December 1977

This document describes the purpose and effect of the RMS-11 utilities and the procedures and conventions for their use on the RSTS/E operating system.

RSTS/E
RMS-11 Utilities User's Guide
Updated for V6C

Order No. AA-0003B-TC

SUPERSESSION/UPDATE INFORMATION:	This manual contains information concerning RSTS/E V6C and RMS-11 V1.5 as of December, 1977.
OPERATING SYSTEM AND VERSION:	RSTS/E V6C and later
SOFTWARE VERSION:	RMS-11 V1.5

To order additional copies of this document, contact the Software Distribution Center, Digital Equipment Corporation, Maynard, Massachusetts 01754

digital equipment corporation · maynard, massachusetts

First Printing, February 1977
Revised: December 1977

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 only be used or copied 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 or its affiliated companies.

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

The following are trademarks of Digital Equipment Corporation:

DIGITAL	DECsystem-10	MASSBUS
DEC	DECtape	OMNIBUS
PDP	DIBOL	OS/8
DECUS	EDUSYSTEM	PHA
UNIBUS	FLIP CHIP	RSTS
COMPUTER LABS	FOCAL	RSX
COMTEX	INDAC	TYPESET-8
DDT	LAB-8	TYPESET-11
DECCOMM	DECSYSTEM-20	TMS-11
ASSIST-11	RTS-8	ITPS-10

CONTENTS

TABLES

FIGURES

PREFACE

The RSTS/E RMS-11 Utilities User's Guide describes the use of RMS-11 V1.5 utility programs on the RSTS/E V6C operating system.

The reader of this manual is assumed to be familiar with the material presented in the Introduction to RMS-11 manual. Further, since several RSTS/E language processors utilize RMS-11 to create and access files, this manual is intended for users with varying language backgrounds. The reader, therefore, should also be familiar with the appropriate language processor (see the RSTS/E Documentation Directory).

CHAPTER 1

INTRODUCTION

1.1 RMS-11 UTILITIES OVERVIEW

Record Management Service for PDP-11 operating systems (RMS-11) is a set of file processing functions. These functions are available to programmers writing applications in assembly or high-level languages and to all users working at interactive terminals. To users writing application programs, RMS-11 is the underlying mechanism supporting the source language's file processing statements. To users working from interactive terminals, RMS-11 is a collection of utility programs that create, manipulate, and maintain files from system command level. This manual describes those utilities.

1.1.1 General Conventions

Though each utility has a distinct purpose, many aspects of the terminal interface with RMS-11 are common to all utilities. The command lines used to invoke utilities follow conventions in the areas of format of file specifications and the format and use of options that modify the actions of utilities. Additionally, both the format of error messages generated by utilities and the means for user termination of utility processing follow common conventions.

1.1.2 Functions of RMS-11 Utilities

The BACKUP utility allows the user to ensure against the total loss of data due to hardware failure or software error. When invoked, this utility creates backup copies of one or more files on a storage medium. Thereafter, if the original files are lost or damaged, these copies can be used to return the files to their original condition.

The CONVERT utility permits the insertion of records into an indexed or relative file, the creation of a sequential file from the records of another file of any organization, and the appending of records to an existing sequential file.

Both versions of the DEFINE utility create RMS-11 files, one via command line interpretation, the other as the result of an interactive sequence of questions and requests for data. The amount and type of information needed to create a file and define its attributes varies with the file organization selected; for example, an indexed file requires key definitions, while sequential and relative files do not.

The DISPLAY utility lists the attributes of one or more files or the names of RMS-11 files in backup format on magnetic tape.

The RESTORE utility reverses the effect of the BACKUP utility. Whereas BACKUP creates backup copies of files, RESTORE takes backup files as input and produces standard RMS-11 files as output. The structure, content, and attributes of these restored files are those of the original files at the time they were backed up.

1.2 ORGANIZATION OF INFORMATION IN THIS MANUAL

The organization of this manual corresponds to the two areas discussed in the preceding overview.

Chapter 2 details general conventions applicable to the invocation and use of all RMS-11 utilities.

Chapter 3 consists of detailed descriptions of RMS-11 utilities including the distinctive capabilities, command line format, and sample command invocations for each utility.

Appendix A contains a description of error messages that may be displayed by RMS-11 utilities and suggested user actions in response to each message.

Appendix B contains a list of all HELP messages available from the RMS-11 utilities.

CHAPTER 2

GENERAL CONVENTIONS

This chapter discusses the features of the terminal interface with RMS-11 utilities.

The first section of this chapter shows how the user gains access to RMS-11 utilities and enters command lines. Several methods of entering utility command lines, including the convention for entering very lengthy command lines, are described.

Each utility command line consists of one or more file specifications and command options. The second section of this chapter, therefore, discusses the format of file specifications (including defaults and wild card conventions) and three types of command options: global options, file specification options, and output file protection options.

RMS-11 utilities type an error message at the terminal if the user enters a command line incorrectly or if an error occurs during processing. Certain errors, called fatal errors, cause the termination of utility processing. Other errors, known as non-fatal errors, permit processing to continue. The third section of this chapter describes the format of error messages and how the user distinguishes between fatal and non-fatal errors.

Lastly, RMS-11 utilities provide two means for user termination of utility processing. Each means employs a control character typed by the user at the terminal. The final section of this chapter shows both methods of terminating utilities.

NOTE

Throughout this chapter, all text displayed by the operating system or RMS-11 utilities is underlined to distinguish it from user input. Further, a sample utility command illustrates the various conventions. This command line, using the CCL form of invoking the BACKUP utility, is as follows:

```
BCK  DK0:*.*/QU=FIL1.DAT,DK1:[50,1]FIL2.DAT,FIL3.*/CD:17-SEP-76
```

2.1 ACCESSING UTILITIES AND ENTERING COMMAND LINES

The user can access an RMS-11 utility through either of two methods. After having gained access to a specific utility, the user similarly has two choices for entering command lines. The subsections that follow describe, respectively, the methods of accessing utilities and entering command lines.

2.1.1 Accessing Utilities

Either of two methods can be used to gain access to an RMS-11 utility:

1. Typing the RUN command
2. Typing a CCL command

The paragraphs below describe each method.

2.1.1.1 **Typing the RUN Command** - In response to the system READY prompt, the user can type the following general form of the RUN command:

```
RUN $filename
```

where

filename is one of the following:

RMSBCK	-	BACKUP utility
RMSCNV	-	CONVERT utility
RMSDFN	-	DEFINE utility
RMSDSP	-	DISPLAY utility
RMSRST	-	RESTORE utility

As an example, the user can gain access to the BACKUP utility by typing the following:

```
RUN $RMSBCK
```

The BACKUP utility types the following prompt to indicate that it is ready to accept a command line:

```
BCK>
```

2.1.1.2 **Typing a CCL Command** - If the RMS-11 utilities are included in the CCL for the system, the user may invoke a utility by typing its three-character CCL name. Table 2-1 lists the recommended set of CCL names for RMS-11 utilities.

Table 2-1
CCL Names for RMS-11 Utilities

Utility	CCL Name
Backup	BCK
Convert	CNV
Define	DFN
Display	DSP
Restore	RST

As an example, the user can gain access to the BACKUP utility by typing:

```
BCK
```

The utility indicates its readiness to accept a command line by typing the following prompt:

```
BCK>
```

Alternatively, the user may employ the following general form of the CCL command:

```
BCK     <command-line>
```

This form causes the BACKUP utility to run, process <command-line> (described in the following section), and return to the READY state.

2.1.2 Entering Command Lines

There are two methods for entering utility command lines:

1. Typing the complete utility command line
2. Using an indirect command

Each of these methods is discussed in the following paragraphs.

2.1.2.1 **Typing Complete Command Lines** - Either in response to a utility's prompt for input or as part of the CCL command, the user can enter the complete command line necessary for processing. Three examples of this method are shown below.

1. Example 1. The user employs the RUN command to invoke the BACKUP utility. The utility prompts for the command line. After execution is completed, the utility reprompts for additional command input.

```
READY
RUN $RMSBCK
BCK>DK0:*.*/QU=FIL1.DAT,DK1:[50,1]FIL2.DAT,FIL3.DAT/CD:17-SEP-76
BCK>
```

2. Example 2. The user invokes the BACKUP utility by typing its CCL name. The utility prompts for the command line. After execution is completed, the utility reprompts for additional command input.

```
READY
BCK
BCK>DK0:*.*/QU=FIL1.DAT,DK1:[50,1]FIL2.DAT,FIL3.DAT/CD:17-SEP-76
BCK>
```

3. Example 3. The user invokes the utility by typing its CCL name followed by a space and the complete utility command line. After execution of the BACKUP utility, the system again prints the READY prompt.

```
READY
BCK  DK0:*.*/QU=FIL1.DAT,DK1:[50,1]FIL2.DAT,FIL3.DAT/CD:17-SEP-76
READY
```

For lengthy command lines, the user can type a partial command line followed by a hyphen (-) as the last character on the line. Each appearance of a hyphen as the last character on a line causes the utility to reprompt. Following this reprompt, the user may continue typing the remainder of the command line.

This continuation mechanism is shown in the examples below.

Example 1.

```
READY
RUN $RMSBCK
BCK>DK0:*.*/QU=FIL1.DAT,DK1:[50,1]FIL2.DAT,-
BCK>FIL3.DAT/CD:17-SEP-76
BCK>
```

Example 2.

```
READY
BCK  DK0:*.*/QU=FIL1.DAT,DK1:[50,1-
BCK>]FIL2.DAT,FIL3.DAT/CD:17-SEP-76
READY
```

Note that the user can continue a command line at any point within the line.

2.1.2.2 **Using Indirect Command** - The second method of entering a utility command line is the use of an indirect command. When the user specifies an indirect command, the utility interprets the contents of the specified file as a series of one or more command lines. The advantage of using an indirect command is that the user can type a commonly used command line sequence once and store it in a file rather than repeatedly typing the sequence for each execution.

An indirect command has the @ character as the first character of the command line. Immediately following the @ character is a file specification. The following is the complete format for an indirect command:

```
@dev:[proj,prog]file.ext
```

The user may optionally omit certain elements of the file specification. The following defaults are applied to form a full file specification:

```
dev:            - SY:
[proj,prog]     - current user account
ext             - CMD
```

The following examples show the use of indirect commands.

Example 1.

```
READY
RUN $RMSBCK
BCK>@ALPHA
BCK>
```

Example 2.

```
READY
BCK @BETA.BAC
READY
```

In the first example, the user has specified only the filename, ALPHA. The device, account, and extension fields are defaulted. In the second example, the user has specified a filename and extension while again allowing the device and account elements of the file specification to default.

2.2 COMMAND LINE ELEMENTS

A utility command line consists of the following elements:

- One or more file specifications
- Command options

The following subsections describe each of these elements.

2.2.1 File Specifications

The number and type of file specifications that can appear in a command line vary according to the utility being invoked.

When the action of a utility involves the creation of one or more files, a single output file specification is the first element of the command line. In the sample BACKUP command line:

```
BCK  DK0:*.*/QU=FIL1.DAT,DK1:[50,1]FIL2.DAT,FIL3.*/CD:17-SEP-76
```

The following is an output file specification:

```
DK0:*.*
```

When an input file is also specified, the input file specification follows the output file specification and an intervening equal sign (=). When specifying several input files, the user must separate each input file specification from the preceding by a comma. Thus, the sample BACKUP command shows the following three input file specifications:

1. FIL1.DAT

2. DK1:[50,1]FIL2.DAT

3. FIL3.*

Showing only the CCL command and the command line elements discussed so far, the sample BACKUP command now appears as:

```
BCK  DK0:*.*=FIL1.DAT,DK1:[50,1]FIL2.DAT,FIL3.*
```

The equal sign shown in the sample command is necessary only when the user specifies both output and input files. Certain RMS-11 utilities do not require both input and output files. In entering command lines for such utilities, the user must not type an equal sign. The first example below shows a utility command line with only input file specifications. The second example shows a command that allows only an output file specification.

1. DSP BETA.MST,[20,11]GAMMA.DAT

2. DFN ALPHA.DAT

Whether the utility requires both input and output files, just input files, or just output files, file specifications have the following format:

```
dev:[proj,prog]name.ext
```

where

dev	is the device specification.
[proj,prog]	is the account specification.
name	is a 1- to 6-character name of the file. If the user types more than 6 characters, the system automatically truncates the name to the first 6 characters.
ext	is a 1- to 3-character file extension. If the user wishes a null extension, a period must follow the name of the file. If the user types more than 3 characters for a file extension, the system automatically truncates the extension to the first 3 characters.

In the sample BACKUP command:

```
BCK  DK0:*.*/QU=FIL1.DAT,DK1:[50,1]FIL2.DAT,FIL3.*/CD:17-SEP-76
```

The second input file demonstrates the complete format of a file specification:

```
DK1:[50,1]FIL2.DAT
```

Only the above input file specification demonstrates the complete format because the sample BACKUP command shows two file specification conventions. First, if the user omits certain fields within a file specification, RMS-11 utilities apply defaults to them. Second, if the user places an asterisk (wild card) in a field, the utility selects all files that satisfy the remaining fields. The following subsections describe these defaults and wild cards.

2.2.1.1 **Defaults in File Specifications** - The user can supply incomplete file specifications in utility command lines. If any of the fields (except filename and extension) of an output file specification or a single input file specification is omitted, each utility uses a default. These default values are listed in Table 2-2.

Table 2-2
File Specification Defaults

Field	Default Value
dev	SY: (the system device public structure).
[proj,prog]	The current user account number.
name	No default -- must be specified explicitly or implicitly using an asterisk.
ext	No default -- must be specified explicitly or implicitly using an asterisk, or null through the use of a period.

When the user supplies multiple input file specifications in a command line, the default values for the second and subsequent file specifications are the corresponding field values used for the immediately preceding file specification.

The sample BACKUP command illustrates the use of file specification defaults. The sample command, again including only the elements discussed so far, is:

```
BCK  DK0:*.*=FIL1.DAT,DK1:[50,1]FIL2.DAT,FIL3.*
```

Assuming a current user account number of [200,11], this command line is equivalent to:

```
DK0:[200,11]*.*=SY:[200,11]FIL1.DAT,DK1:[50,1]FIL2.DAT,DK1:[50,1]FIL3.*
```

Note that the current user account number is the default for the output file specification and the first input file specification. Further, the default device for file FIL1.DAT is SY:. The defaults applied to file specification FIL3.*, however, are taken from the immediately preceding file specification.

2.2.1.2 **Wild Cards in File Specifications** - In some cases, the user may specify more than one file in a file specification by using an asterisk (*) convention called a wild card. An asterisk may be placed in any field of a file specification except the device field. Wild cards in input file specifications have a different meaning from wild cards in output file specifications.

For input file specifications, the asterisk or wild card indicates that the contents of the field are to be ignored and all files that satisfy the remaining fields are to be selected. The following example from the sample command line shows the use of wild cards in an input file specification.

FIL3.* - Select all files with the name FIL3 and any extension.

For output file specifications, the asterisk or wild card indicates that the field is to be replaced with the corresponding field of each input file specification. Each time an input file is selected for processing, RMS-11 utilities use appropriate elements of the input file specification to supply values for fields in the output file specification that contain wild cards. Thus, the output file specification in the sample command

DK0:*.* indicates that files are to be created under the current user account on DK0 with the same filenames and extensions as each input file selected.

2.2.2 Command Options

Command options modify the action of a utility or specify additional information needed for utility processing. The following is the general form of an option.

```
/xx[:arg[:arg]]
```

where

xx is the 2-character name of the option.

arg is an argument value associated with the option. Certain options do not require argument values. Other options require a single value or permit multiple values. The initial argument value is separated from the option name by a colon. Subsequent argument values, if any, are similarly preceded by colons.

The following are examples of options, taken from the sample BACKUP command line:

```
/QU
/CD:17-SEP-76
```

Certain options can be negated. That is, the user can explicitly specify that the function represented by an option is not desired. The general form of a negated option is:

```
/NOxx
```

where

xx is the 2-character name of the option whose function the user does not desire.

The user may specify three types of options in a command line:

1. Global options
2. File specification options
3. Output file protection option

The utility descriptions in Chapter 3 specify the number, type, argument values, and negatability of options allowed for each utility. The following subsections use the sample BACKUP command to illustrate types of command options. The user should refer to the BACKUP utility description in Chapter 3 for a detailed explanation of the sample options.

2.2.2.1 **Global Options** - Global options modify the total action of the utility being invoked. Therefore, the user may specify a global option immediately following any file specification in a command line. The actual placement of the option is immaterial.

As an example, consider the sample BACKUP command line:

```
BCK  DK0:*.*/QU=FIL1.DAT,DK1:[50,1]FIL2.DAT,FIL3.DAT/CD:17-SEP-76
```

One global option is specified:

/QU - Enable Query mode.

