RMS Expert
User Manual

Order Number: IDSI-RMX001

This document describes the theory, installation, and operation of RMS Expert. RMS Expert is an intelligent file performance and management tool developed by International Digital Scientific. It is intended for use by VAX/VMS System Managers.

Revision/Update Information:	This is a new manual.
Operating System and Version:	VAX/VMS Version 5.0 or later.
Software Version:	RMS Expert 1.0

International Digital Scientific Incorporated, Valencia, CA

December 7, 1990

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

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

The following are trademarks of International Digital Scientific Incorporated:

RMS Expert

The following are trademarks of Digital Equipment Corporation:

DEC
VAX
VAXnotes
VAXdocument
DECnet
VMS
All-in-One

This document was prepared using VAX DOCUMENT, Version 2.0

Contents

FIGURES

TABLES

Preface

The ***RMS Expert User Manual*** provides information needed to install and use RMS Expert. RMS Expert may be installed on any VAX/VMS Version 5.0 or later system.

Intended Audience

This manual is intended for use by VAX/VMS System Managers.

Document Structure

This manual contains the following parts:

Part 1 describes the utilities and operating theory of RMS Expert.

Part 2 describes the installation of the RMS Expert utilities and documentation from the distribution kit.

Part 3 of the document describes the use of the utilities.

Part 4 is a reference section that describes the RMS Expert commands and qualifiers.

Part 5 lists messages and any user action related to that message.

Associated Documents

The following documents, provided as part of the VAX/VMS document set, contain information related to this manual:

- *The Guide to VMS Applications* contains related information about VAX Record Management Services (RMS), File Descriptor Language (FDL), and the uses of files.
- *The Record Management Services (RMS) Reference Manual* contains information regarding RMS data structures and RMS system services.

Conventions

For all step-by-step procedures contained in this document, the following conventions are used:

- All entries have an implied carriage return following them unless some other entry (e.g., the Escape Key ESC is specified.)
- If a carriage return is the only response, it is specified as Return.
- All default entries are bracket by "< >".
- All optional entries are bracket by "[ ]".
- All mandatory entries are bracket by "( )".
- Control character are indicated by CTRL/char, i.e., CTRL/C.

- The symbol *mmun:* is used to refer to a tape unit on the system, e.g., MUA0.
- The symbol *ddun:* is used to refer to a disk unit on the system, e.g., DUA1.
- The symbol "[xxxxxx]" is used to refer to a directory, e.g., [SYS$SYSTARTUP].

Part I Operating Theory

This part contains a description of the components contained in the RMS Expert tool set. Included in this part is a overview and detailed description of the each of the utilities.

1 Introduction

This chapter introduces RMS Expert. Background information necessary for the understanding of the utilities is given along with an overview of the basic principles of operation.

1.1 Background

Under VAX/VMS, files are managed by a fully functioned but complicated Record Management System known as RMS. All Digital Equipment Corporation supplied languages support RMS ~~either~~ [illegible] by callable libraries or, in cases of languages like COBOL and PL/I, RMS is transparently utilized to implement language-generic structures.

Clarify 3 types seq ind & relative.

There are three types of VMS files managed by RMS; sequential, indexed, and relative. The most commonly used type is sequential. Most text editors and languages use a generic OPEN statement to create sequential files. Because of their simple structure, they are very easy to maintain. Indexed files are more complicated in structure, but are as important to VMS applications. Unfortunately they are also very difficult to maintain and require a high degree of tuning to be used most effectively.

RMS is tightly integrated with the VMS operating system and provides such services as record locking and transparent network (DECnet) file access. As a consequence, most serious applications written for VAX/VMS utilize RMS files one way or another. Therefore, having RMS files properly optimized and organized is fundamental for good performance. In fact, experience has shown that fine tuning RMS files can improve throughput more than all other types of system tuning combined!

Tuning an RMS index file can be a complicated and tedious task. Not only is it prone to errors, but it requires a great deal of knowledge on the part of the person doing the tuning. Digital supplies several tools to help with tuning files, but these tools are insufficient in obtaining the best possible performance. Since the tuning of files lies somewhere between System Management and Applications Programming, it is rare that the RMS indexed files on a VAX/VMS system are ever properly tuned.

