DATAPOINT

RMS

Quick Reference Guide

1151J
ENV :BB;I
NET
NODE
RES <NEWACK>
CAT
PASS

Document No. 61017. Printed in U.S.A.

RESOURCE MANAGEMENT SYSTEM

Datapoint's Resource Management System™ (RMS) operating system provides increased efficiency, flexibility;, and security for both stand-alone and Attached Resource Computer ™ (ARC™) systems. Supporting multiple users, multiple languages, and multiple functions, RMS operates in both shared logic and multiprocessing environments.

Through dynamic resource sharing, RMS makes optimum use of all system components including processors, memory, disk storage, peripherals, and communications facilities.

This dynamic sharing is provided through a unique approach to resource and file management. Every time a user signs on to RMS, a new operating environment is created--custom tailored to his needs.

This RMS Quick Reference Guide provides a quick overview of the signing on, environment establishing, and logging off procedures, along with a short explanation of available utilities.

Signing On to RMS

The RMS SIGNON program may be activated by entering a special key sequence from an inactive workstation as follows:

STEPS	WORKSTATIONS 1800/3800/5500/6000/6600/8600 KEYS	WORKSTATIONS 3600/8200 KEYS	WORKSTATIONS WITH IEOS KEYBOARD 8600/3800/8200/8220
1	DSP	NEW LINE	DSP
2	BACK SPACE	BACK SPACE	⇤ ←
3	DSP	NEW LINE	DSP

SIGNON

Establishing Environments

Three environments must always be present in the User Environment Table (UET):

:W — The Working Environment which is used to hold temporary work files created by user programs.

:U — The Utility Environment which is used to hold system utilities or other programs accessible to all users.

:L — The Local Environment used to access single file resources attached to the local node. This environment is automatically created by the SIGNON program.

ENVIRONMENTS

Following SIGNON, the user must define the WORKING (:W) and UTILITY (:U) environments. These prompts will not be displayed if a STARTUP file is used to automatically establish these two environments. Prompts appear as:

```
WORKING (:W) ENVIRONMENT
   NET NAME:
   NODE NAME:
   DISK NAME:
   CATALOG:
   PASSWORD:

UTILITY (:U) ENVIRONMENT
   NET NAME:
   NODE NAME:
   DISK NAME:
   CATALOG:
   PASSWORD:
```

ENVIRONMENTS

where

NET — is the name of the actual ARC system network to which the user wishes to gain access.

NODE — is the symbolic name for the processor on which the user wishes to run RMS.

DISK NAME — is the symbolic name of the multifile resource where the catalog for the environment being defined resides.

CATALOG — is the Hierarchical Structure Information (HSI) of the catalog file to be accessed.

PASSWORD — functions as the security key.

ENVIRONMENTS

LOGGING OFF

To exit from RMS, the workstation operator should log off by entering the following special key sequence from the appropriate ACTIVE workstation:

STEPS	WORKSTATIONS 1800/3800/5500/ 6000/6600/8600 KEYS	WORKSTATIONS 3600/8200 KEYS	WORKSTATIONS WITH IEOS KEYBOARD 8600/3800/ 8200/8220
1	DSP	□*	KBD
2	BACK SPACE	BACK SPACE	⇤ ←
3	DSP	□*	KBD

*Blank key above NEW LINE key

LOGOFF

RMS UTILITIES

RMS provides a comprehensive assortment of helpful utility programs. In the following pages, each RMS utility is briefly described. The first part of each section describes the purpose or function of the command. Next, the section shows a prototype command line, illustrating the general format of the command along with a brief explanation of the fields on the command line.

This guide is designed for quick reference; for more detailed information on any of these commands, refer to the RMS User's Guide, Document #50445. The CHAIN utility is not listed in this guide; a separate RMS CHAIN Quick Reference Guide, Document #61143, is available.

This quick reference guide uses notations similar to those used in the Concepts of the Resource Management System, Document #50592. These standard notations are explained as follows:

UTILITIES

STANDARD NOTATIONS AND GENERIC SYMBOLS

<field> A field in corner brackets indicates a replacement field. When the command is actually entered, some string will replace the name within the brackets. For example, <filename> might be replaced by INFILE.

[option] Square brackets enclose optional fields. Keywords, such as IN=, are not required if files are listed positionally.

ALL CAPS Words or letters appearing as upper case are to be entered exactly as they appear.

lower case Words or letters in lower case indicate that a value should be entered in their place.

+ The plus sign indicates that any combination of characters will be accepted from the + to the end of the name in a filename, extension, or environment.

? The question mark indicates that any one character, except a blank, will be accepted in that position in a filename or extension.

— The negation character indicates that any one character, except a blank or the character following it, will be accepted in that position in a filename or extension.

The following RMS Utilities provide the capability to manipulate user environments, display files contained in a catalog, and obtain information on resources, user tasks, and system status:

- BACKUP
- BACKUP9301
- BJF
- BUILD
- CAT
- CHANGE
- CHAREDIT
- CONFIG
- COPY
- CREATE
- CTFORMAT
- DCTAPE
- DELETE
- DUMP
- EDIT
- ENCODE
- ENV
- ERRLOG
- FMS
- GETARC
- GETDOS
- INDEX
- INFO
- LIB
- LINK
- LIST
- LOG
- MINIDOS
- NQDQ
- PACKGEN
- PCR
- PUTARC
- PUTDOS
- REFORMAT
- RENAME
- REPAIR
- SECURE
- SORT
- STARTUP
- TAPEUTIL

All RMS utilities have a HELP option which causes the display of command line syntax and options. The format of this HELP option is illustrated in the following example:

```
<utility name>;HELP
```

BACKUP-CATALOG MODE

The BACKUP utility allows the user to back up part or all of a disk's catalogs, including subordinate catalogs and files, onto one or more disks. BACKUP operates in three modes: catalog mode, serial mode, and restore mode, each of which uses a slightly different command line.

BACKUP in catalog mode backs up user-specified catalogs.

```
BACKUP   :<env 1>,:<env 2>[,<filespec>] [;<options>]
```

where

<env 1> is the input environment specification which identifies the disk resource and the catalog level to be backed up.

<env 2> is the output environment specification that identifies the disk resource and the catalog level to receive the backup.

<filespec> is the workfile specification used by BACKUP.

Refer to the BACKUP options (page 14) for a description of those that may be used in this mode.

BACKUP in serial mode copies catalogs too large to fit on the backup disk into one or more specially formatted serial output files so they can be backed up on multiple disks.

```
BACKUP   :<env>,<filespec 1>[,<filespec 2>];SERIAL[,<options>]
```

where

<env> is the input environment specification.

<filespec 1> is the specification for the first serial output file to receive the input files.