Since global options may be placed after any file specification in a command line, all of the following commands are equivalent to the sample command:

1. BCK DK0:*.*=FIL1.DAT/QU,DK1:[50,1]FIL2.DAT,FIL3.*/CD:17-SEP-76
2. BCK DK0:*.*=FIL1.DAT,DK1:[50,1]FIL2.DAT/QU,FIL3.*/CD:17-SEP-76
3. BCK DK0:*.*=FIL1.DAT,DK1:[50,1]FIL2.DAT,FIL3.*/QU/CD:17-SEP-76

2.2.2.2 **File Specification Options** - Unlike global options, the position of file specification options in a command line is significant. A file specification option applies only to the immediately preceding file specification.

The following file specification option appears in the sample BACKUP command:

/CD:17-SEP-76

Note the placement of this option in the sample command:

```
BCK  DK0:*.*/QU=FIL1.DAT,DK1:[50,1]FIL2.DAT,FIL3.DAT/CD:17-SEP-76
```

The /CD option applies only to the immediately preceding file specification, FIL3.*.

2.2.2.3 **Output File Protection Option** - Certain RMS-11 utilities allow the user to assign protection codes to output files. When protection codes can be specified, the user may append the protection option to an output file specification. The general form of the protection option is:

/PR:<prot>

where

prot — is a protection code restricting access to a file. The degree of restriction is determined by a code or combination of codes as shown in Table 2-3.

Table 2-3
Protection Codes

Code	Meaning
1	Read protect against owner
2	Write protect against owner
4	Read protect against owner's project number
8	Write protect against owner's project number
16	Read protect against all others who do not have owner's project number
32	Write protect against all others who do not have owner's project number.

These codes can be used singularly, or combined, to provide greater degrees of protection. For example, a protection code of <48> would deny read or write access to anyone logged into the system under an account number whose project number differs from the owner. The code <48> is the sum of 32 (write protect against all others) and 16 (read protect against all others). Similarly, the code <42> protects a file against any write operations (32 = write protect against all others, 8 = write protect against other group members, and 2 = write protect against owner, 42 = 32+8+2). Unless an installation has selected otherwise, code <60> is the protection given to files by default.

2.3 ERROR MESSAGES

When RMS-11 utilities encounter an error condition during execution, they print an error message at the user's terminal.

There are two types of error messages, fatal error messages and non-fatal error messages. The following subsections describe both types of messages. The user should refer to Appendix A for a complete listing of all error messages.

2.3.1 Fatal Error Messages

When an error occurs that requires the termination of processing, RMS-11 utilities display a fatal error message. The format of fatal error messages is as follows:

```
?utl -- message
```

where

? — Indicates a fatal error condition

utl — Is a 3-character abbreviation of the utility displaying the error message

message — Is a brief description of the error.

The following are examples of fatal error messages:

```
?BCK -- COMMAND SYNTAX ERROR
?DFN -- ILLEGAL RMS RECORD FORMAT
```

2.3.2 Non-fatal Error Messages

When a non-fatal error occurs, RMS-11 utilities display a non-fatal error message and continue processing. The format of a non-fatal error message is as follows:

```
utl--message
```

where

utl — is a 3-character abbreviation of the utility displaying the error message

message — is a description of the error.

The following is an example of a non-fatal error message that could result from the processing of the sample BACKUP command line:

```
BCK -- FILE NOT FOUND - SY:[200,11]FIL1.DAT
```

The preceding error message indicates that file FIL1.DAT in the list of input files in the sample command could not be found in the default account on SY:. The condition is non-fatal since additional input files have been specified and can be processed.

2.4 TERMINATING UTILITIES

The user terminates utility processing by typing CTRL/C or CTRL/Z. The following subsections illustrate both methods of terminating utilities.

2.4.1 Terminating Utilities with CTRL/C

Typing a CTRL/C (by holding down the CTRL key and typing the C key and releasing both) always causes an immediate return to the default run-time system and the printing of READY. Processing is aborted at the moment the CTRL/C is entered. Each of the examples below illustrates the use of CTRL/C.

1.
```
BCK>^C
READY
```

2.
```
READY
BCK DK0:*.*/QU=^C
READY
```

2.4.2 Terminating Utilities with CTRL/Z

Typing a CTRL/Z (by holding down the CTRL key and typing the Z key and releasing both) causes a return to the default run-time system after the current processing is finished. That is, if the user types CTRL/Z while the BACKUP utility is executing a command line, the utility will display the BCK> prompt before terminating. The example below illustrates the action of typing CTRL/Z.

```
BCK>^Z
READY
```

2.5 HELP MESSAGES

All RMS-11 utilities, except the command DEFINE, respond to a stand-alone question mark (?) by displaying a message designed to help the user with its operation. The interactive DEFINE utility has a separate help message for each question asked, while the BACKUP, CONVERT, DISPLAY, and RESTORE utilities have a single help message. See Appendix B for a list of the help messages.

The question mark may be typed in response to any prompt (utilityname> or a message). The utility displays the pertinent help message, then redisplays the prompt.

CHAPTER 3

UTILITY DESCRIPTIONS

This chapter contains detailed descriptions of the RMS-11 utilities:

- The BACKUP utility creates backup copies of one or more files on a storage medium. These backup copies may be used with the RESTORE utility to replace the original files should they become lost or damaged.
- The CONVERT utility reads a file of any type and inserts the records into an indexed or relative file, creates a sequential file from the records of another file of any organization, or appends records from any file to an existing sequential file.
- The command DEFINE utility creates an RMS-11 file and defines its attributes, based on the interpretation of a command line.
- The DISPLAY utility lists the attributes of one or more files or prints the names of files found in specially formatted backup container files on magnetic tape.
- The RESTORE utility reverses the process performed by the BACKUP utility. RESTORE reads backup files as input and produces standard RMS-11 files as output. The structure, content, and attributes of these restored files is that of the original files at the time they were backed up.
- The interactive DEFINE utility creates an RMS-11 file and defines its attributes as the result of information entered by the user in response to requests for data and questions displayed by the utility.

3.1 CONVENTIONS USED IN THIS CHAPTER

The following conventions are used to describe the format and function of utility command lines:

1. Upper case words and letters, as well as punctuation marks other than those described in this section, are typed as shown.
2. Lower case words indicate that a value is to be substituted. Specifically, in descriptions of values that may be used as arguments on options, the following notations are used:

- dd-mmm-yy - a date specification. For example, if the description were:

  /BD:dd-mmm-yy

  then the following would be correct:

  /BD:17-OCT-76

- filespec - a file specification as described in Chapter 2.

  For example, if the description were:

  /SL:filespec

  then the following would be correct:

  /SL:DK1:[200,11]BACK.LST

- n - a user selected decimal number.

  For example, if the description were:

  /AL:n

  then the following would be correct:

  /AL:180

- val - an argument selected from a specific and limited set of arguments listed in the description of the option.

3. In all cases, except where [proj,prog] is specified, brackets signify optional elements of a command string.

4. An ellipsis (...) indicates that the preceding item may be repeated any number of times.

3.2 BACKUP - CREATING BACKUP COPIES OF FILES

The BACKUP utility allows the user to ensure against the loss of data due to hardware failure or software error. Each time the user invokes this utility, specially formatted backup copies of one or more files are placed on a backup medium. Thereafter, if the original files are ever lost or damaged, the RESTORE utility and these backup copies can be used to return the files to their original condition.

The internal structure of a backup file is known only to the BACKUP and RESTORE utilities. Therefore, user programs cannot access a backup file and the data records within such a file for normal input and output operations. BACKUP, however, preserves all attributes of the original file within the backup copy. In particular, the backup file carries the account, creation and last access dates, and the protection code of the original file.

In addition to retaining original file attributes within a special backup file structure, the BACKUP utility provides three extended functions:

1. Explicit and implicit file selection

2. Extended diagnostic message capability

3. Summary listing capability

Each of these functions are described in the following paragraphs.

The BACKUP utility command line permits a single input file specification or a list of input file specifications. File specifications without wild cards explicitly identify input files for the BACKUP utility. File specifications with wild cards, however, implicitly identify a collection of files to be selected by the BACKUP utility. When using wild cards, the user can restrict the number of files selected for processing by including a date option. A date option causes the BACKUP utility to compare either the date of creation or date of last access of each file selected as a result of the wild cards. The utility will backup the file only if the internal file date conforms to the user specified date.

In addition to allowing the user to either explicitly or implicitly identify input files, the BACKUP utility provides an extended diagnostic message capability known as Query mode. When it encounters an error condition, BACKUP prints a diagnostic message at the user's terminal. Immediately following the message, BACKUP prints a question (query). Each such query requires the user to type a response (Y or N) indicating whether the utility is to continue or discontinue processing. By indicating that the utility is to continue processing, the user can ensure that BACKUP does not terminate the processing of a collection of files due to errors encountered in the processing of any one file. Optionally, the user can disable Query mode. When Query mode is disabled, the utility prints a diagnostic message if it encounters an error condition but does not give the user the option of continuing processing. Rather, the utility terminates.

Lastly, the user can specify that BACKUP is to list a summary of files selected and all error messages. BACKUP can produce this listing at the user's terminal or, optionally, store the summary listing as the contents of a file. By selecting this latter option, the user can ensure that a permanent record exists of a specific invocation of the utility.

The following restrictions apply to use of the BACKUP utility:

1. The user can specify only files residing on directory devices as input to the BACKUP utility.

2. The BACKUP utility cannot save a system disk as a bootable volume.

3. The backup medium must be on-line and mounted.

4. When the backup medium is disk, the user cannot rename output files. That is, the name of each backup file produced must be the same as the name of the original input file.

5. When the backup medium is magnetic tape, the user can use only ANSI labeled tapes. Further, the following restrictions apply to the use of ANSI labeled tapes as backup media:

   - All backup files produced as a result of the execution of a single command line are contained within a single file on magnetic tape. This file is known as a container file since it contains individual backup files. The user must explicitly name the output container file in the command line invoking the BACKUP utility. The name of the container file need not correspond to the name of any input file.

- The BACKUP utility cannot produce multi-volume tape files: it terminates if there is not enough room left on the magtape to contain all files specified in a command line.

The following subsection presents the format and several examples of the BACKUP command line.

3.2.1 Format of the BACKUP Command Line

The BACKUP command line takes the form:

```
BCK outfile[/options]=infile[/options][,infile[/options]...]
```

where

outfile is the file specification of the backup file(s) to be created by the utility.

- if the backup medium is a directory device, outfile name and extension must be *.*. BACKUP creates a backup file for each input file, using its filename and extension.
- if the backup medium is magnetic tape, both the filename and file extension of the output container file must be explicitly specified (no wild cards).

infile is the file specification of a file for which a backup copy is to be created. The user can employ wild cards to specify implicitly a collection of files.

Table 3-1 lists the options that can appear in a BACKUP command line. Detailed descriptions of each option follow the table.

Table 3-1
BACKUP Utility Options

FIELD	OPTION	DESCRIPTION	NOxx	DEFAULT PROCESS
Help	?	Print HELP message.		No help.
Global	/ID	Identify current version of BACKUP utility.		No id.
	/QU	Enable Query mode.	NOQU	QU
	/SL[:filespec]	Provide summary listing (in file, if specified).		No summary.
Outfile	/RW	Rewind magnetic tape before writing.	NORW	NORW
	/SU	Supersede old files.	NOSU	NOSU
Infile	/CD:dd-mmm-yy[:val]	Backup files based on creation date.		No date checking.
	/RD:dd-mmm-yy[:val]	Backup files based on revision date.		No date checking.

Global Options

ID - Identify current version of BACKUP utility. This option can appear alone on a command line.

QU - Enable Query mode (default). When Query mode is enabled, the BACKUP utility gives the user the option of either continuing or discontinuing processing following the occurrence of any of the following error conditions:

- Read error on input file. If the user allows processing to continue, the utility will automatically retry all read errors at the end of its processing.
- The table allocated internally for automatic retry of read errors is full.

In Query mode, the BACKUP utility displays a diagnostic message specifying the type of error and the name of the file being processed. Further, the message indicates that the user must type a response of Y or N. A response of Y indicates that the utility is to continue processing the current file and any remaining input files. A response of N indicates that the utility is to terminate processing of the current file and bypass processing any remaining input files.

When the user disables Query mode through the NOQU option, the BACKUP utility prints a diagnostic message when it encounters an error condition but does not give the user the option of continuing processing.

SL[:filespec]- Provide summary listing file. A summary listing file provides the user with a summation of what actually occurred during the execution of the command. Specifically, this file will contain:

- The names of all files actually backed up during the execution of the command.
- Copies of any error messages, including diagnostic messages produced in Query mode and a summary of uncorrectable read errors.

If the user does not specify an argument (filespec) with the SL option, BACKUP prints the summary listing file at the user's terminal. If, however, the user specifies a file specification as an argument with the SL option, BACKUP creates the specified file and writes the summary listing as the contents of the file.

NOTE

If the filespec argument is used to create a summary listing file on a disk or magnetic tape, the user can employ the QUE command to print the contents of the file at a terminal or line printer.

Outfile Options

RW - Rewind magnetic tape before writing. It is an error if RW is specified for a backup medium other than magnetic tape. This option indicates that BACKUP should rewind the volume to the beginning of tape before writing backup container files. Rewinding causes all existing files on the tape to be logically deleted. The normal default when RW is not specified is to add new container files to the logical end of tape.

SU - Supersede old files. This option applies only to directory devices. It is an error if SU is specified for magnetic tape. When the user specifies this option, the BACKUP utility supersedes files in the output account with the same name and extension as a new backup file being created.

Infile Options

CD:dd-mmm-yy[:val] - Backup files based on creation date. The user can specify this option in combination with wild cards to identify a group of files. The BACKUP utility will select for processing those files that satisfy the wild card specification and that were created on the specified date.

In addition to a specific creation date, the user can specify a second argument on the CD option. This second argument (val) can be either A or B. A indicates the user wants files selected for backup that were created after the specified date. B indicates that the user wants files to be backed up that were created before the specified date.

RD:dd-mmm-yy[:val] - Backup files based on revision date. This option is identical to the CD option except that BACKUP selects files according to a comparison of the file's date of last access with the date specified by the user.

3.2.2 Examples of BACKUP Command Lines

1. BCK /ID

   Displays the current version of the BACKUP utility.

2. BCK MT0:ALPHA.BKP=ALPHA.DAT

   A backup copy of file ALPHA.DAT under the current user account on SY: is written on the ANSI labeled magnetic tape mounted on device MT0. The name of the container file created on tape is ALPHA.BKP. Since Query mode is enabled (by default), any error conditions encountered will cause a diagnostic message to be printed at the user's terminal. The user can then type a response of Y or N indicating whether processing is to continue.

3. BCK MT1:SAFE.BKP=*.*/QU/SL

   All files under the current user account on SY: are backed up into a container file named SAFE.BKP on ANSI labeled magnetic tape. Query mode is explicitly enabled. Therefore, any errors encountered when reading input files cause diagnostic messages to be printed at the user's terminal. These messages require a response. The user must type a Y or N to indicate whether processing is to continue. When all files have been processed (or when the user has typed N to a query associated with a diagnostic message), BACKUP prints a summary listing at the user's terminal. The summary listing will specify the names of all files successfully backed up and will also contain copies of any diagnostic messages produced during execution.

4. BCK MT0:MASTER.BKP=*.DAT/CD:31-DEC-75:B/SL:BACKUP.LST

   All files under the current user account on SY: that were created before December 31, 1975 and have an extension of DAT are backed up onto ANSI labeled magnetic tape. The output container file has the name MASTER.BKP. The BACKUP utility, by default, enables Query mode, and the user must respond to any diagnostic messages typed by BACKUP. BACKUP creates a summary listing file under the current user account on SY:. The name of the summary listing file is BACKUP.LST. The file contains a listing of the names of files actually backed up along with copies of any diagnostic messages.

5. BCK DK1:[100,10]*.*/SU=[100,20]*.*/RD:01-JUN-76:A/NOQU

   All files under account [100,20] on SY: that have been accessed after June 1, 1976 are to be backed up. The backup volume is a disk mounted on DK1: The backup files are to be written into account [100,10], superseding any files with the same name and extension. In this example, the user has disabled Query mode. Therefore, the occurrence of any error that would normally cause a diagnostic message and a query will, instead, cause the entire process to be terminated.

3.3 CONVERT - MOVING RECORDS BETWEEN FILES

The CONVERT utility provides a versatile mechanism for performing any of the following functions:

- Creating a new sequential file from the records of an existing sequential, indexed, or relative file.
- Superseding an existing sequential file with the records of another sequential file or the records of an indexed or relative file.
- Appending to a sequential file the records of another sequential file or the records of an indexed or relative file.
- Inserting to an existing relative or indexed file the records of another file of any organization.

