
Output file switches
  RMSBCK utility, 6-6
  RMSIFL utility, 3-5
  RMSRST utility, 7-7
OWNER
  RMSDES utility, 2-34

Padding characters
  RMSCNV utility, 4-12
  RMSIFL utility, 3-6
Patch level
  RMSBCK utility, 6-5
  RMSCNV utility, 4-9
  RMSDEF utility, A-5
  RMSDSP utility, 5-5
  RMSIFL utility, 3-5
  RMSRST utility, 7-5
Peripheral Interchange
  Program (PIP), 5-2
PIP/ENTER command (RMSBCK), 6-3
Placement
  RMSBCK utility, 6-1
  RMSDEF utility, A-14
  RMSRST utility, 7-1
POSITION
  RMSDES utility, 2-46
Primary index
  RMSCNV utility, 4-1
  RMSIFL utility, 3-10 to 3-11
Primary key
  RMSCNV utility, 4-1, 4-16
  RMSDEF utility
    definition, A-10
  RMSIFL utility, 3-5 to 3-6
    sorting, 3-9
Processing
  RMSBCK utility, 6-8
    data integrity checking, 6-9
    explicit/implicit file-spec, 6-9
    extended diagnostic messages, 6-10
    summary listing, 6-10
  RMSCNV utility, 4-14
    input file organization, 4-16
    output file organization, 4-15
  RMSDEF utility, A-1
  RMSIFL utility, 3-9
    building alternate indexes, 3-12
    command string checking, 3-9
    loading output file, 3-10
    reading and sorting input, 3-9
    sorting alternate key files, 3-11
    summary, 3-1
    terminating, 3-12
  RMSRST utility, 7-9
    data integrity checking, 7-11
    explicit/implicit

file-spec, 7-10
extended diagnostic messages, 7-11
restoration account, 7-10
summary listing, 7-12
Prompt
RMSBCK utility, 6-2
RMSCNV utility, 4-3
RMSDES utility, 2-2
RMSDSP utility, 5-2
RMSIFL utility, 3-2
RMSRST utility, 7-2
Prompting
RMSDES utility
conditional, 2-1, 2-13, 2-15, 2-17, 2-29
PROTECTION
RMSDES utility, 2-33
Protection
RMSDEF utility, A-18
RMSRST utility, 7-7

Query mode
RMSBCK utility, 6-5 to 6-7, 6-10
RMSRST utility, 7-6 to 7-8, 7-11
QUIT command
RMSDES utility, 2-2, 2-12, 2-18

Record format
RMSDEF utility, A-8
Record formats
RMSCNV utility, 4-14
Revision date, file
RMSBCK utility, 6-8
REWIND_MAGTAPE
RMSDES utility, 2-34
RMSBCK, 5-1
RMSBCK (RMS-11 File Back-Up Utility), 1-1, 3-3, 4-5, 6-1, 7-1, 7-5, 7-9 to 7-10
/[NO]QU switch, 6-5 to 6-7, 6-10
/CD switch, 6-8 to 6-9
/ID switch, 6-4 to 6-5
/RA switch, 6-6 to 6-7, 6-9
/RC switch, 6-7, 6-9
/RD switch, 6-8 to 6-9
/RW switch, 6-7
/SL switch, 6-6, 6-10
/SU switch, 6-7
back-up file contents, 6-1
back-up medium, 6-3
command string, 6-2 to 6-3
container file, 6-1
data integrity checking, 6-6 to 6-7, 6-9
examples, 6-10
global switches, 6-5
HELP, 6-4
indirect command file, 6-2
input file switches, 6-8
intersystem file transfer, 1-2
invoking, 6-2
magnetic tape, 6-7
multivolume, 6-9
messages, C-1
output file switches, 6-6
processing, 6-8
data integrity checking, 6-9
explicit/implicit file-spec, 6-9
extended diagnostic messages, 6-10
summary listing, 6-10
prompt, 6-2
Query mode, 6-6
query mode, 6-5 to 6-7, 6-10
summary listing, 6-10
terminating, 6-2
RMSCNV (RMS-11 File Conversion Utility), 1-1, 3-1, 4-1
/AP switch, 4-5, 4-7, 4-13, 4-15
/BL switch, 4-8
/CA switch, 4-8
/EO switch, 4-8
/FO switch, 4-9, 4-15
/ID switch, 4-5, 4-9
/IM switch, 4-9
/KN switch, 4-10, 4-16
/KR switch, 4-10, 4-16
/LO switch, 4-10
/MA switch, 4-10
/ML switch, 4-11
/PD switch, 4-12
/SL switch, 4-12
/SU switch, 4-5, 4-7, 4-13, 4-15
/TR switch, 4-13
/WF switch, 4-14
ASCII stream files, 4-3, 4-8
block mode, 4-1
command string, 4-3 to 4-4
CTRL/Z
file terminator, 4-8
examples, 4-16
HELP, 4-5
indirect command file, 4-4
intersystem file transfer, 1-2, 4-1
invoking, 4-3
key name, 4-10
key of reference, 4-10
mass-insertion mode, 4-10
messages, C-1
network facilities, 4-1
processing, 4-14
input file organization, 4-16
output file organization, 4-15