<filespec 2> is the work file specification.

Refer to the BACKUP options for a description of those that may be used in this mode.

BACKUP-SERIAL MODE

BACKUP-RESTORE MODE

BACKUP operates in RESTORE mode to restore files backed up in serial mode to their original format.

```
BACKUP <filespec 1>,:<env>[,<filespec 2>];RESTORE[,<options>]
```

where

<filespec 1> is the input file specification of the first (and possibly only) file generated by serial mode.

<env> is the output environment specification.

<filespec 2> is the work file specification.

Refer to the BACKUP options for a description of those that may be used in this mode.

The BACKUP utility allows the following options:

SERIAL - backs up disk data to one or more serial output files; this output must then be RESTOREd.

RESTORE - restores data backed up in serial mode to its original format.

INCLUDE - specifies generically or specifically the files to be RESTOREd.

EXCLUDE - specifies generically or specifically files that are not to be RESTOREd.

OVERWRIT - in any mode, overwrites existing output files.

BYPASS - in any mode, bypasses backing up a file that cannot be accessed.

SKIPNEXT - in any mode, bypasses backing up or restoring a file that has an error.

RESTART - restarts the current mode operation at the point of last failure.

BACKUP-OPTIONS

BACKUP9301 UTILITY

BACKUP9301 is a utility program designed for Datapoint's 8630 system (an 8600 processor with the 9301 disk/tape subsystem) to provide for mirror image backup copy from disk to tape or tape to disk. BACKUP9301 is completely interactive, prompting you for the specifications for any operation.

```
BACKUP9301;[<option>]
```

The only option available is the HELP option.

Build is used to combine records from various user-specified text files to create a new text file. BUILD allows you to create text files from any combination of three inputs:

1) New records entered from the keyboard
2) Records copied from existing files
3) Complete files that are appended to the output

```
BUILD  [OUT=]<filespec 1>[,WSIN=]<filespec 2>]
       [,WSOUT=]<filespec 3>] [;<options>]
```

where

[OUT=]<filespec 1> is the specifications for the output file being created.

[WSIN=]<filespec 2> is the specification for a user-created command file containing the directives BUILD is to execute.

[WSOUT=]<filespec 3> is the specification for a workstation output file for BUILD messages; otherwise, messages default to the screen.

BUILD UTILITY

BUILD-OPTIONS

BUILD provides the following options for environment control:

APP - directs BUILD to append the new data to the output file.

MAXREC = <n> - must be used when the largest record size in the file exceeds 256 bytes; specify 1, 2, 3, or 4 representing multiples of 1024 bytes.

The BUILD command uses the following directives:

ENTER - build new text file from records entered through the keyboard.

OPEN - used before COPY, DISPLAY, or SKIP to open an input file.

COPY
DISPLAY
SKIP - used to copy, display, or skip records from a specified input file.

TO, UNTIL,
FIRST, NEXT,
LAST - used to specify records for COPY, DISPLAY, SKIP.

END | * - terminates BUILD.

The Batched Job Facility (BJF) is a multi-utility package that provides the capability to schedule, execute, and print jobs in unattended mode. BJF can be combined with networking to provide a remote job processing facility.

Those utilities comprising BJF are:

BJF - initializes or restarts the Central Controller.

EXECUTE - creates an executor that processes CHAIN requests.

UNSPOOL - controls up to four printers and handles all print requests.

SUBMIT - submits requests to execute CHAIN files.

SPOOL - submits requests to unspool print files.

CONNECT - allows a CHAIN being run at an executor to communicate with the Central Controller.

CONSOLE - allows the user to interactively monitor and control BJF activity; requires the use of a workstation.

BJF UTILITIES

BJF-BJF

BJF [<filespec>] [,<:env1>] [,:<env2>] [;<options>]

where

<filespec> names the configuration file to be used by BJF.

<:env 1> specifies the environment for the Central Control File.

<:env 2> specifies the environment for the Central log file.

BJF has the following options:

CREATE - indicates that all BJF system disk files are to be created and intialized.

INDEP - causes the Central Controller to run as an independent task.

LOG = <nnn> - specifies the log file suffix; defaults to 001.

EXECUTE [<config file>] [;<options>] where

<config file> names the configuration file to be used by EXECUTE.

EXECUTE options:

NOCONFIG - causes EXECUTE to prompt for initialization data otherwise provided by the configuration files.

DUP - deletes any existing executor with the same node name and taskname.

The UNSPOOOL command line is:

UNSPOOL [<config file>]

where

<config file> specifies the configuration file that contains, for each of the four possible printer tasks, information such as printer name, class, type, form name, state description, buffers, password, and burst information.

BJF · EXECUTE/UNSPOOL

BJF - SUBMIT/SPOOL

SUBMIT [[IN =]<filespec>] [;<options>] [<profile keyword overrides>]

SPOOL [[IN =]<filespec] [;<options>] [<profile keyword overrides>]

where

<filespec> names the file containing the request profile; the request profile is comprised of keyword-parameter value pairs.

Options:

DUP - assigns a unique request name if the one specified is already in use; this assignment is done by appending a number to the original request name.

<profile keyword overrides> - are keyword-value pairs to be used instead of those specified in the profile. The following keywords are valid:

REQNAME = <name>
CLASS = <class>
REQFILE = <filespec>
PRIORITY = <name>
COPIES = <nnn>

FORM = <name>
REQDISP = (RELEASE | KEEP)
PRTDISP = (KEEP | DELETE)
AUTO = (YES | NO)
TYPE = <code>

REQPASS = <password>
CHAINOPT = <string>
EVENT = <name>
EVPASS = <password>
EVSTATE = <characters>
EVCODE = <code>

CONNECT [;<options>]

Options:

LOGMESSG - specifies that the next line in the CHAIN file contains a message.
RESTART = <option value> - specifies a value that is placed in the CHAINRST field in the request record.
SETEVENT = <event> - indicates the event to modify.
SETPASSW = <password> - is the password for SETEVENT.
SETSTATE = <state> - sets the event to the specified state.
CONTINUE - allows a remote abort.
CHKEVENT = <event> - specifies the event name to check.
CHKPASSW = <password> - is the password for CHKEVENT.
CHKSTATE = <state> - names the state that will not cause CHAIN to abort.
WATEVENT = <event> - names the event to wait on.
WATPASSW = <password> - is the password for WATEVENT.
WATSTATE = <state> - names the state to wait for.