All the functions performed by the CONVERT utility involve the movement of records from one file to another file. CONVERT reads records from the specified input file and writes them into the specified output file. The manner in which records are read and written depends on the file organizations of both the input and output files and on user-specified options in the command line.

OUTFILE FILE ORGANIZATION	PROCESSING
SEQUENTIAL	CONVERT: • creates the output file if neither the supersede (SU) nor the append (AP) options are specified. The utility then reads records from the specified input file and writes them sequentially into the new output file. CONVERT will terminate with an error message if the specified file already exists. • supersedes an existing file if the SU option is specified. The utility reads records from the specified input file and writes them sequentially over the records already in the output file, starting with the first record position in the file. CONVERT will create the file if it does not exist. • appends records onto an existing file if the AP option is specified. The utility reads records from the specified input file and writes them sequentially into the output file, starting with the record position following the last record already in the file. CONVERT will terminate with an error message if the specified file does not exist.
RELATIVE	CONVERT reads records from the specified input file and writes them into successive record cells of the output file, beginning with cell one. CONVERT will abort if the specified file does not exist.
INDEXED	CONVERT reads each record from the specified input file, then applies the output file's record format to the data; this format is defined within the output file and is not dependent on the input file organization. The record is then inserted into the output file on the basis of the value found in the primary key field. CONVERT finally updates all of the output file's indexes (primary and any alternate keys that are defined) to reflect the presence of the record. CONVERT will abort if the output file does not exist.

INPUT FILE ORGANIZATION	PROCESSING
SEQUENTIAL	CONVERT reads the records in the sequential access mode, starting with the first record in the file.
RELATIVE	CONVERT reads the records in the sequential access mode, starting with record position 1.
INDEXED	CONVERT reads the records in the sequential access mode, following the key of reference specified in the command line (or the primary key is default).

The following restrictions apply to the use of the CONVERT utility.

1. If an existing sequential file is used for output, CONVERT requires either the supersede (SU) or the append (AP) option.

2. A relative or indexed output file must exist before CONVERT is invoked. The file may be created through an application program or a DEFINE utility.

3. When the output file organization is relative, CONVERT requires that the file contain no records. If the utility detects any records in the file, it will not execute the command line.

4. If duplicate keys are not allowed for one or more key fields of an indexed output file, CONVERT will bypass any input record that would cause such duplication to occur. In each case, the utility generates a non-fatal error message.

5. If any input record is shorter than the primary key of the indexed output file, CONVERT terminates with an error message.

6. CONVERT will not process any indexed input file with a key of reference greater than nine.

7. When both the input and output files have variable-length records, CONVERT requires that the maximum record size defined for the input file be less than or equal to the maximum record size defined for the output file. The utility will terminate with an error message if the MRS of the input file is greater than that of the output file.

8. When both the input and output files have fixed-length records, CONVERT requires either the TR or the PD option if the fixed record lengths are not equal.

9. When the input file contains variable-length records and the output file contains fixed-length records, CONVERT requires both the TR and PD switches.

The following subsections present the format and several examples of the CONVERT command line.

3.3.1 Format of the CONVERT Command Line

The CONVERT command line takes the following form:

```
CNV outfile[/options]=infile[/option]
```

where

outfile — is the file specification of the output file that is to receive the records of the input file. Wild cards are not permitted in any field of the specification.

infile — is the file specification of the input file that is the source of records to be written to the output file. Wild cards cannot appear in any field of this file specification.

Table 3-2 lists the options that can appear in a command line invoking the CONVERT utility. Detailed descriptions of each option follow.

Table 3-2
CONVERT Utility Options

FIELD	OPTION	DESCRIPTION	DEFAULT PROCESS
Help	?	Print HELP message.	No help.
Global	/ID	Identify current version of the CONVERT utility.	No id.
	/SL[:filespec]	Provide summary listing (in file, if specified).	No summary.
Outfile	/AP	Append records to sequential file.	No append.
	/FO:val	File organization.	Sequential file.
	/SU	Supersede existing sequential file.	No supersede.
	/PD[:[#]x]	Pad input records to output record length.	Abort if different lengths.
	/TR	Truncate input records to output record length.	Abort if different lengths.
	/WF	Write or read fixed control area.	Ignore VFC area.
	/LO	Follow bytes/bucket ratios when writing indexed file.	Fill buckets.
	/BL[:nnnn]	Set magnetic tape block size.	512 bytes.
Infile	/KR:n	Key of reference number.	Primary key.

Global Options

ID - Identify current version of CONVERT utility. This option may appear alone on a command line.

SL[:filespec] - Provide a summary of processing. If a file specification is not included in the option, CONVERT prints the summary on the user's terminal. If a file is specified, the utility creates it and writes the summary into it.

The summary listing includes:

- Copies of any error messages produced during execution of the utility.
- An indication that input records could not be written into an output indexed file because duplicate keys (for one or more key fields) were not permitted.

  If the listing appears on the user's terminal, the indicator is the message:

  SOME DUPLICATE RECORDS NOT WRITTEN

  If the listing is written into a file, CONVERT supplies the indicator "DUP RCD=," followed by the first 72 characters of the record that could not be written. Each record left out of the output file is shown in the summary listing file.

Outfile Options

AP - Append records to an existing sequential file. CONVERT adds records read from the input file to the end of the output file. AP and SU cannot be specified in the same command line.

FO:val - Specify organization of the output file as one of the following:

SEQ - Sequential file (default).
REL - Relative file
IDX - Indexed file

SU - Supersede existing sequential file. CONVERT utility supersedes a file in the output account with the same filename and extension. AP and SU cannot be specified in the same command line.

PD[:[#]x] - Pad records read from the input file to the output file's record length before writing them to the output file. Padding character is specified as follows:

Option	Character
PD	NULL
PD:x	x is ASCII A-Z, 0-9, or special character except #, ?, and @
PD:#x	x is octal number 000-377 (43 for #, 77 for ?, and 100 for @)

TR - Truncate records read from the input file to the output file's record length before writing them to the output file. The trailing bytes of the records are truncated.

WF - Handle variable-with-fixed-control (VFC) format records in either the input or output file. The utility processes the possible combinations as follows:

IN	OUT	PROCESSING
VFC	VFC	If VFC headers are the same size, CONVERT performs a straightforward copy process; if the headers are not identical in length, the utility terminates without converting the input file.
VFC	FIX VAR STM	The VFC header of each input record is written as the first n bytes of each output record, where n equals the size of the VFC header. 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, that is, the VFC header, of each output record, where n is the length of the VFC header. The remaining data in the input record is written into the variable portion of the output record.

If WF is not specified and one of the files contains VFC records, the fixed control area of each record is ignored.

IN	OUT	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 is read from the input file, but is written only into the variable portion of each output record.

LO - Fill the buckets of an indexed file according to the bytes-per-bucket limitations established when the file was created. See Section 3.7.4.5.

BL[:nnnn] - Control the physical block size of the output file when it is being created on magnetic tape. The default block size is 512 bytes; any size specified by nnnn must be between 18 and 8192 bytes and should be a multiple of four (otherwise, RMS-11 rounds it to the next higher multiple of four).

Infile Option

KR:n - Use the indicated key to establish the order in which records are sequentially read from the indexed input file and written to the output file, where n=0 for the primary key, n=1 for the first alternate key, and so on until n=9 for the ninth alternate key. In the absence of the KR option, a default key of 0 (primary key) is used.

3.3.2 Examples of CONVERT Command Lines

1. CNV PAYROL.DAT/FO:IDX=NAME.FIL

   CONVERT reads each record of the input file NAME.FIL and examines the contents in the primary key area (defined as an attribute of the output file PAYROL.DAT). Then the utility inserts the record into the data portion of PAYROL.DAT and updates the file's indexes.

2. CNV NAME.DAT/FO:REL=MASTER.DAT/KR:1

   CONVERT reads the indexed input file MASTER.DAT, using the first alternate key of the file to establish the sequence of access. Then the utility writes the records sequentially into the existing relative output file NAME.DAT, starting with record cell one.

3. CNV ALPHA.BAR/FO:SEQ/AP=BETA.BAR/KR:2

   CONVERT reads the indexed input file BETA.BAR, using the second alternate key of the file to establish the sequence of access. Then the utility appends the records onto the end of the existing sequential output file ALPHA.BAR.

   If the AP option had not been specified, CONVERT would have terminated with an error message because the output file existed.

4. CNV DELTA.DAT=GAMMA.DAT

   CONVERT creates the sequential file DELTA.DAT and copies the records from GAMMA.DAT into it.

5. CNV NEWPAY.DAT/PD:#43/WF/FO:REL=OLDPAY.DAT

   CONVERT reads fixed-length records from the indexed input file OLDPAY.DAT, using the primary key of the file to establish the sequence of access. Then the utility writes the records sequentially in a VFC format into the existing relative output file NEWPAY.DAT. As each record is written, the first bytes become the fixed control area (as defined in the output file's attributes); the record is also padded to the maximum record length for the file with the character #.

6. CNV MT3:INVENT.BCK/TR/BL:1024/AP=SY:INVENT.DAT/KR:0

   CONVERT reads the indexed input file INVENT.DAT, using the file's primary key to establish the sequence of access. Then the utility writes the records to magnetic tape, truncating them to the output file's record length before adding them to the end of the existing file, and formatting the tape blocks to 1024 bytes each.

3.4 DEFINE - COMMAND UTILITY FOR CREATING RMS-11 FILES

The command DEFINE utility creates RMS-11 files. The user defines or defaults the attributes of each file in the command line invoking the utility. DEFINE then stores the attributes within the newly created file. See also the interactive DEFINE utility, Section 3.7.

To create a file with the command DEFINE, the user need specify only the name and attributes of the file. Optionally, the user can specify an initial allocation quantity. DEFINE, however, does not write records into the file. The writing of the actual data contents of the file must occur subsequent to the action of the DEFINE utility. The user can employ either an application program or the CONVERT utility to write records into the file.

The attributes the user can specify when creating a file vary with the file organization selected. However, all file definitions include the following information:

- File organization
- Record Format
- Allocation quantity
- A contiguous or non-contiguous allocation
- Protection Code

Indexed files also require the specification of a primary key. Optionally, the user may define up to nine alternate key definitions, as well as key characteristics, including whether the keys may be changed during update or may contain the same value in multiple records.

NOTE

The DEFINE utility can create relative and indexed files only on disk devices.

The following subsections present the format and several examples of the DEFINE command line.

3.4.1 Format of the DEFINE Command Line

The DEFINE command line takes the following form:

```
DFN outfile[/options]
```

where

outfile — is the file specification of the file to be created. Wild cards cannot appear in any field of the specification. The file specified should not already exist in the account indicated.

Table 3-3 lists the options that can appear in a command line invoking the DEFINE utility. Detailed descriptions of the options follow the table.

Table 3-3
Command DEFINE Utility Options

FIELD	OPTION	DESCRIPTION	NOxx	DEFAULT
Global	/ID	Identify current version of the DEFINE utility.		No ID.
Outfile	/AL:n	Allocation quantity.		AL:0
	/BK:n	Number of virtual blocks per bucket.		BK:1
	/CO	Contiguous allocation.	NOCO	NOCO

TABLE 3-3 (CONT.)
DEFINE Utility Options

FIELD	OPTION	DESCRIPTION	NOxx	DEFAULT
	FO:val	File organization		FO:SEQ
	KY:n:m[:(/op:val)]	Key definition		
	PR:<prot>	Protection code		PR:<60>
	RF:val[:n]	Record format and record size		RF:VAR:0

Global Options

ID - Identify current version of DEFINE utility. This option can appear alone on a command line.

Outfile Options

AL:n - Allocation quantity. The specified value indicates the number of virtual blocks RMS-11 is to allocate for the file the first time the file is opened for writing.

BK:n - Number of virtual blocks per bucket. The user can specify this option only for relative and indexed files. A bucket can contain from 1 to 15 virtual blocks. The default value is one virtual block per bucket.

The relationship between this option and the record size value specified in the record format (RF) option is important. Since RMS-11 does not allow records to cross bucket boundaries, the user must ensure that the number of virtual blocks per bucket conforms to one of the following formulas:

1. Indexed files with fixed length records:

   Bnum=(15+(Rlen+7)*Rnum)/512

   where

   Bnum is the number of virtual blocks per bucket, ranging from 1 to 15.

   Rlen is the fixed record length.

   Rnum is the number of records per bucket.

2. Indexed files with variable length records:

   Bnum=(15+(Rmax+9)*Rnum)/512

   where

   Bnum is the number of virtual blocks per bucket, ranging from 1 to 15.

   Rmax is the maximum size of any record in the file.

Rnum is the number of records per bucket.

3. Relative files with fixed length records:

   Bnum = ((Rlen+1)*Rnum)/512

   where

   Bnum is the number of virtual blocks per bucket, ranging from 1 to 15.

   Rlen is the fixed record length.

   Rnum is the number of records per bucket.

4. Relative files with variable length records:

   Bnum = ((Rmax+3)*Rnum)/512

   where

   Bnum is the number of virtual blocks per bucket, ranging from 1 to 15.

   Rmax is the maximum size of any record in the file

   Rnum is the number of records per bucket.

5. Relative files with VFC format records:

   Bnum = ((Rmax+Fsiz+3)*Rnum(/512

   where

   Bnum is the number of virtual blocks per bucket, ranging from 1 to 15.

   Rmax is the maximum size of the data portion of any VFC record in the file.

   Fsiz is the size of the fixed control area portion of the VFC records.

   Rnum is the number of records per bucket. VFC records in a relative file bucket always occupy Rmax+Fsiz+3 bytes.

In all cases, if Bnum is not an integer, it must be rounded up to the next integer value.

CO - Contiguous allocation. The system is to attempt to allocate contiguous space on the output medium for the new file, the first time the file is opened for writing.

FO:val - File organization. The user can specify one of the following:

IDX - indexed file

REL - relative file

SEQ - sequential file

The default file organization is sequential.

The presence or absence of this option in a command line determines what additional options may or must occur in the command line. For example, if FO:IDX appears in the command line, then at least one key definition option (KY) must also appear.

KY:n:m[:(/OP:val[:val])]
- Key definition. This option can be used only for indexed files. The following considerations apply to this option:

1. The user must specify at least one key definition; i.e., the primary key definition.

2. The user can specify multiple KY options in a command string. DEFINE interprets the first such option encountered (scanning from left to right) as the primary key definition. It considers the next KY option encountered as the first alternate key definition, and so forth.

3. The user can specify a total of 10 key definitions (i.e., 1 primary and 9 alternate) for a single indexed file.

The minimum general form of a key definition is:

```
/KY:n:m
```

where

n is the location of the key field within the record. A value of 0 represents the first byte of the record, 1 the second, etc.

m is the length of the key field in bytes. Key fields can range from 1 to 255 bytes in length.

Example

```
/KY:9:16
```

This example option defines a key field that begins in the tenth byte of each record. The key field is 16 characters long.

The user can specify additional key definition information as part of the KY option. This information takes the following form:

```
/KY:n:m:(/OP:val[:val])
```

where

OP:val[:val] specifies characteristics of the key field. The following are characteristics the user can choose:

DUP - duplicate key values are allowed to occur among records in the file. This is the default for alternate keys.

CHG - key fields are allowed to change during an update operation. This is the default for alternate keys.

NODUP - duplicate key values cannot occur among records in the file. This is the default for the primary key.

NOCHG - key fields cannot be changed during an update operation. This is the default for the primary key.

The allowed combinations of key characteristics depend on the type of key (i.e., primary or alternate) being defined. Table 3-4 summarizes these combinations.

Table 3-4
Key Characteristic Combinations

Key Type	Combination			
	CHG + DUP	CHG + NODUP	NOCHG + DUP	NOCHG + NODUP
Primary	Error	Error	Allowed	Default
Alternate	Default	Error	Allowed	Allowed

When specifying the OP option, the user must use the parentheses shown in the general form above.

Example

```
/KY:0:8:(/OP:DUP:CHG)
```

This example defines a key field. In each record of the file, this key field begins in the first byte and is eight bytes long. Duplicate key values can occur among records of the file and key fields are allowed to change during update operations.

PR:<prot> - Protection code. Refer to Chapter 2.

RF:val[:n] - Record format and record size

where

val is the record format of all records in the file expressed as one of the following:

FIX - fixed size records

STM - stream format (disk sequential files only)

VAR - variable size records

VFC - variable records with fixed control (sequential and relative files only). All records will contain a default fixed control area of 2 bytes.

n

1. The actual record size in bytes when FIX is specified.

2. The maximum record size in bytes when STM, VAR, or VFC is specified. RMS-11 will check all records subsequently written into the file against this maximum size. The default record size for STM, VAR, and VFC records is 0. That is, RMS-11 is not to perform record size checks since the user has not specified a maximum record size. If the file organization is relative, the user must specify a non-zero maximum record size. If the record format is VFC, the maximum size specified by the user must not include the size of the fixed control area.