prompt, 4-3
summary listing, 4-12
supersession, 4-13
terminating, 4-4
RMSDEF (RMS-11 File Definition Utility), 1-1, A-1
/ID switch, A-5
alignment, A-14
allocation
data, A-16
file, A-14
alternate key, definition, A-11
area definition, A-13
area numbers, A-16
block boundaries, A-9
bucket size, A-15
carriage control, A-9
command file, A-5
command string, A-5
contiguity, A-15
default extension quantity, A-15
file creation, A-19
file organization, A-8
file specification, A-7
fill numbers, A-16
fixed control area, A-8
HELP, A-5
invoking, A-5
location, A-14
magnetic tape, A-8, A-10
maximum record number, A-9
maximum record size, A-9
messages, A-20
placement control, A-14
primary key, definition, A-10
processing, A-1
protection, A-18
record format, A-8
terminating, A-5
RMSDES (RMS-11 File Design Utility), 1-1, 2-1, 4-2, 5-1
allocation
areas, 2-10, 2-25
areas
defining, 2-1, 2-42, B-1
and BUCKET_SIZE attribute, 2-32
and DATA_AREA attribute, 2-40
and INDEX_AREA attribute, 2-41
and LEVEL1_INDEX_AREA attribute, 2-41
and SIZE attribute, 2-36
by default, 2-43
explicitly, 2-43
defining by default, 2-17, 2-38
CREATE command, 2-8
SET ALL command, 2-14
defining explicitly, 2-17
numbering, 2-8, 2-28, 2-43
attributes, 2-29
ALLOCATION (area section), 2-44
ALLOCATION (file section), 2-31
BLOCK_SPAN, 2-37
BUCKET_SIZE (area section), 2-43, 2-45
BUCKET_SIZE (file section), 2-32
CARRIAGE_CONTROL, 2-37
CHANGES, 2-40
clearing, 2-22, 2-25
clearing individual values, 2-26
clearing sections, 2-26
CLUSTER_SIZE, 2-31
CONTIGUOUS (area section), 2-46
CONTIGUOUS (file section), 2-35
CONTROL_FIELD_SIZE, 2-37
DATA_AREA, 2-40
DATA_FILL, 2-40
displaying, 2-15, 2-26
displaying individual values, 2-26
displaying sections, 2-27
DUPLICATES, 2-40, 2-42, 2-44
EXACT_POSITIONING, 2-47
EXTENSION (area section), 2-45
EXTENSION (file section), 2-32
FILE PLACEMENT, 2-30
FORMAT, 2-36
INDEX_AREA, 2-41
INDEX_FILL, 2-41
informational, 2-1, 2-17, 2-29
LEVEL1_INDEX_AREA, 2-41
MAGTAPE_BLOCK_SIZE, 2-34
MAX_RECORD_NUMBER, 2-34
NAME (file section), 2-31
NAME (key section), 2-38
NULL_KEY, 2-39
NULL_VALUE, 2-39
ORGANIZATION, 2-31
OWNER, 2-34
POSITION, 2-46
PROTECTION, 2-33
resetting, 2-21
REWIND_MAGTAPE, 2-34
RMS-11, 2-1, 2-17, 2-29
SEG0_POSITION, 2-42
SEGn_LENGTH, 2-42
SEGn_POSITION, 2-42
setting, 2-13
setting from data file, 2-25