1.1.1 Internal File Optimization

In addition to the obvious components of a file's structure, the File Access Blocks (FABs), Record Access Blocks (RABs), and the keys, RMS also has several hidden internal structures (These structures will be referred to as the internal layout in this document) ~~It has long been known that~~ system performance is very much dependent on how an application uses a file structure. ~~Similarly,~~ the internal layout can also drastically affect the performance.

The internal layout of the file is controlled by many inter-related factors, including such things as whether the keys and/or data are compressed; how many records fit into a single bucket, etc. The method of data access depends on whether the keys/data are compressed or not. Typically compressed keys can reduce the access throughput by a factor of 3 to 4. On the other hand "bucket size" controls the number of records which will be contiguously stored together. Larger bucket size reduces the index levels.

Define - Bucket and Index lev

When tuning RMS index files, all these factors need to be taken into consideration. A complete description of the internal structure of RMS index files is given in the VAX/VMS document, *Record Management System User's Guide.*

An RMS index file is most often created by an application program. In the cases where language generic statements are used such as in COBOL, there is a no means to control the internal layout of the files created. The language'~~s run-time library~~ uses the defaults designed for lowest common denominators, which happens to be a VAX/VMS system configured with the limited hardware of over a decade ago. In cases where an application programmer selects alternate values for the internal file layout (which has proven to be rare) by use of RMS subroutine calls, it is unlikely that optimal values are chosen since they are system dependent. Should the files or programs be moved to a different system configuration, performance degradation would probably result.

It is more efficient and almost always necessary to post-optimize files after they are created. Failure to do so can result in significant performance degradation.

1.1.2 Internal File Organization

Even after the file is properly optimized, as it gets updated, performance will eventually decrease as the file becomes internally disorganized or fragmented. Internal disorganization is caused by such things as deleted records which are marked for deletion but continue to use up space; and split buckets which cause the creation of Record Reference Vector (RRV) records and hence a layer of indirection during record access. Internal disorganization gets worse with time and can become disastrous, especially if the file was not properly optimized in the first place. In the extreme case, internal disorganization can bring an application to a stand still.

Often this "slowing down" effect is interpreted as being caused by the addition of data. "More data, slower access" sounds right. However, the purpose of an index file is to avoid slowing down as the amount of data increases. For an index file, increasing the amount of data should have a minimal effect on performance (except in the rare case where a file must be scanned sequentially). This "slow down" effect is caused by internal disorganization much more than by the amount of data.

2 Description

RMS Expert, developed by IDSI, provides a perfect solution for the problems described in Chapter 1, Introduction. It is a set of intelligent tools capable of using expert analysis to report, optimize and organize RMS index files in an automated fashion. It includes three totally independent utilities which address different aspects of RMS tuning and which, when used together, give the best record access throughput and maximum available space within RMS index files.

The utilities contained in the RMS Expert tool set may be accessed through DCL commands or a menu driven interface, REVIEW/I~~NTERACTIVE.~~ Screen mode.

The first utility, REVIEW, is used to analyze exiting RMS index files and provides a report of the files internal structures and parameters. The report recommends the actions necessary to improve file performance either for access time (thoughput) or disk utilization.

The second utility, OPTIMIZE, analyzes an exiting RMS index file and fine tunes its internal structures and parameters to cause minimum file fragmentation and best access time. It uses utilities provided with VMS along with internal analysis to perform its functions.

The third utility, ORGANIZE, does "inplace" reorganization of the index files to improve record access throughput. It eliminates the levels of indirection in record access by removing the RRV and deleted records. It also makes maximum space available for use within the file which was otherwise wasted. It accomplishes this by reclaiming the data buckets that contained only deleted and or RRV records and reclaiming space used by such records. Lastly it minimizes the levels of the index records by pushing the maximum number of index records into the index buckets and reclaims the now empty buckets.

3 Design Considerations

Throughout the design and creation of RMS Expert, the main concern was to keep the software "safe". The utilities were designed such that under any circumstances and for any reason, if an RMS Expert utility is not allowed to finish its execution (e.g., a system crash) the index file will not be corrupted. Further, ORGANIZE includes a journaling capability. In the event of a disaster during execution, file(s) can be quickly recovered.

future.

4 Theory of Operation

All VMS users use RMS index files in some form or another. Many utilities, Digital or third party, on VAX/VMS keep information in RMS index files. For example, All-in-One, VAX Mail, and VAXnotes. Addition and deletion of records, typically frequent for such files, internally fragment the files. This fragmentation automatically triggers critical performance problems, i.e., slow record access rate, wastage of disk space, etc. Therefore it is often necessary to reformat and reorder these files.