   The maximum size record allowed by RMS-11 is 16383 bytes.

3.4.2 Examples of DEFINE Command Lines

1. DFN /ID

   Displays the current version of the DEFINE utility.

2. DFN DK1:ALPHA.DAT/FO:REL/CO/RF:FIX:128

   This example shows the creation of a relative file within the current user account on the volume mounted on DK1:. DEFINE allocates the file contiguously. All records in the file will be 128 bytes in length. The default bucket size is 1 virtual block.

3. DFN TEST.INP

   DEFINE creates a sequential file named TEST.INP under the current user account on SY:. All records of the file will be variable in format with no specified maximum size.

4. DFN MASTER.FIL/FO:IDX/KY:0:20

   This example demonstrates the minimum amount of information the user must supply to create an indexed file. DEFINE creates a file named MASTER.FIL under the current user account on SY:. All records of the file will be variable in format with no specified maximum size. Each bucket in the file consists of one virtual block. Within each record, the primary key field begins in the first byte and is 20 bytes long.

5. DFN TRANS.DAT/FO:IDX/BK:4/RF:FIX:112/KY:0:12 -
   DFN> /KY:19:6/KY:29:4

   This command line creates indexed file TRANS.DAT under the current user account on SY:. Each bucket of the file contains 4 virtual blocks. All records of the file will be in fixed format and 112 bytes in length. Each record contains three key fields.

   Primary key - begins in the first byte of each record and is 12 bytes long. By default, duplicate key values are not allowed and key values cannot be changed during update operations.

   First alternate key - begins in the twentieth byte of the record and is 6 bytes long. By default, duplicate key values are allowed and key values can be changed during update operations.

   Second alternate key - begins in thirtieth byte of the record and is 4 bytes long. By default, duplicate key values are allowed and key values can be changed.

3.5 DISPLAY - DISPLAYING FILE ATTRIBUTES AND THE CONTENTS OF CONTAINER FILES

The DISPLAY utility performs either of two functions:

1. It lists the attributes of a single file or a group of files. The user can direct the utility to place the attribute listing in a specified output file or, by default, obtain the listing at the terminal.

2. The DISPLAY utility can be used to list the individual files in backup format contained within a container file on magnetic tape. Again, the user can direct the listing to a specified output file or obtain the listing at the terminal.

When invoked for the purpose of listing attributes of one or more files, DISPLAY provides the following:

- Filename and extension
- Creation date and time
- File organization
- Protection code
- Allocation information

- Last access date and time
- Record format
- Record size
- Primary and alternate key definitions (for indexed files)
- Bucket size (for relative and indexed files)

The following subsections present the format and several examples of the DISPLAY command line.

3.5.1 Format of the DISPLAY Command Line

The DISPLAY command line takes the following form:

```
DSP [outfile=]infile[/options][,infile[/options]...]
```

where

outfile — is the file specification of a file to be created by the DISPLAY utility and contain the attributes of the specified input file(s) or a list of the contents of the specified container file(s). If the user does not specify this field, DISPLAY will list the appropriate information at the terminal.

infile — is the file specification of either:

- Any type of file whose attributes are to be listed by the DISPLAY utility.
- A backup container file on magnetic tape whose contents are to be listed by the utility. The user must include the BP option (see below) to indicate that the files are container files.

Wild cards are permitted in any field of the infile specification.

Table 3-5 lists the options that can appear in a command line invoking the DISPLAY utility. Detailed descriptions of each option follow.

Table 3-5
DISPLAY Utility Options

FIELD	OPTION	DESCRIPTION	DEFAULT PROCESS
Help	?	Print HELP message.	No help.
Global	/ID	Identify current version of DISPLAY utility.	No id.
	/BP	List contents of backup container file.	Disk file.
	/FU	Display all indexed file attributes.	Basic display only.

Global Options

ID - Identify current version of DISPLAY utility. This option can appear alone on a command line.

BP - List contents of magtape container file (refer to BACKUP utility description for explanation of container files). The utility lists the names of all backup files contained within the specified container file. All input files specified in the command line must be container files on magnetic tape.

FU - List all attributes of RMS-11 indexed files, including complete definition and allocation information for individual keys, in addition to the basic data file attributes shown when the FU option is not included.

3.5.2 Examples of DISPLAY Command Lines

1. DSP FILE.DAT

   DISPLAY lists the attributes of the file FILE.DAT on the user's terminal.

2. DSP ATT.LST=*.*/FU

   DISPLAY creates the file ATT.LST and writes a full listing of attributes of all files in the default account into it.

3. DSP MT0:ALPHA.BKP,BETA.BKP/BP

   DISPLAY lists on the user's terminal the names of all files within two container files ALPHA.BKP and BETA.BKP on the tape volume mounted on MT0:.

3.5.3 Examples of DISPLAY Lists

1. For sequential files:

```
dvn:[prj,pgm]file-name.ext       FILE ORGANIZATION:     SEQUENTIAL
CREATED: dd-mon-year hh:mm  [REVISED: dd-mon-year hh:mm]
FILE PROTECTION:          nn
RECORD FORMAT:            format=length
RECORD ATTRIBUTES:        [character control] [NO SPANNING]
FILE ATTRIBUTES:
        ALLOCATION=nnnn EXTEND QUANTITY=nnnn
                        [CONTIGUOUS]
```

2. For relative files:

```
dvn:[prj,pgm]file-name.ext       FILE ORGANIZATION:     RELATIVE
RMS VERSION NUMBER:     nnn
CREATED: dd-mon-year hh:mm  [REVISED: dd-mon-year hh:mm]
FILE PROTECTION:        nn
RECORD FORMAT:          format=length
                        MAXIMUM RECORD NUMBER=nnnn
RECORD ATTRIBUTES:      [character control]
FILE ATTRIBUTES:
        ALLOCATION=nnnn EXTEND QUANTITY=nnnn      BUCKET SIZE=nnnn
                        [CONTIGOUS]
```

3. For indexed files:

```
dvn:[prj,pgm]file-name.ext       FILE ORGANIZATION:     INDEXED
RMS VERSION NUMBER:     nnn
CREATED: dd-mon-year hh:mm  [REVISED: dd-mon-year hh:mm]
FILE PROTECTION:        nn
RECORD FORMAT:          format=length
RECORD ATTRIBUTES:      [character control]
FILE ATTRIBUTES:
        ALLOCATION=nnnn EXTEND QUANTITY=nnnn      BUCKET SIZE=nnnn
                        [CONTIGUOUS]
NUMBER OF KEYS:         nnnn
```

4. For indexed files with FU option specified:

NOTE

See the interactive DEFINE utility in Section 3.7 for an explanation of key attributes shown below.

```
dvn:[prj,pgm]file-name.ext       FILE ORGANIZATION:     INDEXED
RMS VERSION NUMBER:     nnn
CREATED: dd-mon-year hh:mm  [REVISED: dd-mon-year hh:mm]
FILE PROTECTION:        nn
RECORD FORMAT:          format=length
RECORD ATTRIBUTES:      [character control]
FILE ATTRIBUTES:
        ALLOCATION=nnnn EXTEND QUANTITY=nnnn      BUCKET SIZE=nnnn
                        [CONTIGUOUS]

PRIMARY KEY:    STRING
        duplicates      changes         NULL=ccc
        ROOT VIRTUAL BLOCK NUMBER=nnnn  ROOT LEVEL=nnn
        1ST DATA BUCKET VIRTUAL BLOCK NUMBER=nnn
        KEY NAME:       ccccccccccccccccccccccccccccccc
        POSITION:       nnn
        SIZE:           nnn
        TOTAL KEY SIZE=nnn      MINIMUM RECORD LENGTH=nnn
        DATA LEVEL FILL SIZE=nnnn       INDEX LEVEL FILL SIZE=nnnn
        DATA AREA=nn    INDEX AREA=nn     LOWEST LEVEL AREA=nn

ALTERNATE KEY:  n       STRING
        duplicates      changes         NULL=ccc
        ROOT VIRTUAL BLOCK NUMBER=nnnn  ROOT LEVEL=nnn
        1ST DATA BUCKET VIRTUAL BLOCK NUMBER=nnn
        KEY NAME:       ccccccccccccccccccccccccccccccc
        POSITION:       nnn
```

```
SIZE:           nnn
TOTAL KEY SIZE=nnn       MINIMUM RECORD LENGTH=nnn
DATA LEVEL FILL SIZE=nnnn       INDEX LEVEL FILL SIZE=nnnn
DATA AREA=nn    INDEX AREA=nn     LOWEST LEVEL AREA=nn
                          .
                          .
                          .
```

where dvn:[prj,prg]file-name.ext
is the file specification of the file whose attributes are being displayed,

dd-mon-year hh:mm
is the day, month, year, and time the file was created or revised,

format
is FIXED, VARIABLE, VFC, STREAM, or ??? (undefined),

length
is an integer number of bytes set as the maximum record length,

character control is CARRIAGE RETURN or FORTRAN, and

bucket size is in blocks

3.6 RESTORE - RESTORING BACKUP FILES

The RESTORE utility reverses the action of the BACKUP utility. Whereas BACKUP creates backup copies of files, RESTORE takes those backup files as input and produces standard RMS-11 files as output. The structure, content, and attributes of these restored files are those of the original files at the time they were backed up.

RESTORE provides the user with the following capabilities:

- Explicit and Implicit File Selection
  The user may restore a single file or a collection of files with each call of the RESTORE utility. Multiple files may be specified explicitly with a string of file specifications or implicitly with wild cards. The user may select from a wild card group of files by name or on the basis of backup date.

- Choice of Restoration Account and/or Volume
  The user may restore file(s) into any account on any disk device to which the user has proper access privileges.

- Extended Diagnostic Messages
  When the utility is operating in the Query mode, RESTORE gives the user the option of continuing or terminating processing whenever two special errors occur (see below). The user may thereby ensure that the utility does not terminate the processing of a collection of files due to errors encountered in the processing of any one file. If the Query mode is explicitly disabled, RESTORE terminates immediately when it encounters the special errors.

- Summary Listing
  The user may specify that RESTORE list a summary of the files selected and restored and of all error messages. This summary may be generated at the user's terminal or it may be written into a file whose specification is included in the utility's command line.

The RESTORE utility has the following restrictions:

1. Files can be restored only on disk devices.
2. A system disk cannot be restored as a bootstrappable volume.
3. The clustersize assigned to the original file is not restored with the file. The restored file assumes the volume cluster-size.
4. A file that was contiguous when it was backed up will not be contiguous when it is restored.

The following subsections present the format and several examples of the RESTORE command line.

3.6.1 Format of the RESTORE Command Line

The RESTORE command line takes the following form:

```
RST outfile[/options]=infile[/options][,infile[/options]...]
```

where

outfile is the file specification of the file(s) to be restored by the utility. The file name and extension must be wild cards, that is, *.*; the user therefore cannot rename files as they are restored by the utility. BACKUP interprets the account specification as follows:

Account Number	Files restored to
None	Current account
[*,*]	Original accounts from which the files were backed up; the numbers are stored in the backup files.
Explicit	Specified account

infile is the file specification of a backup container file or a file in backup format. The user can use wild cards to specify multiple files.

Table 3-6 lists the options that can appear in a RESTORE command line. Detailed descriptions of each option follow the table.

Table 3-6
RESTORE Utility Options

FIELD	OPTION	DESCRIPTION	NOxx	DEFAULT
Help	?	Print HELP message.		No help.
Global	/ID	Identify current version of RESTORE utility.		No id.
	/QU	Enable Query mode.	NOQU	QU
	/SL[:filespec]	Provide summary listing (in file, if specified)		No summary.
Outfile	/SU	Supersede existing files.	NOSU	NOSU
Infile	/BD:dd-mmm-yy	Restore disk files based on date of backup.		No date checking.
	/OA:[prj,pgm]	Select files based on original account.		OA:[*,*]
	/SE:filespec or /SE:(filespec,...)	Select specified file(s) for restoration from back-up container file.		SE:*.*

Global Options

ID - Identify current version of BACKUP utility. This option can appear alone on a command line.

QU - Enable Query mode (default). When Query mode is enabled, the BACKUP utility gives the user the option of continuing or terminating processing whenever a read error occurs on an input file or the table allocated internally to monitor read errors is full.

When one of these errors occurs, the utility prints a diagnostic message specifying the type of error and the name of the file being processed. If the user answers Y (YES), RESTORE continues processing the current file and command line. If the user answers N (NO), the utility terminates processing immediately, bypassing the rest of the command line.

When the user diables the Query mode through the NOQU option, RESTORE terminates processing after it encounters one of the above errors and prints the appropriate message.

SL[:filespec] - Provide a summary of processing, including:

- The names of all files restored during the execution of the command line.
- Copies of any error messages, including diagnostic messages produced in Query mode.

If a file specification is not included in the option, RESTORE prints the summary on the user's terminal. If a file is specified, the utility creates the file and writes the summary into it.

SU - Supersede existing files. RESTORE supersedes each file in the output account with the same name and extension as a file being restored.

Infile Options

BD:dd-mon-yy - Restore files based on date of backup. The user may specify this option in combination with wild cards to identify specific files within a group of files. RESTORE will select for processing only those files that satisfy the wild card specification and that were backed up on the specified date.

The BD option cannot be used if the backup files are stored on magnetic tape. Since RESTORE cannot access the creation date of a magtape file, it generates the following error message if BD is included in a magtape infile specification:

ILLOGICAL USE OF OPTION - /BD

OA:[prj,pgm] - Select files based on original account, that is, the account from which the original files were backed up. The user can specify this option in combination with wild cards to identify specific files within a group of files. RESTORE then:

- selects for processing each file that satisfies the wild card specification
- obtains the original account specification from each selected backup file
- compares the PPN with [prj,pgm].
- restores the file only if the original account matches [prj,pgm]

SE:filespec or
SE:(filespec,...)
- Select specified file(s) for restoration from magtape container file only. This option allows the user to select explicitly one or more files from a container file. The option argument, filespec, can contain only file name and extension; either can be a wild card. Parentheses are required if more than one filespec is specified. Up to ten filespecs may be included, enclosed in parentheses and separated by commas if there is more than one. If the user specifies a container file without a select option, RESTORE uses all contained files as input to the restoration process.

3.6.2 Examples of RESTORE Command Lines

1. RST *.*=MT0:ALPHA.DAT

   RESTORE searches the volume mounted on MT0: for a container file named ALPHA.DAT. It then restores all contained files to their original form into the current user account on SY:. Since the SU option was not specified, the current user account on SY: must not contain a file with the same name and extension as a file to be restored.

2. RST *.*/SU=MT1:*.*

   RESTORE searches the volume mounted on MT1: for all container files. For each such file, it restores all contained files into the current user account on SY:. If the current user account contains a file with the same name and extension as a file being restored, the existing file is superseded.

3. RST *.*/SU/SL=DK0:*.*/BD:30-OCT-76,MT0:*.*

   RESTORE reads all files under the current user account on DK0: that were backed up on October 30, 1976, and restores them to the current user account on SY:. Then the utility restores all files within all container files on the tape mounted on MT0:. Throughout the process, the utility supersedes any existing files with the same name and extension as the file being restored. Finally, RESTORE prints a summary at the user's terminal.

4. RST [100,10]*.*/SU=MT0:*.*/SL:[100,10]REST.LST

   RESTORE restores all files within all container files on the tape on MT0: into account [100,10] on SY:, superseding existing files with the same name and extension. It also creates a file named REST.LST in account [100,10] and writes a summary of its activities into it.

5. RST *.*=MT2:MASTER.BKP/SE:(PARTS.DAT,CUST.FIL)

   RESTORE searches the volume mounted on MT2: for the container file MASTER.BKP. When this file is found, the utility searches within it for the files PARTS.DAT and CUST.FIL. These files are restored to their original form in the default account on SY:.

6. RST *.*/SU=MT0:ACCT.BKP/SE:*.DAT;*/OA:[100,11]

   RESTORE restores all files with an extension of DAT and an original account of [100,11] into the default account on SY: from container file ACCT.BKP on MT0:. The utility supersedes existing files in the default account if necessary.

7. RST DK2:[*,*]*.*;*=MT1:*.*

   RESTORE restores all files within all container files on MT1: into their original accounts on DK2:.

8. RST [200,11]*.*=MT3:A200.BAC/OA:[200,11]

   RESTORE restores all files with an original account of [200,11] in container file A200.BAC into account [200,11] on SY:.