BJF - CONNECT

BJF - CONSOLE

```
CONSOLE  [;<options>]
Options:
        NOTIMOUT - causes periodic contact with the Central Controller to avoid being timed out if
                   there are 10 minutes of inactivity.
The following subcommands are accepted by CONSOLE:

        ADD             - EVENT<event name>
        CHANGE          - EVENT<event name>
        CHANGE          - NODE<node name> ID {<task name> | <resource name>}
        CHANGE          - REQUEST<request name> CLASS<class name>
        DELETE          - EVENT<event name>
        DELETE          - NODE<node name> ID {<task name> | <resource name>}
        DELETE          - REQUEST<request name> CLASS<class name>
        LOG OFF         - NODE<node name> ID {<task name> | <resource name>}
        RESET           - LOG
        RESTART         - REQUEST<request name> CLASS<class name>
        RESUBMIT        - REQUEST<request name> CLASS<class name>
        SCAN            - EVENT<event name>
     DATE                         EXECUTOR | NODE | REQUEST | UNSPOOLER
     {END | *}                      MENU
     {HELP | ?}                     PASSWORD
        SHUTDOWN                    TIME
        START           - FAST | WAIT
        STOP            - NODE<node name> ID {<task name> | <resource name>}
                        - NODE<node name> ID {<task name> | <resource name>}
```

The CAT utility is used to display a list of the disk files currently accessible. It can display all the files available, all files that match a generic input mask, or a specific file.

In addition, through the use of the options, status information can be displayed for each file, and the format of the display can be altered.

```
CAT   [[IN=]<filespec 1>] [,[OUT=]<filespec 2> [;<options>]
```

where

[IN=]<filespec 1> represents either a single or generic file specifications.

[OUT=]<filespec 2> is the output file specification if the PRINT or FILESPEC options are used.

CAT UTILITY

CAT-OPTIONS

A single file specification names one file in the following format:

<filename>/>ext>:<env>

A generic file specification names more than one file and is created by using the symbolic replacement characters (+, ?, and -) in the name, extension, or environment fields. CAT +/+:+ displays all acccessible file specifications, for example.

CAT provides the following options:

INFO - displays status information on each file as well as file name, extension, and environment.

SORT - sorts the output according to file name, extension, or environment.

PRINT - directs output in print file format to disk or to printer.

CATALOG - selects and displays catalog names instead of files.

FILESPEC - causes the name of each file or catalog selected to be written to the output file specified by <filespec 2>; do not use this option and the PRINT option at the same time.

CHANGE provides the ability to display and modify the security access information for individual disk files or catalogs.

CHANGE [FILE=]<filespec>{;<options>}

where

[FILE=]<filespec> is the specification of the file to be changed.

CHANGE provides the following option:

CATALOG - used for specifying that security to a catalog file is to be changed. When this option is used, the <filespec> must name the environment.

CHANGE UTILITY

CHAREDIT UTILITY

CHAREDIT, the character set utility;, allows you to generate or modify a character set as an alternative to the English character set or to access other language character sets.

CHAREDIT [<filename>] where

<filename> is the file specification for the library to be modified (default is CHARFNT/SYS).

When the following CHAREDIT command is entered, you may select a directive from the following:

1 INPUT CHARACTER FONT TABLE
2 INPUT KEY TRANSLATE MEMBER
3 LIST DATABASE NAME
4 OVERWRITE CHARACTER FONT TABLE
5 OVERWRITE KEY TRANSLATE TABLE
6 EDIT CHARACTER FONT
7 EDIT KEYIN TRANSLATE TABLE
8 EDIT 82XX KEYBOARD TRANSLATE TABLE
9 CREATE CHARLOAD LIBRARY FILE
10 LOAD KEY TRANSLATE TABLES
11 DELETE MEMBER FROM DATABASE
12 ADD CHAR FONT TO DATABASE
13 ADD TRANSLATE TABLE TO DATABASE
14 SELECT CHAR FONT FROM DATABASE
15 SELECT TRANSLATE TBL FROM DATABASE
16 RETURN TO RMS

The CHARLOAD command interfaces with the CHAREDIT utility and allows the user to load character fonts and keyboard translation tables.

CHARLOAD [[IN]=<filespec>] [;<options>]

where

<filespec> is the file specification for the CHARLOAD library file containing the character set and key translation table to be loaded.

CHARLOAD COMMAND

CONFIG-SYSTEM MODE

CONFIG operates in two modes--System Mode and Utility Mode. In System Mode, it creates a file containing software configuration information that is read when the system is initialized.

```
CONFIG   [[OUT = ] [<filespec 1>]] [,[IN = ] [<filespec 2>]]
         [,[NUC = ] [filespec 3>]] [,[WSIN = ] [<filespec 4>]]
         [,[WSOUT = ] [<filespec 5>]] [;<options>] where
```

[OUT =]<filespec 1> is the System Configuration File (SCF) to be updated.

[IN =]<filespec 2> is the provided Subrecord Definitions File (SDF) containing the input to CONFIG.

[NUC =]<filespec 3> is a user-created command file which contains the directives CONFIG is to execute.

[WSIN =]<filespec 4> is a user-created command file which contains the directives CONFIG is to execute.

[WSOUT =]<filespec 5> is a workstation output file where the messages from CONFIG are written.

CONFIG provides the following options:

REPAIR - directs CONFIG to recover data from a badly formatted SCF.

VERSION - specifies nucleus and command libraries to be configured.

SYSTEM - directs CONFIG to run in System Mode.

CONFIG uses the following directives to complete its tasks:

ALL - causes all nucleus library members to be loaded for possible inclusion in the configuration specification.

DISK - causes all disk supporting nucleus library members to be loaded for possible inclusion in configuration.

INCLUDE - allows the user to select a specific nucleus library member for inclusion in configuration.

END - terminates configuration and causes the new SCF to be created.

CONFIG-SYSTEM MODE

CONFIG-UTILITY MODE

CONFIG in Utility Mode creates or modifies a file containing configuration information read when the member corresponding to the configurable utility is initialized.

```
CONFIG   [[OUT = ] [<filespec 1>]] [,[IN = ] [<filespec 2>]] [,[WSIN = ]
         [<filespec 3>]] [,[WSOUT = ] [<filespec 4>]] [;<option>]
```

where

[OUT =]<filespec 1>   is the System Configuration File (SCF) to be updated.

[IN =]<filespec 2>   is the provided Subrecord Definitions File (SDF) containing the input to CONFIG.

[WSIN =]<filespec 3>   is a user-created command file which contains the directives CONFIG is to execute.

(WSOUT =)<filespec 4>   is a workstation output file where the messages from CONFIG are written.

CONFIG in Utility Mode provides the following options:

REDEFINE - directs CONFIG to ignore any existing data for the specified member.

EXIST - directs CONFIG to terminate if the specified member does not exist.

NOEXIST - directs CONFIG to terminate if the specified member exists.

UTILITY - directs CONFIG to run in Utility Mode.