Unfortunately, due to the complex nature of RMS index files, there are no utilities available on the market which effectively maintain and cleanup RMS index files "inplace" in order to give optimum record access rate and maximum free space within the files. Digital Equipment Corporation does provide utilities to maintain these files. But most of these utilities require a very large amount of disk space to create new copies of the, often huge, files (sometimes even multiple copies of them). They also require the files to be locked against any type of access while they are in use. This is very inconvenient and often not feasible for many users. Some organizations, for instance banks and hospitals, must have access to their files at all times.

RMS Expert offers the perfect solution for these problems. It is made up of three mutually independent utilities, REVIEW, OPTIMIZE and ORGANIZE, that address the problems created by the structure and internal organization of files.

4.1 REVIEW

REVIEW analyzes exiting RMS index files and provides a report of the files internal structures and parameters. The report recommends the actions necessary to improve file performance either for access time (thoughput) or disk utilization. The expected performance gain is based on using one or both of the other utilties.

4.2 OPTIMIZE

OPTIMIZE fine tunes index file structure and parameters to provide optimal disk usage and enable minimal internal file fragmentation. It was developed with the following taken into consideration.

The internal layout of the file is controlled by many inter-related factors, including such things as whether the keys and/or data are compressed; how many records fit into a single bucket, etc. Therefore while tuning an index file, it is necessary to look at the effects of changing the values of internal file parameters.

The method of data access depends on whether the keys/data are compressed or not. Typically if there is no compression involved, records are searched by binary search methods which can be faster by a factor 3 to 6 than the linear search which is adopted if the keys are compressed. Extra time is spent in decompressing all the keys before finding the correct key. Therefore in order to decide if the keys/data should be compressed, it is necessary to ensure that after doing the compression, a significiant amount of disk space will be saved.

On the other hand "bucket size" controls the number of records which will be contiguously stored together. Larger bucket size reduces the index levels. But if key compression is enabled, more time will be spent in doing the linear search of the records within the buckets than the time spent going through the levels of index buckets. Therefore a cautious decision must be made on choosing the proper bucket size.

OPTIMIZE analyzes an existing RMS index file and checks the values of file parameters. It checks if the bucket size specified contains enough space for extra records so that there will not be too many index levels which will slow down the record access. It also checks if key/index/data compression would effectively benefit the file structure, i.e., if the compression would in fact save considerable amount of the disk space. Depending on the state of internal fragmentation of the file, it also checks if some extra space should be left (called the "fill amount") within the data buckets so that bucket splits will be minimized when new records are added to the file.

OPTIMIZE performs this analysis before creating a new version of the file with optimal parameters. It uses Digital provided utilities to perform some of its functions, while doing it's own validity checking to ensure that it works only on index files, that system files are excluded, and when the new file, with optimum structure is created, contains the proper data.

OPTIMIZE may be used in one of two manners to optimize the file for minimum disk useage or throughput. By default, it optimizes the file for performance. It accepts the /PREFER=SPEED or /PREFER=SPACE qualifier in order that a user can specify if performance or disk usage is the most critical. If the /PERFER=SPACE qualifier had been specified and the key/data compression was found not to have saved much disk space, OPTIMIZE disables the compression (i.e., /PREFER=SPEED is used).

Unlike the VAX/VMS utilities (e.g., CONVERT , ANALYZE, etc.), OPTIMIZE accepts wildcard characters in its file specifications. It recognizes RMS Prolog 3 index files and works only on files that match the file specification that are not system files. This makes it easy for a System Manager to OPTIMIZE all the index files on a particular disk. The only command needed is:

```
$ OPTIMIZE ddun:[*...]*.*;*
```

If this command is repeated for all the disks on a system, all the index files will be optimized. Since immediately after optimizing a file, the software makes sure that the new version of the file still contains valid data, there is no risk in using the optimizer overnight without operator's intervention. This utility can be used interactively, as a detached process, or in batch mode.

4.3 ORGANIZE

The ORGANIZE utility provides internal defragmentation of the file to give best record access throughput and maximum available space within the file. It allows "inplace" reorganization of the RMS index files without making the files inaccessible. It reclaims (makes available for reuse) the empty buckets and removes the bucket splits by removing RRV records. It also cleans up the data buckets by removing deleted records, thereby allowing free space within the file for more records without having to extend the file. If this utility is used along with a disk defragmentor, the disk and files will effectively be totally defragmented.