3.7 DEFINE - INTERACTIVE UTILITY FOR CREATING RMS-11 FILES

The interactive DEFINE utility also creates RMS-11 files. It is a more powerful utility than the command DEFINE (see Section 3.4), allowing the user to control all attributes of the file being created. The user specifies file attributes by responding to requests for data and questions from the utility. The method of questioning is outlined in Figure 3-1. The figure shows the general flow of processing while the supporting text in Section 3.7.4 shows actual messages and legal responses as well as some guidance on the meaning of each attribute. The user may also get help from the utility by typing a question mark (?) in response to any question or request for data.

NOTE

The flowchart in Figure 3-1 contains circles with section numbers in them. These connectors flag the different areas of the utility's processing. Each section of text indicated by a number is designed to expand and explain the associated portion of the flowchart.

The interactive DEFINE also has the capability of building an indirect command file while the user operates the utility. This command file may be used thereafter to recreate the file and may be modified to create other similar files. See Section 3.7.4.1.

DEFINE, however, does not write records into the file. The creation of the actual data contents of the file must occur after the DEFINE utility has created the file. The user can employ either an application program or the CONVERT utility to write records into the file.

3.7.1 Utility Call

The user calls the interactive DEFINE utility with the three-character name DEF as follows:

1. Using defined CCL:

   ```
   >DEF
   ```

   The utility prints:

   ```
   DO YOU WANT TO GENERATE A COMMAND FILE FOR FUTURE USE(NO)?
   ```

2. Not using CCL:

   ```
   >RUN $RMSDEF
   ```

   The utility prints:

   ```
   DO YOU WANT TO GENERATE A COMMAND FILE FOR FUTURE USE(NO)?
   ```

See Section 3.7.4 for the complete dialog sequence.

3.7.2 Terminating the Utility

The user may terminate the interactive DEFINE utility at any time by typing a CTRL/Z (^Z). The default run-time system will assume control of the user interface.

3.7.3 File Attributes

The list below indicates the information that DEFINE will always request as well as the requests that may be made depending on specifications already typed.

1. Command File?
   a. If yes, file specification
   b. If yes, create only command file or create both RMS-11 and command file?
2. File Specification
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. Name of key
      4. Duplicate keys?
      5. Null key value?

b. Possible

1. Change keys if duplicatable?

2. Null key character if null key value

5. File Structure

1. Areas? (indexed files only)

2. Initial allocation quantity

3. Default extension quantity

4. Contiguous?

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 data for each key if areas

2. Area containing index for each key if areas

3. Area containing lowest index level if areas

7. Clustersize

8. Protection

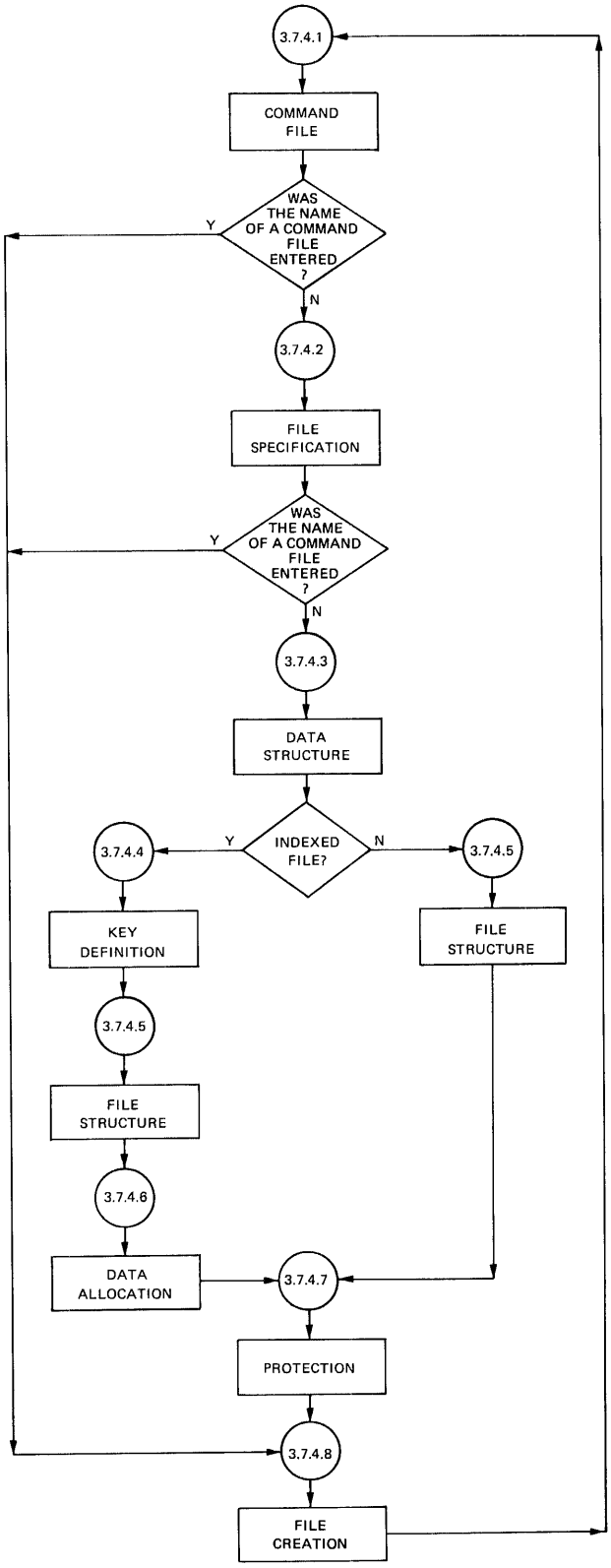
3.7.4.1
COMMAND FILE
WAS THE NAME OF A COMMAND FILE ENTERED ?
Y
N
3.7.4.2
FILE SPECIFICATION
WAS THE NAME OF A COMMAND FILE ENTERED ?
Y
N
3.7.4.3
DATA STRUCTURE
INDEXED FILE?
Y
N
3.7.4.4
3.7.4.5
KEY DEFINITION
FILE STRUCTURE
3.7.4.5
FILE STRUCTURE
3.7.4.6
DATA ALLOCATION
3.7.4.7
PROTECTION
3.7.4.8
FILE CREATION

Figure 3-1

3.7.4 Text Supporting Figure 3-1

3.7.4.1 Command File

1. The terminal prints:

   DO YOU WANT TO GENERATE A COMMAND FILE FOR FUTURE USE(NO)?

   Type one of the following:

   YES — if you want to enter a file specification for an indirect command file and have DEFINE write entries into the file as you move through the utility. Go to step 2.

   NO or (CR) — if you do not want to build a command file. Go to Section 3.7.4.2.

   @filespec — if you have already built a command file and want DEFINE to read it now and create the specified file. Go to Section 3.7.4.8.

   ID — If you want the utility to identify itself with a version number. DEFINE prints the following message.

   THIS IS THE RMS DEFINE UTILITY, VERSION n

   where n is the revision level of the utility itself.

2. The terminal prints:

   ENTER A FILE SPECIFICATION FOR THE INDIRECT FILE YOU WANT:

   Type a file specification, in the form:

   dvn:[prj,pgm]filenm.ext

   The following table shows the default value used for a file specification field when an explicit value is not supplied.

dvn:	SY:
[prj,pgm]	current PPN
filenm	none; must be specified
ext	none; must be specified

   Wild cards (asterisks) are not allowed in any field.

   See also Section 2.2.1.

3. The terminal prints:

   DO YOU WANT TO CREATE THE FILE YOU WILL BE DESCRIBING(NO)?

   Type one of the following:

   YES — if you want DEFINE to create the file being specified as well as the command file.

   NO or (CR) — if you do not want to create the RMS-11 file, only the command file that may be used to create the file at a later date.

NOTE

The command file created by DEFINE takes the form shown below. The utility follows each comment with the appropriate sequence of user entries, each on a separate line. Where the user enters only RETURN to accept the default value, DEFINE places the carriage return/line feed 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
;THE FOLLOWING QUESTIONS DEAL WITH FILE PROTECTION
```

3.7.4.2 File Specification

The terminal prints:

ENTER YOUR FILE SPECIFICATION:

Type one of the following:

filespec — if you want to create (or simulate the creation of; see Section 3.7.4.1, step 3) an RMS-11 file. Type a filespec in the form shown in Section 3.7.4.1, step 2, recognizing the defaults also shown there. Go to Section 3.7.4.3.

@filespec — if you have built a command file (see Section 3.7.4.1) and want DEFINE to use it and create the file specified. Go to Section 3.7.4.8.

3.7.4.3 Data Structure

1. The terminal prints:

   FILE ORGANIZATION (SEQ):

   Type one of the following:

   SEQ or (CR) for sequential organization

   REL for relative organization

   IDX for indexed organization

   NOTE

   If you indicated a magnetic tape device in the file specification, DEFINE does not request a file organization. Since a magtape file requires sequential organization, the utility prints:

   SINCE YOU SPECIFIED A NON-DISK DEVICE, YOUR FILE ORGANIZATION MUST BE SEQUENTIAL!!

2. The terminal prints:

   RECORD FORMAT (VAR):

   Type one of the following:

   VAR or (CR) if the records in the file will have differing, or variable, lengths

   FIX if the records in the file will have the same, or a fixed, length.

   VFC if each record in the file will have a control area with a fixed length and a data area of no standard length, that is, Variable with Fixed Control. VFC is permitted for sequential and relative file organizations only.

   STM if the records in the file will have no specific format, but are delimited only by a record terminator character. This stream format is permitted for sequential disk files only.

   UDF if the records have any format other than the ones shown above. This notation is primarily used for RMS-11 backup files.

   DEFINE will reject a format that is illegal with the file organization already specified, for instance, VFC for indexed file.

3. If you specified VFC in step 2, the terminal prints the following message; otherwise, go to step 4.

   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 and default size is two bytes. The maximum size is 256 bytes.

4. The terminal prints:

   MAXIMUM RECORD SIZE:

   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 non-zero 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.

   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. A (CR) 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)).

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 (NO)?

   Type one of the following:

   YES if you want records to cross block boundaries.

   NO or (CR) if you do not want records to span blocks.

7. The terminal prints:

   DO YOU WANT CARRIAGE RETURN CONTROL (YES)?

   Type one of the following:

   YES or (CR) 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 carriage control device (printer, terminal, and so on). See the note below and go to Section 3.7.4.4 or 3.7.4.7.

   NO 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:

   YES if you want the first byte of each record to be allocated for a FORTRAN forms control character.

   NO or (CR) if you do not want FORTRAN character control.

NOTE

If you indicated a magtape device in the file specification, at this point DEFINE requests:

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 four; if it is not, RMS-11 will round it up to the next multiple of four before writing it as an attribute. A (CR) sets the size to the default of 512 bytes.

The utility then bypasses all other processing and immediately requests protection information (see Section 3.7.4.7 below).

3.7.4.4 Key Definition

As indicated by Figure 3-1, this section applies only to indexed files.

1. The terminal prints:

   IT'S TIME TO DEFINE THE PRIMARY KEY
   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 key (see **Introduction to RMS-11**); the numbers are separated by commas.

2. The terminal prints:

   SIZE OF KEY:

   Type the decimal number of bytes in the key, that is, its length. Minimum length is one byte; maximum length is 255 bytes; there is no default.

A length must be specified for each segment of a segmented key; the numbers are separated by commas. A length must be typed for each position number specified in step 1.

3. The terminal prints:

   ENTER A NAME FOR YOUR KEY, IF YOU SO DESIRE (NONE):

   Type one of the following:

   (CR) if you do not want to specify a name for the key.

   name if you want to name the key being defined; up to 32 ASCII characters are allowed.

4. The terminal prints:

   WILL YOU ALLOW DUPLICATE KEYS?

   Type one of the following:

   YES if the file may contain more than one record with the same value for this key. Keys must be specified as duplicatable before they can be specified as changeable.

   NO if each record in the file must have a unique value for this key.

   Defaults for this question are:

   Primary key - NO

   Alternate keys - YES

STEPS 5, 6, AND 7 APPLY ONLY TO ALTERNATE KEYS.

5. If you specified YES in step 4, the terminal prints the following message; otherwise, go to step 6.

   WILL YOU ALLOW KEYS TO CHANGE (YES)?

   Type one of the following:

   YES or (CR) if this alternate key may be changed 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.

   NO if this alternate key must not change after the record is originally created.

6. The terminal prints:

   DO YOU WISH TO DEFINE A NULL KEY VALUE (NO)?

   Type one of the following:

   YES if you want the file to contain some records that cannot be accessed via this key. When RMS-11 writes a record into an indexed file, it normally

updates all indexes of the file to reflect the values found in the corresponding key fields of the record. However, if a null key value is defined for an alternate key, RMS-11 examines the contents of the key field in the record. If this field consists solely of the null key character specified, RMS-11 will not make an entry in the associated alternate index for that particular record. Go to step 7.

NO or (CR) if you do not want to specify a null value for this key. Go to step 8.

7. The terminal prints:

   O.K., ENTER YOUR NULL KEY VALUE CHARACTER

   Type one of the following:

   c the single character itself, if it is not #, ?, or @.

   #43 for the reserved character #.

   #77 for the reserved character ?.

   #100 for the reserved character @.

   #n any octal byte value (000-377), specified by n.

8. The terminal prints:

   DO YOU WANT TO DEFINE MORE KEYS (NO)?

   Type one of the following:

   YES if you want to define more keys for the file. You may define up to 255 alternate keys; however:

   - the CONVERT utility will not read higher than the ninth alternate key, that is, the KR option must be less than or equal to nine (see Section 3.3.1).
   - your application language may not support that many keys. See the appropriate **User's Guide**.

   DEFINE requests the above information for each alternate key indicated; the alternate keys are defined in order, beginning with the first alternate after the primary key has been defined.

   NO or (CR) if all keys for this file have been defined.

3.7.4.5 File Structure

1. If you specified IDX for 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:

YES if you want parts of this file to be logically different, with different attributes. The following questions will be asked for each area. For instance, you may enhance file access speeds by locating the index portion of the file in one area, with the most appropriate bucket size, full ratio, and extension quantity, while the data buckets are stored in another area with the proper attributes for them.

NO or (CR) if you want this file located in one area. DEFINE requests the following information once.

2. The terminal prints:

   ALLOCATION (0 -- IT IS SUGGESTED YOU ENTER A VALUE):

   Type a decimal number indicating the initial size of the file in blocks. The maximum file size depends on the volume on which it is being created. RMS-11 determines if there is enough space for the file as specified when it tries to create the file, returning an error if there is not enough room. A (CR) sets the value to zero. The file will be created, but it will have to be expanded before any records can be written into it. Since automatic file extension is a time-consuming procedure, a definite, but reasonable allocation quantity speeds up processing.

3. If you specified REL or IDX for file organization, the terminal prints the following message; otherwise, go to step 4.

   BUCKET SIZE (1):

   Type a decimal number indicating the number of blocks in a bucket for this area. The minimum is the number of blocks that will contain one record (according to the size specified in Section 3.7.4.3, step 4); the maximum is 15 blocks for RSTS/E; the default is one. This number determines the number of blocks read into memory during each file access operation and therefore affects speed of processing and the amount of memory a program accessing this file requires. See your operating system's **User's Guide**.

4. The terminal prints:

DEFAULT EXTENSION QUANTITY (0 - IT IS SUGGESTED YOU ENTER A VALUE):

Type a decimal number of blocks that should be added to the area each time it is extended. RMS-11 requests this number of additional blocks from the operating system; RSTS/E, however, actually extends the file to the next full cluster larger than the request. The DEQ should be a multiple of the bucket size. A (CR) sets the value to zero.

5. The terminal prints:

   DO YOU WANT A CONTIGUOUS AREA (NO)?

   Type one of the following:

   YES if you want the disk space for this area allocated in contiguous, that is, physically adjoining, blocks. If RMS-11 cannot find that much contiguous space, it will not create the file, even if sufficient non-contiguous blocks are available.

   A contiguous file cannot be extended by the RSTS/E operating system; be sure your original allocation quantity (step 2) is sufficient.

   NO or (CR) if you do not require contiguous block allocation.

6. If you answered YES in step 1 (you have an indexed file), the terminal prints the following messages; otherwise, go to the next section.

   JUST FINISHED WITH AREA NUMBER n
   DO YOU WANT TO DEFINE MORE AREAS (NO)?

   Type one of the following:

   YES if you want to specify attributes for another area of your file. Areas are numbered sequentially, starting with zero. The areas will be associated with the index and data portions of the file in the next section of the utility. Go to step 2 of this section.

   NO or (CR) if you have defined enough areas for this file. Go to step 7.

   Type a decimal number of blocks that should be added to the area each time it is extended. RMS-11 requests this number of additional blocks from the operating system; RSTS/E, however, actually extends the file to the next full cluster larger than the request. The DEQ should be a multiple of the bucket size. A (CR) sets the value to zero.

3.7.4.6 Data Allocation

As indicated by Figure 3-1, this section applies only to indexed files. The questions are asked for each key defined (see Section 3.7.4.4).

1. The terminal prints:

   AREA NUMBER FOR DATA BUCKETS FOR THIS KEY (0):

   Type an integer (0-n) indicating the area already defined (see Section 3.7.4.4) which should contain the data portion of this file.

2. The terminal prints:

   THE BUCKET SIZE IS nnn
   HOW MANY BYTES DO YOU WANT FILLED IN THE DATA BUCKET (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 RMS-11 CONVERT utility (see Section 3.3) and may be honored by MACRO-11 programs.

   The number should be set low (less than half the bucket size) if a significant number of records will be added to the file after it is originally populated. A low initial density allows data to be added without frequent bucket and file expansion.

   A (CR) sets the number to zero, indicating that buckets will be completely filled and that no free space will be available for records added during update operations.

3. The terminal prints:

   AREA NUMBER FOR INDEX BUCKETS FOR THIS KEY (0):

   Type an integer (0-n) indicating the area already defined (see Section 3.7.4.4) which should contain the index portion of this key.

4. The terminal prints:

   THE BUCKET SIZE IS nnn
   HOW MANY BYTES DO YOU WANT FILLED IN THE INDEX BUCKET (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 RMS-11 CONVERT utility (see Section 3.3) and may be honored by MACRO-11 programs.

   The number should be set low (less than half the bucket size) if a significant number of records will be added to the file after it is originally populated. A low initial density allows data to be added without frequent bucket and file expansion.

   A (CR) sets the number to zero, indicating that buckets will be completely filled and that no free space will be available for records added during update operations.

5. The terminal prints:

   AREA NUMBER FOR THE LOWEST INDEX LEVEL FOR THIS KEY (0):

   Type an integer (0-n) indicating the area already defined (see Section 3.7.4.4) which should contain the lowest index level portion of this key. If the lowest index level is stored with the data portion of the file, RMS-11 can handle most file expansion operations by accessing that one area. Additionally, the lowest index level is also the largest. If it is stored apart from the rest of the index, those blocks may be more tightly structured and thus easier to access.

3.7.4.7 Protection

1. The terminal prints:

   CLUSTERSIZE (0):

   Type a decimal number of blocks in a cluster for this file. The number is a system-oriented extension quantity and must be equal to or a power of two greater than the pack clustersize (maximum of 256). A (CR) sets the clustersize to zero, which means RMS-11 defaults to the volume clustersize; and -1 sets the clustersize to 256.

2. The terminal prints:

   PROTECTION (48):

   Type a decimal protection number for the file. See Part I of the **RSTS/E System User's Guide** for an explanation of and a guide to calculating protection numbers.

3.7.4.8 File Creation

The DEFINE utility attempts to create the file. If RMS-11 does not return an error, the utility prints:

YOUR FILE HAS BEEN CREATED!! -- filespec

If you chose to create a command file (see Section 3.7.4.1), DEFINE also prints:

YOUR FILE HAS BEEN PROCESSED AND A COMMAND FILE GENERATED!! - filespec
DO YOU WANT TO CLOSE THE INDIRECT FILE(NO)?

Type one of the following:

YES if you are finished specifying files for the command file. The utility returns to the question about command file generation (see Section 3.7.4.1).

NO or (CR) if you want to specify another RMS-11 file and you want the command file to include your input. DEFINE 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 3.7.4.1, step 3). The utility returns to the request for a file specification (see Section 3.7.4.2).

You may start the process to create another file or type CTRL/Z (^Z) to terminate the utility.

DEFINE allows two recoverable errors, shown below. All other errors result in an error message and a message that the file as specified cannot be defined. The utility returns to the file specification request to let you try again (see Section 3.7.4.2).

1. The terminal prints an error message, followed by:

THIS FILE CANNOT BE CREATED DUE TO AN ERROR IN THE FILE SPECIFICATION.
DO YOU WISH TO REENTER THE ENTIRE FILE SPECIFICATION (YES)?

Type one of the following:

YES or (CR) if you know how to correct the error and/or want to enter another file specification. The utility requests the filespec and attempts to create the file using it.

NO if you don't know how to correct the error, want to start again, and/or want to terminate the utility. DEFINE returns to the file specification request.

2. The terminal prints:

DO YOU WANT TO SUPERCEDE THE FILE (NO)?

A file with the file specification you entered already exists. Type one of the following:

YES if you want to delete the file that already exists and create the file you have specified through the utility. DEFINE deletes the existing file and attempts to create the specified one.

NO or (CR) if you do not want to supersede the existing file with the one you have just specified. DEFINE returns to the file specification request.

APPENDIX A

ERROR MESSAGES

RMS-11 utilities print a message on the user's terminal whenever an error condition is encountered during execution . The format of an error message indicates the severity of the condition encountered. A question mark (?) appearing as the first character of the message indicates a fatal error condition. Fatal error conditions cause the utility to terminate processing at the point the condition was encountered. The absense of a question mark in the first position of the message indicates a non-fatal error condition. When a non-fatal error contion occurs, the utility terminates the processing of only the file that caused the error. The utility continues processing any additional files specified in the command line.

In both types of error messages, the three character CCL name of the utility encountering the error precedes the main text of the message.

Lower case letters in the error messages listed in this appendix indicate that the utility substitutes a value in the text of the message printed at the user's terminal. The following conventions are used to indicate these substitutions.

code	- A numeric value representing an internal RMS-11 error condition. When an error message appears containing a code, the user should file an SPR with DIGITAL and document the conditions under which the error occurred.
date	- A user-specified date within the command line.
device	- A device specification.
directory	- A directory specification.
filename	- A full file specification resulting from application of defaults and/or the resolution of wild cards.
option	- An option appearing within the command line.
utl	- The three character CCL name of the utility encountering the error.
value	- An option value appearing within the command line.
vbn	- A virtual block number.

In addition to error messages containing an internal RMS-11 error code, certain fatal error conditions, if they occur, require analysis by DIGITAL personnel. These conditions are identified by the appearance of the following lines at the user's terminal:

```
************RMS-11 UTILITIES DAMAGE ASSESSMENT ROUTINE***
                        .
                        .
                        .
                      text
                        .
                        .
                        .

PLEASE DETACH AND SUBMIT WITH YOUR SPR
```

In these instances, the user should submit a copy of the displayed text to DIGITAL.

A.1 ERROR MESSAGES

?utl--CLOSE ERROR ON FILE - filename, ERROR CODE - code

Description

The specified input or output file cannot be properly closed. The indicated code refers to one of the codes listed in Appendix C of the RSTS/E System User's Guide.

Suggested User Action

Refer to Appendix C of RSTS/E System User's Guide for explanation of the indicated code.

?utl--COMMAND FILE NESTING DEPTH EXCEEDED

Description

An indirect command file contains an indirect command. RMS-11 utilities do not permit nesting of indirect commands.

Suggested User Action

Reenter the contents of one or both of the indirect command files as a sequence of individual command lines.

?utl--COMMAND LINE TOO LONG

Description

The total length of the command as typed by the user has exceeded the size of an internal buffer (e.g., the user has typed too many continuation lines).

Suggested User Action

Use several invocations of the utility to perform the functions of the original command line. If the condition occurs repeatedly, submit an SPR to DIGITAL.

?utl--CONFLICTING OPTION - option

Description

The indicated option contradicts a previous option within the same command line (scanning from left to right).

Suggested User Action

Reenter the command line without the indicated option.

?utl--DEVICE OFF LINE - device

Description

The indicated device exists on the system but the attempt to access it has been prohibited for one of the following reasons.

1. The device is not ready.
2. No volume is mounted on the device.
3. The device is currently reserved by another job.
4. The device requires privileges for ownership and the user does not have privilege.
5. The device has been disabled by the system manager.
6. The device is a keyboard line reserved for pseudo keyboard use only.

Suggested User Action

Determine the nature of the problem and take corrective action.

?utl--DEVICE WRITE PROTECTED - device

Description

The utility cannot access the indicated device for write operations.

Suggested User Action

Check the hardware condition of the indicated device. Write enable the unit.

utl--DEVICE/FILE IS FULL - device/filename

Description

The utility cannot create an output file on the indicated device because of insufficient space or the indicated file cannot be extended due to insufficient space.

Suggested User Action

Reenter the command using another device for output files or copy the indicated file to another device and retry the command. Optionally, delete unneeded files on the indicated device and reenter the original command line.

?utl--DIRECTORY NOT FOUND - filename

Description

The directory does not exist on the specified device.

Suggested User Action

Reenter the command with the correct directory specification.

?utl--DIRECTORY SYNTAX ERROR - filename

Description

The directory portion of the indicated file specification does not conform to the syntax rules.

Suggested User Action

Reenter the command line with the correct syntax.

utl--DUP RCD - string

Description

The utility could not write a record into an indexed file because duplicate key values for one or more keys in the record were not permitted. Writing the record would cause duplication. The displayed string represents the first 72 characters of the record that could not be written.

Suggested User Action

None.

utl--EMPTY INDEX INPUT FILE

Description

The utility has attempted to read records in an indexed file using an alternate index that has no entries. This condition can occur when all records written into the file were too short to contain the alternate key.

Suggested User Action

If desired, specify the primary index or another alternate index in a reentered command line.

utl--ERROR DURING WILD CARD LOOKUP - filename, RSTS/E CODE - code

Description

An error code has been returned to the utility from the RSTS/E routine that attempted to resolve wild cards in the indicated file specification. The indicated code refers to one of the codes listed in Appendix C of the RSTS/E System User's Guide.

utl--ERROR IN FILE PROLOGUE - filename

Description

The utility has encountered an error in the contents of the prologue (internal RMS-11 information within a file) of the specified file. The file is bypassed and processing continues.

Suggested User Action

Submit a dump of the file to a software support specialist.

?utl--ERROR INSERT TOO BIG, ERROR NUMBER - code

Description

The utility is unable to print the text of an error message at the user's terminal.

Suggested User Action

Submit an SPR to DIGITAL documenting the conditions under which the error occurred.

?utl--FAILURE TO CREATE CONTIGUOUS FILE

Description

The user has specified the CO option but there is insufficient contiguous space on the device to allocate the file.

Suggested User Action

Reenter the command without the CO option or delete unneeded files from the device to supply contiguous space.

?utl--FAILURE TO OPEN COMMAND FILE - filename

Description

An I/O or file system error (e.g., a privilege violation, an atempted write to a locked unit, parity error, etc.) has occurred when the utility attempted to open an indirect command file.

Suggested User Action

Determine the cause of the problem, rectify or circumvent the problem, and retry the command.

?utl--FILE ALREADY EXISTS - filename

Description

The utility has attempted to create an output file that already exists in the output account.

Suggested User Action

Reenter the command line using a new or corrected filename or delete the existing file and reenter the original command line.

?utl--FILE, filename, CORRUPTED

Description

The internal structure of an RMS-11 file is corrupted and RMS-11 cannot store or retrieve records in the file.

Suggested User Action

If a backup copy is available, use the RESTORE utility to restore the file to its condition at the time of backup.

?utl--FILE NAME SYNTAX ERROR - filename

Description

The name portion of the indicated file specification does not conform to the syntax rules.

Suggested User Action

Reenter the command line with the correct syntax.

?utl--FILE NOT AVAILABLE - filename

Description

The indicated file is being accessed for exclusive use by another job.

Suggested User Action

Periodically retry the command until the file has been released.

utl--FILE NOT FOUND - filename

Description

The indicated file was not found in the designated account and device.

Suggested User Action

Verify the file specification and reenter the command line.

?utl--FILE POSITION LOST - filename

Description

The utility has lost its position within a container file on magnetic tape while rewinding or backspacing. The error may have been caused by hardware failure.

Suggested User Action

Determine from the output account or summary listing file the extent of the processing that was completed prior to the occurrence of the error. Reenter the command line eliminating file specifications of files successfully processed. Use a new tape volume and/or tape drive.

utl--FILE PROCESSING COMPLETE - filename

Description

The indicated file was successfully processed (e.g., backed up or restored) by the utility.

Suggested User Action

None. This is an informational message.

?utl--FILE TYPE SYNTAX ERROR - filename

Description

The extension portion of the indicated file specification does not conform to the syntax rules.

Suggested User Action

Reenter the command line with the correct syntax.

utl--I/O ENCOUNTERED ON INPUT FILE - filename, VBN vbn1 TO vbn2.

CONTINUE (Y,N)?

Description

The indicated virtual blocks cannot be read from the indicated file. If the file is input to BACKUP, the utility cannot back-up the data records within the virtual blocks. Similarly, if the file is input to RESTORE, the utility cannot restore the data records within the indicated blocks of the original file.

Suggested User Action

Type Y if the utility is to continue despite the error. Type N if all processing is to be terminated.

?utl--I/O ERROR ENCOUNTERED ON OUTPUT FILE - filename

Description

One of the following conditions exists:

1. The device is not on line.
2. The device is not mounted.
3. The hardware has failed.
4. The volume is full.

Suggested User Action

Rectify the condition and reenter the command line.

?utl--I/O ERROR ON COMMAND LINE INPUT

Description

The utility is unable to read the comand line typed by the user at the terminal or a line within an indirect file.

Suggested User Action

Reenter the command line or recreate the indirect file and retry the command.

?utl--ILLEGAL DEVICE - device

Description

The indicated device does not exist.

Suggested User Action

Reenter the command line with a corrected device specification.

?utl--ILLEGAL FILE SPECIFICATION - filename

Description

The indicated file specification does not conform to the syntax rules.

Suggested User Action

Reenter the command line with the correct syntax.

?utl--ILLEGAL INPUT OPTION - option

Description

The indicated option is not a legal input file option.

Suggested User Action

Reenter the command line with the correct option or without the indicated option.

?utl--ILLEGAL NUMBER OF INPUT FILES

Description

Multiple input files have been specified by the user for a utility that accepts only a single input file (e.g., CONVERT).

Suggested User Action

Reenter the command specifying a single input file.

?utl--ILLEGAL NUMBER OF OUTPUT FILES

Description

Multiple output files have been specified by the user for a utility that accepts only a single output file (e.g., DEFINE).

Suggested User Action

Reenter the command specifying a single output file.

?utl--ILLEGAL OPTION VALUE - option

Description

The indicated option contains an illegal option value.

Suggested User Action

Reenter the command line with the correct option value.

?utl--ILLEGAL OUTPUT OPTION - option

Description

The indicated option is not a legal output file option.

Suggested User Action

Reenter the command line with the corrct option or without the indicated option.

utl--ILLEGAL RMS FILE ORGANIZATION - value

Description

The indicated option value does not represent a legal RMS-11 file organization.

Suggested User Action

Reenter the command with the correct option value.

?utl--ILLEGAL RMS RECORD FORMAT - value

Description

One of the following conditions exists:

1. The indicated option value does not represent a legal RMS-11 record format.

2. The indicated record format is not permitted within a particular file organization. For example, VFC format records are not permitted in indexed files.

Suggested User Action

Reenter the command with the correct record format.

?utl--ILLEGAL RMS RECORD SIZE

Description

One of the following conditions exists:

1. The user has specified a record size greater than 16000.

2. The user has failed to specify a maximum record size greater than zero when creating a relative file with variable format records.

3. The specified record size is too large for the buckets of the file.

Suggested User Action

Reenter the command with the correct record size.

?utl--ILLEGAL USE OF /AP OR /SU WITH FILE ORGANIZATION

Description

The user has specified Append (/AP) or Supersede (/SU) for a non-sequential file.

Suggested User Action

Reenter the command line without the /AP or /SU option.

?utl--ILLEGAL USE OF WILD CARDS IN SUMMARY LISTING FILE

Description

The argument of an SL option is a file specificatin containing wild cards. Wild cards are not permitted in this context.

Suggested User Action

Reenter the command line without wild cards in the summary listing file specification.

?utl--ILLEGAL USE OF WILD CARDS ON INPUT FILE

Description

The utility (e.g., CONVERT or COPY with the AP or ME option) does not permit wild cards in an input file specification.

Suggested User Action

Reenter the command line without wild cards in the input file specification.

?utl--ILLEGAL USE OF WILD CARDS ON OUTPUT FILE

Description

The utility (e.g., COPY with the AP option, DEFINE) does not permit wild cards in the output file specification.

Suggested User Action

Reenter the command line without wild cards in the output file specification.

?utl--ILLOGICAL DEVICE - device

Description

The indicated device is not permitted in the context of the command line. For example, the user cannot DEFINE and indexed file on magnetic tape.

Suggested User Action

Reenter the command line with an appropriate device specification.

?utl--ILLOGICAL USE OF OPTION - option

Description

The indicated option is illogical in the context of the command. For example, the SU option (Supersede) is specified for files on magnetic tape or a key definition option (KY) is specified for a sequential file.

Suggested User Action

Reenter the command line with correct options.

utl--INCOMPLETE BACKUP FILE REMAINS - filename

Description

The error condition described by a prior error message has caused the termination of processing before the indicated backup file could be completed and properly closed.

Suggested User Action

Type Y if the utility is to delete the indicated output file. Type N if the utility is to attempt to close the file and thereby preserve the data in the file.

?utl--INCORRECT FILE ORGANIZATION - filename

Description

The organization of the file does not match the organization specified by the user.

Suggested User Action

Reenter the command line with the correct file organization.

?utl--INPUT AND OUTPUT RECORD FORMATS DO NOT CORRESPOND

Description

The user is attempting to write records from one file to another. However, the input file records are variable and the output file records are fixed.

?utl--INPUT AND OUTPUT RECORD SIZES DO NOT CORRESPOND

Description

The user is attempting to write records from one file to another. However, one of the following conditions exists:

1. Both files have fixed format records but the fixed size differs.

2. Both files have variable format records but the maximum size of the input file is greater than the maximum size of the output file.

Suggested User Action

Redefine the output file and retry the command.

utl--INPUT FILE IS NOT BACKUP FILE - filename

Description

The utility requires that the input file be a backup file. The user has specified a file that is not in backup format. For example, a file not in backup format is specified as input to RESTORE.

Suggested User Action

Reenter the command line with the correct file specification.

utl--INTEGRITY CHECK TABLE FULL. CONTINUE (Y,N)?

Description

The table allocated internally by the utility to monitor input read errors has exceeded its capacity.

Suggested User Action

Type Y if the utility is to continue processing. Type N if the utility is to terminate processing.

?utl--INVALID /KR VALUE

Description

The user-specified key of reference indicates that an indexed file has more than 10 keys defined. The utility cannot process an indexed file with more than 10 defined keys.

Suggested User Action

To process an indexed file with more than 10 keys, the user must write an application program to perform the desired action.

?utl--KEY OPTION COMBINATION CHG-NODUP ILLEGAL

Description

The user is attempting to create an indexed file with an illegal combination of characteristics for one or more keys.

Suggested User Action

Reenter the command with correct key characteristics.

?utl--/KR NOT ALLOWED FOR SEQUENTIAL OR RELATIVE FILE

Description

The /KR option is permitted for indexed files only. The user has either specified the wrong file or specified /KR in the wrong context.

Suggested User Action

Reenter the command with the correct filename or without the /KR option.

?utl--LABEL ERROR - dev

Description

The specified magnetic tape volume does not have ANSI labels.

Suggested User Action

Mount the appropriate ANSI-labeled volume.

utl--MISSING OPTION VALUE - option

Description

The indicated option requires a user-specified option value.

Suggested User Action

Reenter the command specifying a value for the indicated option.

?utl--NO KEY DECLARED FOR INDEXED FILE

Description

The user has attempted to create an indexed file without defining a primary key.

Suggested User Action

Reenter the command line with an appropriate primary key definition.

?utl--NO RENAME ALLOWED

Description

The BACKUP and RESTORE utilities require that output file names and extensions be the same as the corresponding input file names and extensions. Only magnetic tape container file can have a name that does not correspon to an input name.

Suggested User Action

Reenter the command line specifying wild cards for the name and extension components of the output file specification.

utl--NO SUCH FILE

Description

No files in the account correspond to the wild cards of a file specification.

Suggested User Action

Obtain a listing of the files in the desired account. Reenter the command with the desired file specification.

?utl--NO SUCH KEY FOR FILE - value

Description

The specified key of reference value represents a non-existent key in an indexed file.

Suggested User Action

Reenter the command with a correct key of reference value.

?utl--NOT A DIRECTORY DEVICE - device

Description

The user has issued a directory-oriented command for a device (such as a printer) that does not have directories (accounts).

Suggested User Action

Reenter the command line without specifying an account.

?utl--NOT A SHARABLE DEVICE - device

Description

The command line as typed by the user would require the sharing of a non-sharable device. For example, the BACKUP utility cannot create a backup container file and a summary listing file on the same magnetic tape during a single invocation.

Suggested User Action

Reenter the command line with a corrected device specification.

?utl--NOT ENOUGH MEMORY AVAILABLE

Description

The amount of memory required by the indicated utility is greater than the job's private memory size maximum.

Suggested User Action

Submit an SPR to DIGITAL documenting the conditions under which the condition occurred.

?utl--ONLY [*,*] or [X,Y] is LEGAL AS DESTINATION

Description

The user has specified an output file specification with an account other than [*,*] or [X, Y] (e.g., [200,11]).

Suggested User Action

Reenter the command line with an explicit account specification of [*,*].

utl--OPEN ERROR ON filename, RSTS/E ERROR CODE - code

Description

The utility is unable to open the specified file. The indicated code refers to one of the codes listed in Appendix C of the RSTS/E System User's Guide.

Suggested User Action

Refer to Appendix C of the RSTS/E System User's Guide for explanation of the indicated code.

?utl--OUTPUT FILE MISSING

Description

The user has failed to type an output file specification in the command line to the indicated utility.

Suggested User Action

Reenter the command line with an output file specification.

?utl--OUTPUT VOLUME CORRUPT - filename

Description

While accessing the indicated file, the utility has determined that the output volume is corrupt.

Suggested User Error

Mount another volume or utilize a different device and reenter the command with appropriate device specifications.

?utl--PRIMARY KEY CAN NOT CHANGE

Description

The user has specified that the primary key of an indexed file can change. RMS-11 does not permit this characteristic for primary keys.

Suggested User Action

Reenter the command without the change characteristic for the primary key.

?utl--PRIVILEGED VIOLATION - filename

Description

The user does not have the privileges necessary to access the indicated file.

Suggested User Action

Have the owner of the file change its privilege specification.

utl--READ ERROR, INTEGRITY CHECK TABLE AND REWRITE DATA MAY HAVE BEEN LOST

Description

The utility has encountered an error while attempting to read internal data maintained in a backup file or container file.

Suggested User Action

Retry the command. If the same error occurs, check summary listing file created at the time the backup or container file was created. Determine from this file which file or files cannot be completely restored.

utl--READ ERROR ON FILE ATTRIBUTES - filename

Description

The volume is corrupted or the user does not have the necessary privileges to access the indicated file.

Suggested User Action

For disk devices, use the CLEAN operation in UTILITY.

utl--READ ERROR ON FILE PROLOGUE, filename

Description

The utility is unable to read the prologue (internal RMS-11 information within a file) of the specified file. The file is bypassed and processing continues.

Suggested User Action

Reenter the command specifying the subject file only. If the same error occurs, the indicated file cannot be properly accessed on the device. Use the RESTORE utility to retrieve a new copy of the file.

utl--READ ERROR OR INCONSISTENT DATA. MAY HAVE LOST FILES.

Description

The utility has encountered an error while reading a backup or container file.

Suggested User Action

Retry the command. If the same error occurs, check summary listing file created at the time the backup or container file was created. Determine from this file which file or files may have been lost.

?utl--RELATIVE OUTPUT FILE NOT EMPTY

Description

The utility requires that a relative output file not contain any records. While verifying that the output file is empty, the utility has found one or more records.

Suggested User Action

Use the DEFINE utility to create an empty relative file.

utl--REQUESTED RECORD LOCKED. CONTINUE (Y,N)?

Description

A file being accessed by the utility is being simultaneously accessed by another user. Within the file, a particular record has been locked by the second user, possibly for the purposes of changing the contents of the record.

Suggested User Action

Type Y if the utility is to wait for the record to be released. Type N to terminate the utility. Since this condition can occur repeatedly while the file is being accessed simultaneously by another user, it may be desirable to terminate the utility and retry the command at a later point in time.

?utl--REWIND OR SPACE ERROR ON FILE,-filename

Description

The utility has encountered an error while rewinding or backspacing on magnetic tape.

Suggested User Action

Retry the command. If the condition occurs again, mount the volume on another drive and retry the command.

?utl--SELECT ERROR ON dev

Description

One of the following conditions exists:

1. The device is not on-line.
2. The device is not mounted.
3. The hardware has failed.

Suggested User Action

Rectify the condition and reenter the command line.

utl--SOME DUPLICATE RECORDS NOT WRITTEN

Description

The utility could not write one or more records into an indexed file because duplicate key values for one or more keys were not permitted. Writing such records would cause duplication.

Suggested User Action

None.

?utl--SYNTAX ERROR - string

Description

The format of the command line does not conform to the syntax rules. The utility prints the command line from the point of the error to the end of the line.

Suggested User Action

Refer to the appropriate utility description in Chapter 3. Reenter the command line with the correct syntax.

?utl--TI LUN ASSIGNMENT ERROR

Description

The utility is unable to communicate with the user's terminal.

Suggested User Action

Consult with a DIGITAL Software Support specialist.

?utl--TOO MANY KEYS - filename

Description

The specified indexed file has more than 10 keys defined. The utility cannot process an indexed file with more than 10 defined keys.

Suggested User Action

To process an indexed file with more than 10 keys, the user must write an application program to perform the desired action.

?utl--TOO MANY SELECTIVE FILE SPECIFICATIONS

Description

More than 10 file specifications appear as option values of the SE option.

Suggested User Action

Reenter the command line specifying no more than 10 file specifications as option values of the SE option. Use multiple invocations of the utility to select additional files.

utl--UNABLE TO RESTORE DATE ATTRIBUTE - filename

Description

The utility was unable to restore the file with one or more of its original date attributes (e.g., creation date, revision date).

Suggested User Action

Use the DISPLAY utility to determine date attributes of restored file.

utl--UNCORRECTABLE ERROR, VBN vbn1 TO vbn2, REASON, INPUT ERROR

Description

During automatic retry of input read errors, the utility has encountered an uncorrectable error.

Suggested User Action

Maintain record of ranges of virtual blocks that were erroneous.

?utl--USE OF RMS FEATURE NOT PURCHASED

Description

The user has specified an indexed facility but the installation does not provide the RMS-11K feature.

Suggested User Action

Reenter the command using only non-indexed facilities.

?utl--WRITE ERROR ON ATTRIBUTES OF FILE - filename

Description

The volume is corrupted or the user does not have the necessary privileges to write the file.

Suggested User Action

Verify access to file. If access is correct, run the CLEAN operation in UTILTY to determine if volume is corrupted.

?utl--WRITE ERROR ON CREATE OF OUTPUT FILE - filename

Description

One of the following conditions exists:

1. The device is not on line.
2. The device is not mounted.
3. The hardware has failed.
4. The volume is full.

Suggested User Action

Rectify the condition and reenter the command line.

?utl -- WRITE ERROR ON INTEGRITY CHECK TABLE ON OUTPUT FILE - filename

Description

The utility is unable to write internal data integrity checking tables in the output backup file.

Suggested User Action

If the output medium is magnetic tape, use a different tape volume and retry the command. 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.

APPENDIX B

RMS-11 UTILITY HELP MESSAGES

B.1 BACKUP UTILITY