CONFIG-UTILITY MODE

COPY UTILITY

COPY allows the user to produce a duplicate COPY of a single file or group of files for backup or distribution. It also provides the capability of deleting source files after the copies are performed.

```
COPY   [IN=]<filespec 1>,[OUT=]>filespec 2>[;<options>]
```

where

[IN=]<filespec 1> is either a single file specification or a generic file specification for the file being copied. Refer to page 9 for generic replacement symbols. Using +/CMD:+ specifies all files in all environments whose extensions are CMD, for example.

[OUT=]<filespec 2> is the specification for the output file.

COPY provides the following options:

ASIS - retains current disk structure of output file.

BACKUP - transfers security information of the input file to the output file.

BYPASS - continues processing the next file specification if an output file already exists; used with generic copy.

CANCEL - terminates processing of COPY if the output file exists.

DELETE - deletes specified input file after COPY completes; file specification may be generic.

MAYBE - asks what action to take if output file already exists.

OVERWRIT - deletes existing output file and creates new output file with input file parameters.

PROMPT - prompts user to verify each COPY before it is performed; for generic copy only.

SORT - sort output according to file name, extension, and environment; for generic copy only.

COPY-OPTIONS

CREATE UTILITY

CREATE allows you to create a file specifying space allocation that reserves contiguous disk space.

```
CREATE   [[OUT=]<filespec 1>],[[LIKE=]<filespec 2>] [;<options>]
```

where

[OUT=]<filespec 1>	is the file specification for the file to be created.
[LIKE=]<filespec 2>	is the specification for a file which defines the following parameters: FMT, LEN, INIT, INC.

CREATE provides the following options:

FMT=< x >	sets the file format.
LEN=<n>	allocates <n> sectors to a file.
INIT=<n>	allocates <n> contiguous sectors in the first file segment.
INC=<n>	allocates <n> sectors per segment as required.
ABORT	terminates CREATE and deletes the output file if the LEN, INIT, or INC cannot be completed.

CTFORMAT allows you to prepare a cartridge tape for use with the 8800 and to add an IPL file to the cartridge tape. If no option is specified, CTFORMAT interactively prompts for any desired information.

```
CTFORMAT   [<filespec>] [;<options>]
```

where

<filespec>	is the specification for the file containing the IPL block; the file must be a command library type.

CTFORMAT has the following two options:

FORMAT	- formats a cartridge tape into two tracks of 256-byte data blocks.
IPL	- writes the user-specified IPL file on the tape.

CTFORMAT UTILITY

DCTAPE allows the transfer of text and binary data (only) between industry standard magnetic tapes and RMS disk. DCTAPE also intializes unlabeled, ANSI labeled, and IBM labeled tapes for use by COBOL programs. DCTAPE must be executed using one of four subfunction options.

The tape intialization subfunction creates labels on magnetic tape.

```
DCTAPE  [[RES=]<filespec 1>];TINIT,LABEL={A|I|U} [,<options>]
```

The tape input subfunction transfers files from magnetic tape to RMS disk.

```
DCTAPE  [RES=]<filespec 1>[,[[LOADFILE=]<filespec 2>];TIN,
        LABEL={A|I|U} [,<options>]
```

The tape output subfunction transfers files from RMS disk to magnetic tape.

```
DCTAPE  [RES=]<filespec 1>[,[LOADFILE=]<filespec 2>];TOUT,
        LABLE={A|I|U} [,<options>]
```

The tape examination subfunction analyzes the label of any tape to determine the tape's structure.

```
DCTAPE  [RES=]<filespec 1>[,[LOADFILE=]<filespec 2>]
        [,[PRT=]<filespec 3>];TEXAM [,<options>]
```

where

<filespec 1>	specifies the magnetic tape device.
<filespec 2>	specifies a library containing any user-assembled translate tables and/or user-label processing routines.
<filespec 3>	specifies the printer resource or the disk print file.

DCTAPE OPTIONS

DCTAPE provides the following options; each subfunction (TINIT, TIN, TOUT, TEXAM) has its own HELP option which defines compatible options.

LABEL = {A|I|U} - label type A is (ANSI), I (IBM), or U (unlabeled).

LOAD - leaves tapes ready on completion.

TRL = {719} - specifies 7- or 9-track tape.

LTRANS = {ASCII / EBCDIC / BCD},<file member name>
- specifies the type of label translation.

MODE {CIT} - T converts 6-bit to 8-bit bytes; C, 8-bit to 6-bit.

DTRANS = {ASCII | EBCDIC | BCD | BIN},<file member name>
- determines the data translation table to be used. BCD may be used only with 7-track types.

ULAB = <filemember name> - names the user-label processing routine filename.

BPI = <556|800|1600} - verifies the recording density of the tape.

OVERWRIT - overwrites any specified and already existing file.

MLTIREEL = <nn> - specifies the number of tape reels to be used.

DELETE provides the ability to delete an unwanted file or group of files from any accessible environment. The deleted file is no longer listed in the directory and its disk space is deallocated.

DELETE [IN =]<filespec>[;<options>]

where

[IN =]<filespec> is either a single file specification or a generic file specification which limits the scope of the files to be deleted. Refer to page 9 for generic replacment symbols. Using ABC?/ + : + specifies all 4-character filenames whose first 3 characters are ABC, with any extension in any environment, for example.

DELETE provides the following options:

PROMPT - asks the user to verify each DELETE before it is performed.

NOPROMPT - deletes specified files without prompting for verification.

SORT - sorts according to file name, extension, or environment.

DELETE UTILITY

DUMP UTILITY

DUMP allows the user to interactively display, print, or modify data stored on a specified disk resource.

```
DUMP   [IN=]<filespec>[,[PRT=]<printer>];[<options>]
```

where

[IN=]<filespec> is the RMS file specification of the file being DUMPed.

[PRT=]<printer> is the device specification of the printer to which hardcopy ouput is directed.

DUMP provides the following options:

RES - states that the <filespec> is the RMS resource name of the disk.

WRITE - allows the user to modify the file being displayed.

EDIT allows the user to create and modify text files.

```
EDIT   [IN=]<filespec 1>[,[OUT=]<filespec 2>]
       [,[WORK=]<filespec 3>] [;<options>]
```

where

[IN=]<filespec 1> is the specification for the input file to be created or modified.

[OUT=]<filespec 2> is the specification for the ouput file where the updated file is stored when EDIT is terminated.

[WORK=]<filespec 3> is the specification for the scratch file.

EDIT provides the following options:

CREATE - creates the specified input file.
NOCOMP - causes uncompressed records to be written.
READ - allows the input file to be opened in read only mode.

The EDIT command allows modification of text through directives (special characters) that are entered from the keyboard.