setting from description file, 2-24
setting individual values, 2-21
setting sections, 2-21
SIZE, 2-36
SOURCE, 2-30
SUPERSEDE, 2-35
TARGET, 2-30
TYPE, 2-38
CLEAR command, 2-6, 2-18, 2-25 to 2-26
defining areas by default, 2-44
in indirect command file, 2-4
PROTECTION attribute, 2-33
with CREATE, 2-8, 2-28
with SAVE, 2-27
clearing attributes, 2-6
commands, 2-4
CREATE command, 2-7, 2-18, 2-28, 2-31
defining areas by default, 2-44
defining areas explicitly, 2-43
INDEX_AREA attribute, 2-41
key numbering, 2-38
LEVEL1_INDEX_AREA attribute, 2-41
setting attributes, 2-21
with GET, 2-10, 2-25
creating data files, 2-28
CTRL/Z command, 2-2, 2-9, 2-18
data file, 2-3, 2-17, 2-24
creating, 2-3, 2-7, 2-28
GET command, 2-10
defaults, 2-21
defining areas by default
CREATE command, 2-28
description file, 2-3, 2-17, 2-24
EXIT command, 2-10
GET command, 2-10
SAVE command, 2-27
saving the design, 2-13
design buffer, 2-1
using, 2-16
design session, 2-1
resuming, 2-22
ESC command, 2-9, 2-21
examples, 2-47
EXIT command, 2-3, 2-9, 2-18, 2-27
GET command, 2-10, 2-17, 2-24
PROTECTION attribute, 2-33
with SAVE, 2-28
HELP command, 2-11
indirect command file, 2-3
comments, 2-4
invoking, 2-2
keys
numbering, 2-8, 2-28, 2-38
magnetic tape, 2-34
messages, 2-52
multiple areas, 2-7, 2-43
multiple keys, 2-7, 2-38
prompt, 2-2
conditional, 2-14 to 2-15, 2-21
prompting
conditional, 2-1, 2-13, 2-15, 2-17, 2-29
QUIT command, 2-2, 2-12, 2-18
SAVE command, 2-13, 2-18, 2-27
setting attributes, 2-21
saving the design, 2-13, 2-22, 2-27
sections, design buffer, 2-1, 2-16, 2-29
area section, 2-29, 2-42
file section, 2-29, 2-31
key section, 2-29, 2-38
key segments, 2-41
record section, 2-29, 2-36
system section, 2-29 to 2-30
SET command, 2-13, 2-17, 2-20, 2-33
allocation, 2-10
defining areas by default, 2-17, 2-43
defining areas explicitly, 2-43
INDEX_AREA attribute, 2-41
LEVEL1_INDEX_AREA attribute, 2-41
PROTECTION attribute, 2-33
with GET, 2-25
SHOW
PROTECTION attribute, 2-33
SHOW ALL command, 2-4
defining areas by default, 2-17
defining areas explicitly, 2-43
setting attributes, 2-21
SHOW command, 2-15, 2-18, 2-26
allocation, 2-10
with GET, 2-25
SHOW ID command, 2-16
terminating, 2-2, 2-9, 2-12
version, 2-16
RMSDSP (RMS-11 File Display Utility), 1-1, 5-1, 6-3, 7-5, 7-10
/BP switch, 5-3 to 5-4
/FU switch, 5-2, 5-4
/ID switch, 5-3, 5-5
ANSI-standard tape, 5-3