```
                RMS-11 BACKUP UTILITY
                ---------------------
FUNCTION: CREATE BACKUP COPIES OF FILES
FORMAT:   BCK OUTFILE[/OPTIONS]=INFILE[/OPTIONS][,INFILE[/OPTIONS]..]

GLOBAL OPTIONS:
 ?  OR HELP     TYPE HELP MESSAGES
 ID             IDENTIFY CURRENT VERSION NUMBER
 QU             ENABLE QUERY MODE
 SL:FILESPEC    PROVIDE SUMMARY LISTING FILE

OUTFILE OPTIONS
 RW             REWIND MAGNETIC TAPE BEFORE WRITING
 SU             SUPERSEDE EXISTING SEQUENTIAL FILE

INFILE OPTIONS:
 CD:DD-MMM-YY   BACKUP FILES BASED ON CREATION DATE
 RD:DD-MMM-YY   BACKUP FILES BASED ON REVISION DATE
```

B.2 CONVERT UTILITY

```
                RMS-11 CONVERT UTILITY
                ----------------------
FUNCTION: MOVE RECORDS BETWEEN FILES
FORMAT:   CNV OUTFILE[/OPTIONS]=INFILE[/OPTIONS]

GLOBAL OPTIONS:
 ? OR HELP      TYPE HELP MESSAGES
 ID             IDENTIFY CURRENT VERSION NUMBER
 SL:FILESPEC    PROVIDE SUMMARY LISTING FILE

OUTFILE OPTIONS:
 AP             APPEND RECORDS TO SEQUENTIAL FILE
 FO:VAL         FILE ORGANIZATION
 SU             SUPERSEDE EXISTING SEQUENTIAL FILE
 TR             TRUNCATE RECORDS
 PD:CHAR        PAD RECORD WITH ASCII CHARACTER
 PD:#OCTAL      PAD RECORD WITH OCTAL VALUE
 WF             WRITE FIXED HEADER OF VFC RECORD
 LO             ENFORCE BUCKET LOADING FACTOR

INFILE OPTIONS:
 KR:N           KEY OF REFERENCE NUMBER
```

B.3 INTERACTIVE DEFINE UTILITY

Help for "DO YOU WANT TO GENERATE A COMMAND FILE FOR FUTURE USE(NO)?"

```
                TYPE A QUESTION MARK (?) FOR HELP AT ANY TIME.
  ALL NUMBERS WILL BE DECIMAL, UNLESS EXPLICITLY SPECIFIED OTHERWISE.
DEFAULTS WILL BE SHOWN IN PARENTHESES.  TYPE CARRIAGE RETURN TO GET THEM.
IF YOU ANSWER YES TO THIS QUESTION, A FILE CONTAINING YOUR CORRECT
RESPONSES TO THE UPCOMING QUESTIONS, WILL BE GENERATED FOR YOUR FUTURE U
SE
```

Help for "DO YOU WANT TO CREATE THE FILE YOU WILL BE DESCRIBING (NO)"

```
IF YOU SAY YES, THE FILE YOU ARE GOING TO DESCRIBE WILL ACTUALLY BE
CREATED.  IF YOU SAY NO, IT WON'T BE. IN EITHER CASE, A COMMAND FILE
WILL BE GENERATED FOR YOUR FUTURE USE.
```

Help for "ENTER YOUR FILE SPECIFICATION:"

```
THE FILE SPECIFICATION SHOULD BE ENTERED IN THE FOLLOWING FORMAT:
            DEV:[GRP,MEMBR] FILENAME.EXTENSION
DEFAULTS ARE: DEV = SY0, UIC = YOUR OWN
THE GROUP AND MEMBER NUMBERS OF THE UIC ARE OCTAL
```

Help for "FILE ORGANIZATION (SEQ):"

```
ENTER SEQ FOR SEQUENTIAL (DEFAULT), REL FOR RELATIVE, OR IDX FOR INDEXED
```

Help for "RECORD FORMAT (VAR:)"

```
ENTER VAR FOR VARIABLE OR FIX FOR FIXED.

ENTER VAR FOR VARIABLE (DEFAULT), FIX FOR FIXED, STM FOR STREAM,
VFC FOR VARIABLE-WITH-FIXED-CONTROL-AREA, OR UDF FOR UNDEFINED.