EDIT UTILITY

ENCODE UTILITY

The ENCODE utility converts disk files containing any type of data into text files suitable for transmission by a data communications program. The ENCODE utility also has a DEC (decode) capability to restore the encoded data to its original form.

```
ENCODE   [IN=]<filespec 1>[,OUT=]<filespec 2>] [;<options>]
```

where

[IN=]<filespec 1> is the file being encoded or decoded. The filename must be provided.

[OUT=]<filespec 2> is the file resulting from the encoding or decoding operation.

ENCODE provides the following options:

DEC - determines that decoding will be performed.

ENC - determines that encoding will be performed.

CONV - converts binary data to 6-bit ASCII. With DEC, it restores data in an encoded 6-bit file to original form.

KEY - encodes or decodes file.

ENV is used to create, delete, and modify additional environments to control a user's scope of access to the system. ENV also allows mapping the hierarchical structure of the disk and system backup.

```
ENV  [[ENV=]:<envname 1>] [,[AFTER=]:<envname 2> [,[PRT=]<filespec 1>]
     [,[WORD=]<filespec 2>] [,[BKENV=]:<envname 3>[ [,[BKCHN=]<filespec 3>]
     [;<options>]
```

where

[ENV=]:<envname 1> names the environment to be created, modified, deleted, or displayed.

[AFTER=]:<envname 2> is used to position <envname 1>.

[PRT=]<filespec 1> specifies the print file to be used for MAP option.

[WORK=]<filespec 2> specifies the work file for the MAP option.

[BKENV=]<envname 3> specifies the backup environment for the BACKUP option.

[BKCHN=]<filespec 3> is the output CHAIN file created by the BACKUP option.

ENV UTILITY

ENV-OPTIONS

ENV provides the following options for environment control:

CREATE	- adds entry to UET for <envname> and creates catalog file.
DELETE	- deletes environment and catalog file.
INSERT	- adds environment to UET for existing catalog file.
REMOVE	- removes environment from UET.
MODIFY	- changes the passwords for an environment.
PASS	- displays environments and passwords.
MAP	- maps hierarchical structure of disk.
BACKUP	- initiates a backup of all files within and subordinate to a catalog.
FORCE	- inserts an entry in the UET even if the resource to be accessed does not exist in the system.
ID	- changes the description of a catalog.

ERRLOG UTILITY

The Error Recording and Reporting (ERRLOG) utility allows you to log hardware failures of disk and communications I/O devices. A report summarizing the logged information may be generated through the ERREPORT utility.

```
ERRLOG  [[LOG = ]<filespec 1> [,[WSIN = ]<filespec 2>]
        [,[WSOUT = ]<filespec 3] [;<options>]
```

where

[LOG =]<filespec 1>	is the specification for the Error Logging File (ELF) which is used to record hardware failure information.
[WSIN =]<filespec 2>	is a file the user may create to contain ERRLOG directives.
[WSOUT =]<filespec 3>	specifies a file to contain all messages returned during ERRLOG.

ERRLOG UTILITY

ERRLOG provides the following options which can be specified on the command line or chosen from a main menu:

RES[=ALL|<resource name> - specifies the resources which will be logged for errors.

CONT[=ALL|<controller ID> - specifies the controllers which will be logged for errors.

8800 [=ALL|<8800 node num>] - specifies the 8800 nodes that will be logged for memory errors.

START - starts the error logging task.

STATUS - displays the error logging status.

STOP - terminates the error logging task.

The RMS Error Report utility (ERREPORT) generates a report of the RMS error log, displaying the following information:

1) System start time (IPL)
2) Resource name or controller ID
3) Net and node name
4) Communicating drive ID and error threshold
5) Logical port number (controllers only)

```
ERREPORT  [[IN=]<filespec 1>] [,[OUT=]<filespec 2>] [;<options>]
```

where

[IN=]<filespec 1> specifies the error log input file.

[OUT=]<filespec 2> specifies the error report output file.

ERREPORT offers the following options:

NOSORT - assumes input file is pre-sorted.

SELECT - after sorting, allows selected data to be printed.

AVG - accumulates total activity and computes the hourly activity average.

ERREPORT

FMS UTILITIES

The File Management System (FMS) utilities enhance system throughput by providing managed file I/O access as well as standard file access capabilities. FMS is made up of the following components:

FAR — File Access Routines (FAR) make up a library of routines for I/O processing. These routines are either executed by FMS or called by user programs.

MANAGE — creates or updates for each file to be managed a unique Managed File Description (MFD) containing information about the file.

CONFGFMT — creates or updates a unique FMS configuration file for each File Management Task (FMT) to be executed.

FMTCONT — initializes and terminates an FMT.

The MANAGE utility interactively creates or edits the unique Managed File Descriptor (MFD) that must exist for each file to be managed by FMS. Four out of five possible screens, depending on whether the file to be managed is DIRECT or ISAM, prompt the user for file description, FMT directory information, and FMT description.

MANAGE <filespec>[;<options>]

where

<filename> — is the file specification of the MFD to be created or modified.

The following option is available for MANAGE:

CREATE — - directs MANAGE to create the specified file as a new MFD.

MANAGE UTILITY

CONFGFMT UTILITY

The CONFGFMT utility creates or modifies a configuration file for a given FMT. A configuration file must be created for each FMT to be executed. parameters are defined according to the responses supplied to prompts displayed through a series of interactive screens.

```
CONFGFMT   [<filespec>] [;<options>]
```

where

<filespec> is the output file created or modified by this utility containing the configuration information for an FMT.

CONFGFMT has one option:

CREATE - creates a new FMT configuration file.

The FMTCONT utility initializes or terminates an FMT. While initialized, the FMT performs the actual user file access and buffer management for a given file.

```
FMTCONT   [<filespec>] [;<options>]
```

where

<filespec> specifies a configuration file created by the CONFGFMT utility.

FMTCONT offers the following options:

DIE - terminates the FMT after all users have logged off.

ABORT - directs the FMT to disregard all logged-on users and to terminate immediately.

DEBUG = <wsname> specifies an available resource name whose device type must be a workstation; the standard RMS debug display appears on the screen of this station.

FMTCONT UTILITY

FMTCONT UTILITY

GETARC UTILITY

The GETARC utility allows you to transfer files from a DOS ARC file processor to an RMS disk, providing an interface between the two operating systems. GETARC cannot transfer files from a local disk. (See GETDOS for local disk capabilities.)

GETARC [IN=]<filespec 1>[,[OUT=]<filespec 2>] [;<options>]

where

[IN=]<filespec 1> is the name and extension of the DOS ARC file to be copied; this file must be specified.