In order to really appreciate what RMS Expert does to index files, it is necessary to understand the structure of index files and how they are handled by VMS.

Records in an index file are stored in a storage entity, called a bucket, which is an I/O unit for index files. Buckets may include one or more records and are of two types; Index buckets and Data buckets. In form, these two bucket types are in a tree structure for each key defined in the file. Level 0 of the tree contains data buckets. The primary key data buckets contain actual data, while the data buckets of all the other keys contain pointer records, known as Secondary Index Data Records (SIDRs). They contain an alternate key value and one or more pointers to use data records in the primary level 0 that contain the key value. The root bucket, the highest level, points to next lower level index buckets which in turn eventually point to the data buckets at lowest level, level 0.

All the data records with a close range of key values are grouped in the same bucket. As more records are added to the index file, these buckets get filled and at some point a new bucket needs to be added to store the extra records. This is called a bucket split.

Over a period of time, a chain of split buckets for the same range of primary key values are created. Bucket splits cause the creation of RRV records which are pointer records to the actual data records now located in the new split bucket. This overhead, 9 bytes per record, causes an extra level of access time.

Record access time and disk use optimization improve by avoiding bucket splits. Record access will therefore be improved with minimum levels of the index buckets.

The ORGANIZE utility uses the following approach for file reorganization.

- It updates the pointer records (SIDRs) of the alternate keys so that they point to the actual primary key data records (rather than to RRV records).
- It then removes all the RRV records and the records marked for deletion from the primary key data buckets for all keys.
- While doing this, it reclaims all the empty data buckets and makes them available for reuse.

- Finally it pushes the maximum number of index records into the index buckets and then frees up the now empty index buckets, thereby minimizing the total number of levels.

This optimizes the use of disk space and greatly improves the record access time of the Index files.

Most of the disk defragmentors available on the market relocate the files to different logical locations on a disk in order to create logically contiguous files. Therefore the ORGANIZE utility, when used with the a disk defragmentor, effectively creates a virtually as well as logically contiguous file.

Before reorganizing a file, ORGANIZE locks the file in a protected write mode so that other users can still read the file but not write to it. Buckets of the file are read into virtual memory and rearranged if necessary. The software changes the lock on the bucket to exclusive mode just before writing the bucket to the file. So there may be instances when the users may not be able to access certain buckets in the file (maximum two at a time). But this delay is very small and users should not notice it.

ORGANIZE starts reorganizing by first going through all the alternate key data buckets sequentially. It identifies the deleted records and reclaims them. If an existing alternate key data record is pointing to a RRV record in the primary data bucket, it updates the record so that it now points to the actual data record (pointed to by the RRV record). Finally it reclaims the empty data buckets which contained only deleted records.

Since all the records in alternate key trees point to the actual data records (in primary key tree), the RRV records located in the primary key data buckets can be safely deleted. ORGANIZE therefore goes through all the data buckets of the primary key sequentially and reclaims all the deleted and RRV records. It also reclaims the now empty buckets which only contain the deleted and RRV records and makes them available for reuse.

In the final phase, the ORGANIZE goes through all the index buckets (for primary key as well as the alternate keys) and rearranges them so that they contain the maximum number of index records possible. It also reclaims the now empty index buckets. It does this as follows: Starting at the lowest level (level 1) it goes through all the index buckets sequentially. It checks if all the records in the next bucket can be contained into the current bucket (on the same level). If so, it copies all of them into the current bucket and reclaims the now empty next bucket. It also deletes the index record pointing to the next bucket from the level above and updates key value of the record pointing to the current bucket.

The operation of the utility is divided into different phases which are totally independent of each other. Therefore it is not necessary to retrace changes made by an earlier phase when the execution of the utility is interrupted.

During all the phases of operation, ORGANIZE keeps track of how many buckets it traversed at each level for each key, how many were reclaimed, how many deleted records were removed, how many alternate key records were updated to point to correct primary key records (instead of RRV records; thereby improving access time), total number of bytes now available for reuse within each area, etc. At the end, it uses this information to graphically show how the disk space was utilized within the file before and after the ORGANIZE reorganized the file. It also reports the access time improvement that should be expected after the file reorganization while accessing records through different keys.