command string, 5-2 to 5-3
container file, 5-1, 5-3 to 5-4
detailed display, 5-4
display information, 5-1
displays, 5-5
HELP, 5-3
indirect command file, 5-2
intersystem file transfer, 1-2
invoking, 5-2
magtape allocation information, 5-2
messages, C-1
mounting tapes, 5-3
prompt, 5-2
switches, 5-3 to 5-4
terminating, 5-2
RMSIFL (RMS-11 Indexed File Load Utility), 1-1, 3-1, 4-1 to 4-2
/DE switch, 3-7, 3-10, 3-14
/ER switch, 3-5 to 3-6
/ID switch, 3-3, 3-5
/KR switch, 3-8 to 3-9
/LO switch, 3-6
/NOER switch, 3-6
/NOSO switch, 3-8 to 3-10
/PD switch, 3-6 to 3-7, 3-11
/TR switch, 3-7, 3-11
calculating file work space, 3-12
command string, 3-2 to 3-3
examples, 3-14
global switch, 3-5
HELP, 3-3
indirect command file, 3-2
input file switches, 3-7
invoking, 3-2
messages, C-1
output file switches, 3-5
processing, 3-9
building alternate indexes, 3-12
command string checking, 3-9
loading output file, 3-10
reading and sorting input, 3-9
sorting alternate key files, 3-11
summary, 3-1
terminating, 3-12
prompt, 3-2
sort error codes, D-1
sort work files, 3-1, 3-7, 3-12 to 3-13
switches, 3-3, 3-5
terminating, 3-2
RMSRST (RMS-11 File Back-Up Utility)
indirect command file, 7-2
RMSRST (RMS-11 File Restoration Utility), 1-1, 4-2, 5-4, 6-1, 6-3, 6-8, 7-1
/[NO]CV switch, 7-5, 7-10
/[NO]QU switch, 7-6 to 7-8, 7-12
/BD switch, 7-8, 7-10
/FR switch, 7-7
/ID switch, 7-3, 7-5
/OA switch, 7-9 to 7-10
/RA switch, 7-7, 7-11
/RC switch, 7-8, 7-11
/SE switch, 7-3, 7-9
/SL switch, 7-6, 7-12
/SU switch, 7-8
command string, 7-2 to 7-3
data integrity checking, 7-6, 7-8, 7-11
examples, 7-12
global switches, 7-5
HELP, 7-3
input file switches, 7-8
intersystem file transfer, 1-2, 7-1, 7-10
/FR switch, 7-7
file-version number conversion, 7-5
invoking, 7-2
messages, C-1
output file switches, 7-7
processing, 7-9
data integrity checking, 7-11
explicit/implicit file-spec, 7-10
extended diagnostic messages, 7-11
restoration account, 7-10
summary listing, 7-12
prompt, 7-2
query mode, 7-6 to 7-8, 7-11
summary listing, 7-12
terminating, 7-2
SAVE command
RMSDES utility, 2-13, 2-18, 2-27
setting attributes, 2-21
Saving the design
RMSDES utility, 2-22, 2-27
See also Description file
Sections, design buffer
RMSDES utility, 2-1, 2-29
area section, 2-29, 2-42
file section, 2-29, 2-31
key section, 2-29, 2-38
key segments, 2-41
record section, 2-29, 2-36
system section, 2-29 to 2-30
Sections, design buffer (RMSDES), 2-16
SEG0_POSITION
RMSDES utility, 2-42
SEGn_LENGTH
RMSDES utility, 2-42