[OUT=]<filespec 2> is the file specification of the RMS output file.

The following options are available using GETARC:

BIN - specifies a binary type transfer.

VOL= - specifies the DOS ARC value identification.

NAME= - specifies the DOS ARC user name.

CODEWORD= - specifies the DOS ARC user codeword

MEMBER= - is the DOS ARC library member name; the DOS library name must be specified as <filespec 1> with this option.

NET= - is the RMS network name to which the output file is directed.

DOSVER=<n> - is the DOS version being used.

ARCVER=<n> - is the ARC version being used.

DOSREL=<n> - is the DOS release being used.

ARCREL=<n> - is the ARC release being used.

GETDOS UTILITY

GETDOS reads files from a DOS formatted disk and writes them to an RMS formatted disk.

```
GETDOS [IN=]<filespec 1>[,[OUT=]<filespec 2>] [;<options>]
```

where

[IN=]<filespec 1>	is the name and extension of the DOS file to be copied. The name must be specified.
[OUT=]<filespec 2>	is the file specification of the RMS output file.

GETDOS provides the following options:

BIN	- specifies a binary transfer type.
VOL=	- is the DOS volume name.
SUR=	- names the DOS subdirectory of the input file.
LEN=	- length of the output file.
INC=	- file increment of the output file.
NOCOMP	- specifies that the records of the output file will be written uncompressed.
MEMBER=	- specifies a DOS library member name (<filespec 1> must specify the DOS library name).
RECSIZE	- specifies a fixed length record size.

INDEX creates or reorganizes an index file to provide ISAM access to the records of a text file.

```
INDEX          [IN=]<filespec 1>
               [,[OUT=]<filespec 2>]
               [,[PRT=]<filespec 3>]
               [,[WK1=]<filespec 4>]
               [,[WK2=]<filespec 5>]
               [,[SEQ=]<filespec 6>]
               [,[TAG=]<filespec 7>]
               [,[SRC=]<filespec 8>]
               [;<option 1>] [,<option 2>]
```

where the exact order of the file specifications varies depending on the function selected on the command line.

INDEX UTILITY

INDEX UTILITY

INDEX provides options to select the function desired and to provide parameters for the selected function:

Select Function		Provide Parameters	
No option	- use data file	LF = <nn>	- loading factor
I	- use tag file	SPB = <n>	- sectors per block
X	- reorganize	KPB = <n>	- records per block
INFO	- display header	D	- duplicate keys allowed
ENVMOD	- modify env mask	S = "<sort options>"	- identifies fields in the data file to be used as a sort key.
KTAGDEL	- delete tag file when task over		
		M = <n>	- maximum record length.

The INFO utility provides the user with system status information on system resources. No file specifications are required. If no options are specified, resources available through the UET are displayed on a scrolling screen.

INFO [;[<option 1>] [,<option 2>] . . .]

Option 1:

RES	- displays status information for non-disk resources only.
DISK	- displays status information for disk resources only.
TASKS	- displays status information for executing tasks only.
CONNECTS	- displays status information for nodes only.
LINKS	- displays status information for Resource Connection Links.
UTIL	- displays resource utilization for the local node only.

Option 2:

NET = <a>	- includes information from all networks or a specified net.
NODE = <a>	- includes information on all nodes or a specified node.
ENV = <a>	- display items in specified environment.
ALL	- display information for all nets & nodes found in all environments accessible to local node.
SUMMARY	- provide a summary of the number of RES, DISKS, TASKS, CONNECTS, and LINKS displayed.

INFO UTILITY

LIB UTILITY

The LIB utility is used to create and maintain command libraries.

```
LIB  [OUT=]<filespec 1>[,[WRK=]<filespec 2>] [,[WSIN=]
     <filespec 3>] [,[WSOUT=]<filespec 4>] [,[options>]
```

where

[OUT=]<filespec 1>]	is the library that is created, modified, or accessed.
[WRK=]<filespec 2>]	is the temporary work file used by the CLEANUP directive.
[WSIN=]<filespec 3>]	is the specification for a user-created file containing LIB directives; otherwise, directives are entered at the workstation.
[WSOUT=]<filespec 4>	is the specification for a file to receive all messages returned during LIB; otherwise, messages are displayed at the workstation.

LIB has one option;

CREATE	- creates a new output library.

LIB directives provide the following capabilities:

ADD REPLACE EXTRACT DELETE RENAME MERGE	manipulates or renames library members.
ENTRY	designates the first member of a library.
OPEN	specifies a new output library.
GENERATE	creates a member that contains all external definitions from relocatable members in the current library.
CAT	lists all members in the library.
CLEANUP	recovers space lost by DELETEs and REPLACEs.
{END ¦ *}	terminates LIB execution.
CREATE	creates a new output library.
SIGNON	allows the sign-on message to be modified.

LIB-DIRECTIVES

LINK UTILITY

LINK manipulates and transforms relocatable members of libraries into either absolute or relocatable members of a LINK output library.

```
LINK  [[[CMD=]<filespec 1>] [,[OUT=]<filespec 2>] [,[PRT=]<prntfile>]
      [,[ENV=]<env>] [;<options>]
```

where

<filespec 1> is the optional input file of LINK directives.

<filespec 2> is the output library into which any absolute members generated by LINK are added.

<filespec 3> is the print file produced by the PRNT or Q options.

<env> is the environment for the LINK work file.

LINK provides the following options:

ABBREV - asks for abbreviated symbol table or cross-reference list.

DISP - generates and displays loader table listing on screen.

ERR - indicates that the object file is not to be opened and LINK is to be terminated if any errors are encountered.

PRNT - produces print file.

Q - instructs LINK to queue the LINK print file.

REL - generates relocatable code.

SYMB - alphabetically sorts, then prints external symbol table entries.

TABLE - directs symbol table printing in tabular form.

UREL - generates relocatable output that can be loaded and executed.

XREF - produces sorted cross-reference list of symbol table.

NEW - overwrites the old library with the new library.

FASTLIB - provides an alternative library search scheme.

LINK-OPTIONS

LIST UTILITY

The LIST utility provides the capability to transfer files from one resource to another to display files, and to format the files for printed output.

```
LIST   [[IN=]<filespec 1>] [,[OUT=]<filespec 2>]
       [,[VFUCMNDS=]<filespec 3>] [;<options>]
```

where

[IN=]<filespec 1> is the name of the file to be listed.

[OUT=]<filespec 2> is the specification for the output device or for the formatted output file.

[VFUCMNDS=]<filespec 3> specifies the text file containing VFU directives used with the VFU option only.