ENTER VAR FOR VARIABLE (DEFAULT), FIX FOR FIXED
VFC FOR VARIABLE-WITH-FIXED-CONTROL-AREA.
```

Help for "SIZE OF FIXED CONTROL AREA(2):"

```
ENTER A VALUE REPRESENTING THE SIZE, IN BYTES, OF THE FIXED
CONTROL AREA OF EACH RECORD.  THE NUMBER MUST BE BETWEEN 1 AND 255.
A VALUE OF ZERO IS EQUIVALENT TO THE DEFAULT OF 2 BYTES.
```

Help for "MAXIMUM RECORD SIZE:"

ENTER THE MAXIMUM SIZE OF RECORDS THAT YOU WILL ALLOW.
A VALUE OF ZERO MEANS THAT RMS WILL NOT LIMIT THE SIZE OF THE RECORD.

Help for "MAXIMUM NUMBER OF RECORDS(0):"

SPECIFY A VALUE REPRESENTING THE HIGHEST NUMBERED RECORD
THAT CAN EVER BE WRITTEN TO THE FILE. THE DEFAULT IS ZERO,
WHICH MEANS THAT NO CHECKS WILL BE PERFORMED.

Help for "WILL YOU ALLOW RECORDS TO CROSS BLOCK BOUNDARIES (NO)?"

ENTERING YES WILL ALLOW RECORDS TO CROSS VIRTUAL BLOCK BOUNDARIES.

Help for "DO YOU WANT CARRIAGE RETURN CONTROL (YES)?"

IF YOU SAY YES, EACH RECORD WILL BE PRECEDED BY A LINE FEED AND
FOLLOWED BY A CARRIAGE RETURN ON OUTPUT TO A CARRIAGE CONTROL DEVICE.

Help for "DO YOU WANT FORTRAN CHARACTER CONTROL (NO)?"

IF YOU SAY YES, THIS MEANS THAT THE 1ST BYTE OF EACH
RECORD CONTAINS A FORTRAN CARRIAGE CONTROL CHARACTER.

Help for "MAGTAPE BLOCK SIZE (512):"

YOU MAY SPECIFY THE SIZE (IN BYTES) OF THE
PHYSICAL BLOCKS ON THE MAGTAPE. A VALUE OF 0 IS
THE DEFAULT AND IS UNDERSTOOD TO BE THE SYSTEM
DEFAULT OF 512. THE VALUE MUST BE BETWEEN 18 AND 8192.

Help for "POSITION OF KEY:"

ENTER A VALUE REPRESENTING THE STARTING POSITION OF THE KEY.
A KEY CAN START IN POSITION 0, WHICH IS THE 1ST BYTE OF THE RECORD.

IF THE KEY IS SEGMENTED, YOU MAY ENTER UP TO 8 POSITIONS.
IF YOU SPECIFY MORE THAN 1 POSITION, THE POSITIONS MUST BE ENCLOSED
IN PARENTHESES AND SEPARATED BY COMMAS.

Help for "SIZE OF KEY:"

ENTER A VALUE REPRESENTING THE SIZE OF THE KEY.

IF THE KEY IS SEGMENTED, YOU MAY ENTER UP TO 8 SIZES, PROVIDING YOU HAVE ALREADY SPECIFIED AN EQUAL NUMBER OF POSITIONS FOR THE KEY. IF YOU SPECIFY MORE THAN 1 SIZE, THE SIZES MUST BE ENCLOSED IN PARENTHESES AND SEPARATED BY COMMAS. THE COMBINED TOTAL OF THE SIZES MUST BE LESS THAN 256.

Help for "ENTER A NAME FOR YOUR KEY, IF YOU SO DESIRE(NONE):"

YOU MAY OPTIONALLY ASSOCIATE AN ALPHA-NUMERIC CHARACTER STRING WITH THE KEY YOU HAVE JUST DEFINED. RMS WILL NEVER EXAMINE THIS STRING. IF YOU SPECIFY ONE, HOWEVER, RMS WILL STORE IT IN THE FILE WITH THE ASSOCIATED KEY DEFINITION. THE MAXIMUM LENGTH IS 32 CHARACTERS.

Help for "WILL YOU ALLOW DUPLICATE KEYS?"

IF YOU ALLOW DUPLICATES, RECORDS IN THE FILE MAY HAVE THE SAME VALUES IN THE KEY FIELD. OTHERWISE, DUPLICATES ARE NOT ALLOWED. DEFAULT IS NO FOR PRIMARY, YES FOR ALTERNATE KEYS. IN ORDER TO PERMIT ALTERNATE KEYS TO CHANGE, YOU MUST NOW PERMIT THEM TO BE DUPLICATED.

Help for "WILL YOU ALLOW KEYS TO CHANGE(YES)?"

IF YOU SAY YES, YOU ARE ALLOWING KEY VALUES WITHIN RECORDS TO BE CHANGED DURING AN UPDATE OPERATION. THIS IS ONLY ALLOWED FOR ALTERNATE KEYS WHICH CAN BE DUPLICATED.

Help for "DO YOU WISH TO DEFINE A NULL KEY VALUE(NO)?"

IF YOU WANT A NULL KEY VALUE, SPECIFY A SINGLE CHARACTER. IF A RECORD CONTAINS ONLY THAT CHARACTER THROUGHOUT THE ENTIRE KEY FIELD, RMS WILL NOT CREATE AN ENTRY FOR THAT RECORD IN THE ASSOCIATED INDEX.

Help for "O.K., ENTER YOUR NULL KEY VALUE CHARACTER:"

FOR A NONPRINTING CHARACTER, TYPE A '#' FOLLOWED BY AN OCTAL NUMBER. NOTE THAT '@' IS CONSIDERED NONPRINTING, SINCE IT IS THE SYMBOL FOR AN INDIRECT FILE. THEREFORE, IT MUST BE ENTERED AS '#100'. LIKEWISE, A '#' MUST BE ENTERED AS '#43'. ALSO, A '?' MUST BE ENTERED AS '#77' SINCE IT IS THE SYMBOL FOR HELP.

Help for "DO YOU WANT TO DEFINE MORE KEYS(NO)?"

FOR AN INDEXED FILE YOU MUST SPECIFY 1 OR MORE KEYS. YOU CAN THEN REFERENCE THE RECORDS BY KEY.

Help for "DO YOU WANT TO DEFINE AREAS(NO)?"

YOU MAY, IF YOU WISH, DEFINE FILE AREAS. THIS WILL GIVE YOU MORE CONTROL OVER THE INTERNAL STRUCTURE OF AN INDEXED FILE. FOR EACH AREA, YOU CAN ASSIGN AN INITIAL ALLOCATION QUANTITY, A BUCKET SIZE, A DEFAULT AREA EXTENSION QUANTITY, AND A STARTING LOCATION. YOU MAY ALSO SPECIFY THAT YOU WANT A SPECIFIC AREA TO BE CONTIGUOUS. LATER YOU MAY ASSOCIATE THE DATA BUCKET LEVEL, THE INDEX BUCKET LEVEL, AND THE LOWEST INDEX BUCKET LEVEL OF YOUR KEYS WITH AREAS. THE ONLY RESTRICTION IS THAT THE LOWEST INDEX BUCKET LEVEL OF ANY KEY CANNOT BE ASSOCIATED WITH AREA 0.

Help for "ALLOCATION (0 -- IT IS SUGGESTED YOU ENTER A VALUE):

ENTER THE NUMBER OF VIRTUAL BLOCKS YOU WANT IN THE FILE/AREA.

Help for "BUCKET SIZE(1):"

ENTER THE NUMBER OF VIRTUAL BLOCKS CONTAINED IN EACH BUCKET OF THE FILE/AREA.

Help for "DEFAULT EXTENSION QUANTITY (0 -- IT IS SUGGESTED YOU ENTER A VALUE):"

THE DEFAULT EXTENSION QUANTITY IS THE NUMBER OF VIRTUAL BLOCKS TO BE USED WHEN RMS11 MUST AUTOMATICALLY EXTEND YOUR FILE/AREA. FOR THE SAKE OF SPEED, IT IS SUGGESTED THAT YOU SPECIFY A NONZERO VALUE. THE DEFAULT EXTENSION QUANTITY MUST BE A MULTIPLE OF THE BUCKET SIZE.

Help for "DO YOU WANT A CONTIGUOUS AREA (NO)?"

IF YOU WANT YOUR FILE/AREA TO BE CREATED PHYSICALLY CONTIGUOUS, SAY YES.

Help for "DEFAULT EXTEND QUANTITY FOR YOUR FILE (0):"

ENTER A VALUE TO BE THE DEFAULT EXTEND QUANTITY FOR YOUR FILE. THIS VALUE WILL BE USED AS THE "DEFAULT" FOR ANY AREA FOR WHICH YOU HAVE NOT ALREADY DEFINED A NON-ZERO VALUE.

Help for "AREA NUMBER FOR DATA BUCKETS FOR THIS KEY (0):"

YOU MAY ASSIGN THE BUCKETS OF THE DATA LEVEL OF THE INDEX DEFINED BY THIS KEY TO ONE OF THE AREAS YOU JUST DEFINED. WHEN DESCRIBING THE PRIMARY KEY, THE DATA LEVEL OF THE INDEX CONSISTS OF THOSE BUCKETS CONTAINING THE ACTUAL DATA RECORDS OF THE FILE. HOWEVER, WHEN DESCRIBING AN ALTERNATE KEY, THE DATA LEVEL OF THE INDEX CONSISTS OF BUCKETS IN WHICH RMS-11 MAINTAINS POINTER ARRAYS DESCRIBING THE DATA RECORDS.

Help for "HOW MANY BYTES DO YOU WANT FILLED IN THE DATA BUCKET (0)?"

AT CREATE TIME, YOU SPECIFY HOW MANY BYTES YOU WANT USED IN EACH DATA LEVEL BUCKET OF THE ASSOCIATED INDEX. BY SPECIFYING LESS THAN THE TOTAL BUCKET SIZE, YOU INDICATE THAT THE DATA BUCKETS ARE NOT TO BE COMPLETELY FILLED. YOU MAY WANT TO USE THIS FACILITY IF YOU EXPECT TO PERFORM NUMEROUS 'PUT' AND 'UPDATE' OPERATIONS ON THE FILE AFTER IT HAS BEEN INITIALLY POPULATED. ONE BLOCK IS EQUAL TO 512. BYTES

Help for "AREA NUMBER FOR INDEX BUCKETS FOR THIS KEY (0):"

YOU MAY ASSIGN THE BUCKETS OF THE INDEX LEVEL OF THE INDEX DEFINED BY THIS KEY TO ONE OF THE AREAS YOU JUST DEFINED. WHEN DESCRIBING THE PRIMARY KEY, THE INDEX LEVEL OF THE INDEX CONSISTS OF ALL LEVELS OF THE TREE-STRUCTURED PRIMARY INDEX DOWN TO AND INCLUDING THE LEVEL CONTAINING POINTERS TO THE USER DATA RECORDS THEMSELVES. HOWEVER, WHEN DESCRIBING AN ALTERNATE KEY, THE INDEX LEVEL CONSISTS OF ALL LEVELS OF THE TREE-STRUCTURED ALTERNATE INDEX DOWN TO, BUT NOT INCLUDING, THE LEVEL CONTAINING BUCKETS IN WHICH RMS-11 MAINTAINS POINTER ARRAYS DESCRIBING THE USER DATA RECORDS. YOU MAY SEPARATE THE LOWEST LEVEL OF THE INDEX FROM ALL HIGHER LEVELS, BUT THAT'S THE NEXT QUESTION.

Help for "HOW MANY BYTES DO YOU WANT FILLED IN THE INDEX BUCKET (0)?"

AT CREATE TIME, YOU SPECIFY HOW MANY BYTES YOU WANT USED IN EACH INDEX LEVEL BUCKET OF THE ASSOCIATED INDEX. BY SPECIFYING LESS THAN THE TOTAL BUCKET SIZE, YOU INDICATE THAT THE INDEX BUCKETS ARE NOT TO BE COMPLETELY FILLED BUT ARE TO CONTAIN SOME AMOUNT OF FREE SPACE. YOU MAY WANT TO USE THIS FACILITY IF YOU EXPECT TO PERFORM NUMEROUS 'PUT' AND 'UPDATE' OPERATIONS ON FILE AFTER IT HAS BEEN INITIALLY POPULATED. ONE BLOCK IS EQUAL TO 512. BYTES

Help for "AREA NUMBER FOR THE LOWEST INDEX LEVEL FOR THIS KEY (0):"

YOU MAY SEPARATE THE LOWEST LEVEL OF THE INDEX FROM ALL HIGHER LEVELS BY SPECIFYING A DIFFERENT AREA NUMBER THAN THAT SPECIFIED FOR THE INDEX LEVEL. HOWEVER, THE BUCKET SIZES OF THE TWO AREAS MUST BE THE SAME. THE DEFAULT VALUE OF ZERO MEANS THAT THE AREA NUMBER FOR THE LOWEST LEVEL OF THE INDEX WILL BE THE SAME AS THE AREA NUMBER FOR THE HIGHER LEVELS.

Help for "CLUSTERSIZE (0):"

ENTER A NUMBER BETWEEN 1 AND 255 REPRESENTING THE SIZE OF THE CLUSTERS FOR YOUR FILE. ONE CONDITION: THAT NUMBER MUST BE A POWER OF 2. THE DEFAULT VALUE OF ZERO YIELDS TO THE VOLUME DEFAULT. ENTERING A -1 IS EQUIVALENT TO ENTERING A CLUSTERSIZE OF 256.

Help for "PROTECTION (48):"

ENTER A NUMBER LESS THAN 64.
THIS NUMBER IS THE SUM OF VARIOUS CODES WHICH STAND FOR:

```
  1 = READ  PROTECT AGAINST OWNER
  2 = WRITE PROTECT AGAINST OWNER
  4 = READ  PROTECT AGAINST PROJECT
  8 = WRITE PROTECT AGAINST PROJECT
 16 = READ  PROTECT AGAINST ALL OTHERS
 32 = WRITE PROTECT AGAINST ALL OTHERS
```

Help for "DO YOU WANT TO CLOSE THE INDIRECT FILE(NO)?"

IF YOU SAY YES, THE COMMAND FILE YOU HAVE BEEN GENERATING WILL BE CLOSED AND YOU CAN BEGIN AGAIN. OTHERWISE, YOU CAN CONTINUE TO PUT COMMANDS INTO THE CURRENTLY OPEN COMMAND FILE.

Help for "DO YOU WANT TO SUPERCEDE THE FILE(NO)?"

IF YOU SAY YES, AN EXISTING FILE WITH THE SAME NAME WILL BE DELETED, AND IN ITS PLACE WILL BE YOUR NEW FILE.

Help for "DO YOU WISH TO REENTER THE ENTIRE FILE SPECIFICATION (YES)?"

THE FILE CANNOT BE CREATED DUE TO AN ERROR IN THE UIC OR DEVICE SPECIFICATIONS. YOU MAY CORRECT THE ERROR BY ANSWERING YES AND THEN REENTERING THE ENTIRE FILE SPECIFICATION. IF YOU ANSWER NO, THIS FILE WILL NOT BE CREATED, BUT YOU CAN START ANOTHER ONE.

B.4 DISPLAY UTILITY

```
                RMS-11 DISPLAY UTILITY
                ----------------------
FUNCTION: DISPLAY FILE ATTRIBUTES & THE CONTENTS OF CONTAINER FILE
FORMAT:   DSP [OUTFILE=]INFILE[/OPTIONS][,INFILE[/OPTIONS]..]

GLOBAL OPTIONS:
 ? OR HELP      TYPE HELP MESSAGE
 ID             IDENTIFY CURRENT VERSION NUMBER
 BP             LIST CONTENTS OF BACKUP CONTAINER FILE

OUTFILE :
                IS THE FILE SPECIFICATION OF A FILE TO BE
                CREATED BY THE DISPLAY UTILITY AND CONTAIN THE
                ATTRIBUTES OF THE SPECIFIED INPUT FILES OR A LIST
                OF THE CONTENTS OF THE SPECIFIED CONTAINER FILES

INFILE:
                IS THE FILE SPECIFICATION OF EITHER :
                1. AN RMS-11 FILE
                2. A BACKUP CONTAINER FILE ON MAGNETIC TAPE
```

B.5 RESTORE UTILITY

```
                RMS-11 RESTORE UTILITY
                ----------------------
FUNCTION: RESTORE BACKUP FILES
FORMAT:   RST OUTFILE[/OPTIONS]=INFILE[/OPTIONS][,INFILE[/OPTIONS]..

GLOBAL OPTIONS:
 ? OR HELP      TYPE HELP MESSAGES
 ID             IDENTIFY CURRENT VERSION NUMBER
 QU             ENABLE QUERY MODE
 SL:FILESPEC    PROVIDE SUMMARY LISTING FILE

OUTFILE OPTIONS:
 SU             SUPERSEDE EXISTING FILE

INFILE OPTIONS:
 BD:DD-MMM-YY   RESTORE FILES BASED ON DATE OF BACKUP
 OA:[N,M]       SELECT FILES BASED ON ORIGINAL ACCOUNT
 SE:FILESPEC    SELECT SPECIFIED FILE(S) FOR RESTORATION
 SE:(FILESPEC,..) FROM BACKUP CONTAINER FILE
```

INDEX

RSTS/E RMS-11
Utilities User's Guide
AA-0003B-TC

READER'S COMMENTS

NOTE: This form is for document comments only. DIGITAL will use comments submitted on this form at the company's discretion. Problems with software should be reported on a Software Performance Report (SPR) form. If you require a written reply and are eligible to receive one under SPR service, submit your comments on an SPR form.

Did you find errors in this manual? If so, specify by page.

_____

_____

_____

_____

_____

_____

Did you find this manual understandable, usable, and well-organized? Please make suggestions for improvement.

_____

_____

_____

_____

_____

_____

Is there sufficient documentation on associated system programs required for use of the software described in this manual? If not, what material is missing and where should it be placed?

_____

_____

_____

_____

_____

_____

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
- ☐ Non-programmer interested in computer concepts and capabilities

Name _____ Date _____

Organization _____

Street _____

City _____ State _____ Zip Code or Country _____

Please cut along this line.

---------- Fold Here ----------

---------- Do Not Tear - Fold Here and Staple ----------

FIRST CLASS
PERMIT NO. 33
MAYNARD, MASS.

BUSINESS REPLY MAIL
NO POSTAGE STAMP NECESSARY IF MAILED IN THE UNITED STATES

Postage will be paid by:

digital

Software Documentation
146 Main Street ML5-5/E39
Maynard, Massachusetts 01754