Part II Installation

This part contains the instruction for installing RMS Expert from the distribution kit to a VAX/VMS system.

5 Installation

This chapter provides a list of the kit components and information necessary for the installation of the RMS Expert utilities.

RMS Expert is made up of three mutually independent utilities. The first, REVIEW, analyzes and reports on the file structure of RMS indexed files along with the expected performance gain after using one or both of the other utilties. The second, OPTIMIZE, fine tunes index file structure and parameters to provide optimal disk usage and enable minimal file fragmentation. The third, ORGANIZE, provides inplace, internal defragmentation of the file to give the best record access throughput and maximum usable space within the file. The utilities may be access through DCL commands or a menu driven interface, REVIEW/~~INTERACTIVE~~.

RMS Expert is fully operational and supported on VAX/VMS Version 5.0 and higher. Installation is accomplished using the VMSINSTAL command procedure supplied with VAX/VMS.

5.1 Components

Table 5–1 lists the components that are supplied with this release of RMS Expert.

Table 5–1 Components

Part Number	Label	Type	Description
RMXV010-*media*	RMXV010	BACKUP Saveset	Installation Tape
~~IDS1-RMS0001~~		Document	*RMS Expert User Manual*

media my be any one of the following:

9T—Nine track tape
TK—TK50
5 —5¼ floppy disk
3 —3½ floppy disk

The installation tape, supplied with the RMS Expert kit, contains the OPTIMIZE, ORGANIZE, and REVIEW utilities, help files, startup and shutdown command procedures, Command Line Definition (CLD) files, and the User Manual (in BOOKREADER format) in a single saveset.

5.2 Installation Procedure

Follow the steps below to install the product.

1 Log into the SYSTEM account and invoke the system update procedure thus:

```
$ @SYS$UPDATE:VMSINSTAL RMXV010 location
```

The first parameter, RMXV010, is the product save set name. The second is the location of the product save set, either the physical-device name of the tape drive or the device and directory where the save set is located.

2 To have the installation automatically verified, answer "YES" to the question, "Run the IVP after the installation?".

3 Answer the prompts; in most cases the defaults are correct.

4 The BOOKREADER version of the User Manual is automatically copied to DECW$BOOK during the installation. To have it included the BOOKREADER library shelf, LIBRARY.DECW$BOOKSHELF, answer "YES" to the question, "Add the User Manual to the BOOKREADER Bookshelf?".

Symbols: **The Digital command procedure VMSINSTAL deletes all global symbols. Therefore, after installing the product, the installer needs to restore them to regain the users environment. This may be done in one of two ways. The first is to log off and log back in, thus restoring the local environment created by the various login command procedures. The second is to execute the system and user login command procedures.**

```
$ @SYS$SYLOGIN:
$ @SYS$LOGIN:LOGIN.COM
```

The installation procedure creates a directory SYS$SYSDEVICE:[RMS_EXPERT] on the system disk. The logical name RMS$EXPERT is defined to point to the directory during the startup of the utilities. The following files will be found in subdirectories after the installation:

OPTIMIZE.CLD
OPTIMIZE.EXE
OPTIMIZE_SHUTDOWN.COM
ORGANIZE.CLD
ORGANIZE.EXE
ORGANIZE_SHUTDOWN.COM
REVIEW.CLD
REVIEW.EXE
RMS_EXPERT.DECW$BOOK
RMS_EXPERT.HLB
RMS_EXPERT.HLP
RMS_EXPERT_COMMANDS.COM
RMS_EXPERT_STARTUP.COM

5.3 After Installation

RMS Expert comes with a startup command procedure, command line interface, and help files. In order to use the utilities, certain system startup files must be modified.

5.3.1 RMS Expert Startup

After the product has been installed, the following line should be added to the system startup procedure:

```
$ @SYS$STARTUP:RMS_EXPERT_STARTUP.COM
```

5.3.2 Privileges

The software is intended to be used either from an account with the required privileges or as an image installed image with the required privileges (so that individual accounts will not need those privileges.) System managers must make a cautious decision on which is the best way to use this software. The installation of the utilities with privileges is left as an option of the System Manager and site requirements. In the event that the utilities are to be installed with privilege, the following should be added to the system startup command procedure after RMS Expert has been started.

```
$ INSTALL REPLACE RMS$EXPERT:[EXE]REVIEW -
  /OPEN/HEADER/SHARED/PRIV=(TMPMBX,NETMBX,CMKRNL,SYSLCK,WORLD)
$ INSTALL REPLACE RMS$EXPERT:[EXE]ORGANIZE-
  /OPEN/HEADER/SHARED/PRIV=(TMPMBX,NETMBX,CMKRNL,SYSLCK,WORLD)
$ INSTALL REPLACE RMS$EXPERT:[EXE]OPTIMIZE-
  /OPEN/HEADER/SHARED/PRIV=(TMPMBX,NETMBX,CMKRNL,SYSLCK,WORLD)
```