List offers the following options:

FORM	- formats input file.
LREC=<n>	- starts at record n.
COL	- displays output with column headings.
LPP=<n>	- changes page length to <n> lines.
QUEUE	- creates formatted file for output and append to existing file.
VFU	- changes forms control information before LISTing to a printer or a print file.
SYSPARM	- controls channel advancing with the VFU option.
WRAPLN=<n>	- specifies the maximum length of an output line when WRAP is used.
LSEC=<n>	- starts list with the second logical record in the specified sector.
MAXREC=<n>	- states maximum input record size in K bytes, where n is 1, 2, 3, or 4.
WRAP	- wraps to next line; the default line length is 110.
ISAM	- lists on ISAM file.
NONBR	- suppresses line numbers.
PRINT	- creates print file.

LIST-OPTIONS

LOG UTILITY

The LOG utility provides a job trace capability logging all informational and error messages displayed at a workstation screen for the duration of the log. This log file is helpful to keep record of job completions and times, and to help analyze errors.

LOG [[OUT =]<filespec>] [;<options>]

where

[OUT =]<filespec> is the file specified to receive the logged output.

The LOG utility offers the following options:

ON	- activates logging	ERROR	- logs error messages.
OFF	- terminates logging.	ALL	- logs all messages.
QUEUE	- appends output to a log file.	SUSPEND	- suspends logging.
PRINT	- sends log information to a local printer.	RESTART	- restarts logging.

The MINIDOS utilities, MINIDISK and MINIARC, allow an RMS user to simulate a DOS ARC file processor on an RMS node, enabling RMS users to benefit from some DOS software features not available under RMS.

MINIDISK [OUT =]<filespec>[;<options>]

where

[OUT =]<filespec> names the RMS file created (or extended) to contain the DOS pseudo-volume.

MINIDISK options are:

SEC = <n> - specifies the number of RMS sectors to create.

EXT = <n> - specifies the number of RMS sectors by which to extend the filesize.

MINIARC [[CNFG =]<filespec>] [;<options>]

where

[CNFG =]<filespec> is the configuration file containing information needed to establish the DOS volume; if this file is not available, MINIARC prompts for the needed data.

MINIDOS UTILITIES

NQDQ UTILITY

The Enqueue/Dequeue (NQDQ) utility allows the local operator to activate this access-handling system for use by user programs.

```
NQDQINIT   [<filespec 1>] [[LOG=]<filespec 2>] [;<options>]
```

where

<filespec 1> is the configuration file containing configuration data for the NQDQ system.

[LOG=]<filespec 2> specifies the log file if the LOG option is used.

NQDQ provides the following options:

CMD=S	- starts the NQDQ session.
CMD=T	- terminates the NQDQ session.
LOG	- generates a log file of NQDQ activity for debugging.
LOGSIZE	- specifies in sectors the size of the log file.

The PACKGEN utility allows the user to initialize disk for use under RMS. Initialization includes verifying sectors, generating system tables, and formatting soft-sectored disks.

```
PACKGEN   <resource name>[<filespec>];<option>
```

where

<resource> is the resource name of the pack being initialized.

<filespec> is the Initial Program Load (IPL) file specification, if a special IPL is needed, that corresponds to the type of disk being generated.

PACKGEN offers the following options:

IPL	- to write only the IPL blocks.
RES	- to write only the resource name.
GEN	- for a complete pack generation.

PACKGEN UTILITY

PCR UTILITY

The PCR utility allows you to access and modify selected entries in the Program Communications Region (PCR) in support of local requirements and conditions.

```
PCR   [;<options>]
```

No file specifications are required. All current PCR information is displayed if none of the following options are specified.

AVAIL - returns number of bytes of available PCR memory.

FORMLGT=<nnn> - allows you to assign a value for form length.

HEADING={ON|ALL|OFF} - determines display of heading line.

MAXMEM=<nnn> - resets maximum task memory limit in the PCR to <nnn>K bytes.

MINMEM=<nnn> - reserves <nnn>K bytes of minimum task memory.

PRIORITY=<n> - allows you to set priority of task.

RESERVE=<nnnn> - reserves specified number of bytes of PCR memory.

TIME={SET|ADJUST|BOTH> - sets or adjusts time and date (or both).

The PUTARC utility allows you to transfer files from an RMS disk to a DOS ARC file processor providing an interface between the two operating systems. PUTARC cannot transfer RMS files from a local disk (see PUTDOS for local disk capabilities.)

```
PUTARC   [IN=]<filespec 1>[,[OUT=]<filespec 2>] [;<options>]
```

where

<filespec 1> is the file specification of the RMS file to be copied.

<filespec 2> is the name and extension of the DOS output file.

The following options are available using PUTARC:

BIN - specifies a binary transfer type.
VOL= - specifies the DOS ARC volume identification.
NAME= - specifies the DOS ARC user name.
CODEWORD= - specifies the DOS ARC user codeword.
NET= - is the RMS network name to which the output file is directed.
DOSVER=<n> - is the DOS version being used.
ARCVER=<n> - is the ARC version being used.
DOSREL=<n> - is the DOS release being used.
ARCREL=<n> - is the ARC release being used.

PUTARC UTILITY

PUTDOS UTILITY

PUTDOS provides the capability to copy RMS formatted text files to a DOS formatted disk.

```
PUTDOS   [IN=]<filespec 1>[,[OUT]<filespec 2> [;<options>]
```

where

[IN=]<filespec 1> is the RMS text file specification.

[OUT=]<filespec 2> is the DOS text file specification.

PUTDOS offers the following options:

SUR=<DOS subdirectory> - used to select the subdirectory for the output file.

VOL=<DOS volid> - allows the user to specify the DOS volume to receive the data.

BIN - copies the RMS file to DOS sector by sector, overwriting the first three bytes of each sector with the DOS sector verification PFN and LRN.

The REFORMAT command allows the user to alter the format of RMS text files and to modify the structure of records within a text file.

```
REFORMAT  [[IN=]<filespec 1>],[OUT=]<filespec 2>
          [,[WSIN=]<filespec 3>] [,[WSOUT=]<filespec 4>] [;<options>]
```

where

[IN=]<filespec 1> is the text file to be reformatted.

[OUT=]<filespec 2> is the output file to be created by REFORMAT.

[WSIN=]<filespec 3> names a user-created source file containing REFORMAT directives. If no file is specified, REFORMAT defaults to the workstation.

[WSOUT=]<filespec 4> names the destination file for all REFORMAT messages; otherwise, messages are shown at the workstation.

REFORMAT UTILITY

REFORMAT-OPTIONS

REFORMAT provides the following options:

NOCOMP - specifies that the output file is not to be space compressed.

LEN = <n> - allows the user to specify the length of each logical record in number of bytes (n).

TRUN - is used with the LEN option to specify that any logical record longer than n bytes will be truncated.