SEGn_POSITION
  RMSDES utility, 2-42
Serial, nondisk device (RMSIFL), 3-1, 3-12
SET command
  RMSDES utility, 2-13, 2-17, 2-20, 2-33
    allocation, 2-10
    defining areas by default, 2-17, 2-43
    defining areas explicitly, 2-43
    INDEX_AREA attribute, 2-41
    LEVEL1_INDEX_AREA attribute, 2-41
    PROTECTION attribute, 2-33
    with GET, 2-25
Setting attributes (RMSDES)
  See Attributes
SHOW ALL command
  RMSDES utility
    defining areas by default, 2-17
    defining areas explicitly, 2-43
SHOW command
  RMSDES utility, 2-15, 2-18, 2-26
    allocation, 2-10
    PROTECTION attribute, 2-33
    SHOW ALL, 2-4
      setting attributes, 2-21
    with GET, 2-25
SHOW ID command
  RMSDES utility, 2-16
SIZE
  RMSDES utility, 2-36
Sort routine error codes (RMSIFL), D-1
Sort work files (RMSIFL), 3-1, 3-10, 3-12
  allocating space
    primary key sort, 3-13
  device allocation, 3-7
Sorting (RMSIFL)
  alternate key, 3-11 to 3-13
  alternate key files, 3-7
  bypassing primary key sort, 3-8 to 3-9
  input file organization, 3-9
  primary key, 3-9
  work files, 3-7, 3-13
SOURCE
  RMSDES utility, 2-30
Summary listing
  RMSBCK utility, 6-10
  RMSCNV utility, 4-12
  RMSRST utility, 7-12
SUPERSEDE
  RMSDES utility, 2-35
Supersession
  RMSBCK utility, 6-7
  RMSCNV utility, 4-7, 4-13
  RMSDEF utility, A-7
  RMSRST utility, 7-8
Switches
  RMSBCK utility, 6-4 to 6-5
  RMSCNV utility, 4-5, 4-7
  RMSDSP utility, 5-3 to 5-4
  RMSIFL utility, 3-3, 3-5
  RMSRST utility, 7-3, 7-5
  See also individual switch entries
  See also RMSBCK, RMSCNV, RMSDSP, RMSIFL, RMSRST
System disk
  RMSBCK utility, 6-1
  RMSRST utility, 7-1

TARGET
  RMSDES utility, 2-30
Terminating
  RMSBCK utility, 6-2
  RMSCNV utility, 4-4
  RMSDEF utility, A-5
  RMSDES utility, 2-2, 2-9, 2-12
  RMSDSP utility, 5-2
  RMSIFL utility, 3-2
  RMSRST utility, 7-2
Terminating processing (RMSIFL), 3-12
Truncation
  RMSCNV utility, 4-13
  RMSIFL utility, 3-7
TYPE
  RMSDES utility, 2-38

Unit record device
  RMSCNV utility, 4-5

Version
  RMSDES utility, 2-16
Version number
  RMSBCK utility, 6-5
  RMSCNV utility, 4-9
  RMSDEF utility, A-5
  RMSDSP utility, 5-5
  RMSIFL utility, 3-5
  RMSRST utility, 7-5

READER'S COMMENTS

NOTE: This form is for document comments only. DIGITAL will use comments submitted on this form at the company's discretion. If you require a written reply and are eligible to receive one under Software Performance Report (SPR) service, submit your comments on an SPR form.

Did you find this manual understandable, usable, and well organized? Please make suggestions for improvement.

Did you find errors in this manual? If so, specify the error and the page number.

Please indicate the type of user/reader that you most nearly represent.

- ☐ Assembly language programmer
- ☐ Higher-level language programmer
- ☐ Occasional programmer (experienced)
- ☐ User with little programming experience
- ☐ Student programmer
- ☐ Other (please specify) _____

Name _____ Date _____

Organization _____

Street _____

City _____ State _____ Zip Code _____
or Country

— — Do Not Tear - Fold Here and Tape —

digital

No Postage Necessary if Mailed in the United States

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO.33 MAYNARD MASS.

POSTAGE WILL BE PAID BY ADDRESSEE

BSSG PUBLICATIONS ZK1-3/J35
DIGITAL EQUIPMENT CORPORATION
110 SPIT BROOK ROAD
NASHUA, NEW HAMPSHIRE 03061

— — Do Not Tear - Fold Here —

Cut Along Dotted Line