SEG - is used with the LEN option to specify that any logical record longer than n bytes will be divided into two or more records.

FIX - is used to enter FIX mode; this option must be specified before the user can execute the REPLACE, INSERT, and DELETE interactive directives.

MAXREC = <n> - indicates that the largest record size in the file exceeds 256 bytes; <n> must be 1,2,3, or 4, representing 1 to 4 multiples of 1024 bytes (1K).

Along with the options listed above REFORMAT uses the following directives to complete its tasks. Interactive mode is entered only if no input file is specified on the command line.

Not in FIX Mode

```
OPEN<filespec>
COPY(<column>,<length>)
INCLUDE
EXCLUDE
IF<expression><logical operator>
    <expression>...THEN
XIF
GO
END | *
```

In FIX Mode

```
OPEN<filespec>
REPLACE(<column>,<length>)"<string>"
INSERT(<column>,<length>"<string>"
DELETE(<column> <length>)
IF<expression><logical operator>
    <expression>...THEN
XIF
GO
END | *
```

REFORMAT-DIRECTIVES

RENAME UTILITY

The RENAME utility allows you to assign a new name, extension, or environment to an existing file.

RENAME [IN=]<filespec 1>,[OUT=]<filespec 2>[;<options>]

where

[IN=]<filespec 1> is a single or generic file specification for the file being renamed. A generic file specification may contain the symbolic replacement characters.

[OUT=]<filespec 2> is the new file specification for the file.

RENAME provides the following options:

BYPASS
CANCEL
PROMPT
- file protect options which specify the action that the RENAME utility should take if the output file already exists.

SORT=<aaa> - alphabetizes the displayable output according to the filename (F), extension (X) or environment (E). Any combination of FXE can be specified or <aaa>.

The REPAIR utility is a highly interactive utility that can be used to check and correct errors on disk resources. As REPAIR is executed, it prompts for instructions.

REPAIR <resource>[;<options>]

where

<resource> names the disk resource to be REPAIRed.

REPAIR has one option:

OVERFLOW - handles REPAIR's additional memory requirements when REPAIR is running on a 128K single drive system or an 8600 processor only. Caution: This option can cause files on the disk resource to be destroyed.

REPAIR UTILITY

SECURE UTILITY

The SECURE utility allows the user to change the following security parameters of any RMS file:

- File access code
- File sector allocation size
- Password names
- Security level

SECURE [[IN =]<filespec>] [;<options>]

where

[IN =]<filespec> is the input file specification. For selecting a generic file specifications, use the format:

<generic filename>/<generic ext>:<specific env>

A specific environment must be provided on the command line or an error will result.

SECURE provides the following option:

PROMPT - prompts for verification that security processing is to occur.

The following directives may be entered via the SECURE Utility: REA WRI DEL REN SEC

```
SET <Password Reference>; <Acc Bits>
ADD <Password Reference>; <Acc Bits>
REM(ove) <Password Reference> [;<Acc Bits>]
CRE(ate) <Password Reference> [;<Acc Bits>]
DEL(ete) <Password Reference>
REN(ame) <Password Reference> {Password}
SEC(urity) <Secr Level>
IAC [<Acc Bits>]
FINC <Sector Ct>
{HELP | ?} - displays menu of directives.
{END | *} - begins processing any directive entered.
```

SECURE-DIRECTIVES

The SORT utility allows the user to sort a text file according to any specified collating sequence.

```
SORT   [IN=]<filespec 1>,[OUT=]<filespec 2>[,[WORK=]<filespec 3>]
       [,[SEQ=]<filespec 4>] [;<options>]
```

where

[IN=]<filespec 1>	is the input file specification.
[OUT=]<filespec 2>	is the output file specification.
[WORK=]<filespec 3>	is the work file used by SORT.
[SEQ=]<filespec 4>	is the file containing an alternate collating sequence.

SORT provides the following options:

A	- sorts in ascending order.
D	- sorts in descending order.
C	- sets compression on.

(Continuation of SORT options)

N	- sets compression off.
I	- blocks to 1 record/sector.
M<nnn>	- specifies maximum input record length.
Q	- makes SORT faster.
P<nnnn>=<a>	- sorts primary record containing <a> in column <n>.
P<nnnn>#<a>	- sorts primary record not containing <1a> in column <n>.
S<nnnn>=<a>	- sorts secondary record containing <a> in column <n>.
S<nnnn>#<a>	- sorts secondary record not containing <a> in column <n>.
B	- specifies binary tag format.
K	- specifies key tag format.
L	- output records limited to certain input columns/record.
T	- specifies ASCII decimal tag format.
<nnnn>-<nnnn>	- specifies sort key by column number.
X	- writes record count as the first record.
Ennnnnnn	- specifies sectors to allocated output file.
U	- determines sectors allocatd to the output file.
Fnnnnnnn	- allocates sectors to work file.
V	- determines sectors allocatd to the work file.

STARTUP UTILITY

The STARTUP utility provides the capability to store the user's current UET data in a file that can later be used to support automatic long-on or that can be loaded during a work session to restore the UET. Command lines to be executed can also be stored in this startup file. STARTUP runs in create mode or load mode, whose corresponding command lines are:

```
STARTUP  [[OUT=]<filespec>];CREATE[,<options>]
or
STARTUP  [[IN=]<filespec>] [;<options>]
```

where

[OUT=]<filespec> is the file created to store UET data.

[IN=]<filespec> specifies the file to be used to load UET data.

STARTUP provides the following options:

CREATE - allows creation of a startup file.
NORUN - suppresses execution of any stored command line during load mode.
FORCE - inserts UET entries whether or not the corresponding catalogs can be opened.
DISP - displays UET entries (except passwords) as they are inserted during local mode.
NOCHECK - does not check to see if an environment can be accessed before it is inserted in the UET.
NOERR - suppresses the display of UET data during load mode; overrides the DISP option.

The TAPEUTIL utilty allows the user to transfer any type of RMS disk data files to or from an RMS cassette, magnetic tape, or 3M cartridge.

```
TAPEUTIL  [[IN=]<filespec 1>],[VOL=]<volume>
          [,[RES=]<resource>][,[PRT=]<filespec 2> [;<options>]
```

where

[IN=]<filespec 1> - is the specification for the file to be transferred.

[VOL=]<volume> - is the name of the tape containing the file to be transferred.

[RES=]<resource> - is the name of the resource you will use to read from ●

;INFO of disk directory
DUMP /LOAD files
BACKUP/RESTORE environment
BYPASS/OVERWRIT files when load/restore
APPEND HSI on Restore
VERIFY tape/disk
EXPIRE=DDMMYYY (comment only)
RESET

TAPEUTIL UTILITY