5.3.3 Command Line Interface Installation

The product is supplied with a DCL command line interface. Inorder to active the utilities, the DCL commands must become known to the system. This may be accomplished in one of two ways. The first is by using the SET COMMAND command. The second is by inserting the utility's CLDs in the system DCL tables.

In order to providing a single command to active the menu driven interface to the utilities, the symbol RMS_EXPERT should be equated to REVIEW/INTERACTIVE, in either the users LOGIN.COM or the system login command procedure, SYSLOGIN.COM.

```
$ RMS*_EXPERT :== REVIEW/INTERACTIVE
```

If the SET COMMAND method of using the utilities is prefered, then the following command should be placed either in the users LOGIN.COM command procedure or into the the system login command procedure, SYLOGIN.COM.

```
$ @RMS$EXPERT:[COM]RMS_EXPERT_COMMANDS.COM
```

If it is desired to insert the commands into the system DCL tables, then the following should be used:

```
$ SET DEFAULT SYS$COMMON:[SYSLIB]
$ SET COMMAND/TABLE=SYS$LIBRARY:DCLTABLES-
     /OUTPUT=SYS$LIBRARY:DCLTABLES -
     RMS$EXPERT:~~[CLD]~~REVIEW,ORGANIZE,OPTIMIZE
~~$ RENAME SYS$LIBRARY:DCLTABLES.EXE SYS$COMMON:~~
$ INSTALL REPLACE SYS$LIBRARY:DCLTABLES
```

5.3.4 Help Library Installation

The RMS Expert help library may be accessed by either defining a logical name to point to it or by inserting it into the VMS help library. To access the help library via a logical name, it must be determined what logical name to use. VMS HELP first looks in the default system help library, then in the library defined by the logical name HLP$LIBRARY, and finaly in libraries in numerical order with logical names of the form HLP$LIBRARY_*n*. First determine if the system has any help library logical names defined,

```
$ SHOW LOGICAL HLP$LIBRARY*
```

If no logical names have been defined, add the following line to the system startup command procedure:

```
$ DEFINE/SYSTEM/EXEC HLP$LIBRARY RMS$EXPERT:~~[HEL~~P]RMS_EXPERT_HLP
```

If logical names have been defined, add the following line to the system startup command procedure:

```
$ DEFINE/SYSTEM/EXEC HLP$LIBRARY_n RMS$EXPERT:[HE~~L~~P]RMS_EXPERT_HLP
```

Where *n* is the next number in the list.

If it is desired to add the RMS Expert help library to the VMS system help library, then do the following:

```
$ LIBRARY/HELP/REPLACE SYS$HELP:HELPLIB RMS$EXPERT:[HE~~L~~P]RMS_EXPERT.HLP
```

Part III Use of RMS EXPERT

This part describes the use of RMS EXPERT and shows examples of the reporting features of the utilities.

6 Using RMS Expert

RMS Expert is composed of three completely independent utilities which complement each other to give the best RMS tuning solution.

They are online utilities may be used like other VMS utilities. A set of DCL commands and qualifiers as well as online help and documentation are provided. The commands can be used with most of the qualifiers in batch mode. Alternately, the utilities my be used via a menu driven interface, REVIEW/~~INTERACTIVE~~.

Since the RMS Expert utilities accept wildcard characters in their file specifications, inplace reorganization may be done on all index files in a given directory or on a disk.

6.1 Files Supported

RMS Expert may be used on any RMS Prolog 3 index files.

6.2 Required Privileges

The privileges required to use the RMS Expert utilities depend upon the qualifier(s) selected. In order to use all qualifiers, the user must have the following privileges:

TMPMBX, NETMBX, CMKRNL, SYSLCK, WORLD

If the RMS Expert utilities are installed with the above privileges, then any non-privileged user is able to use them, see Section 5.3, After Installation.

6.3 Documentation

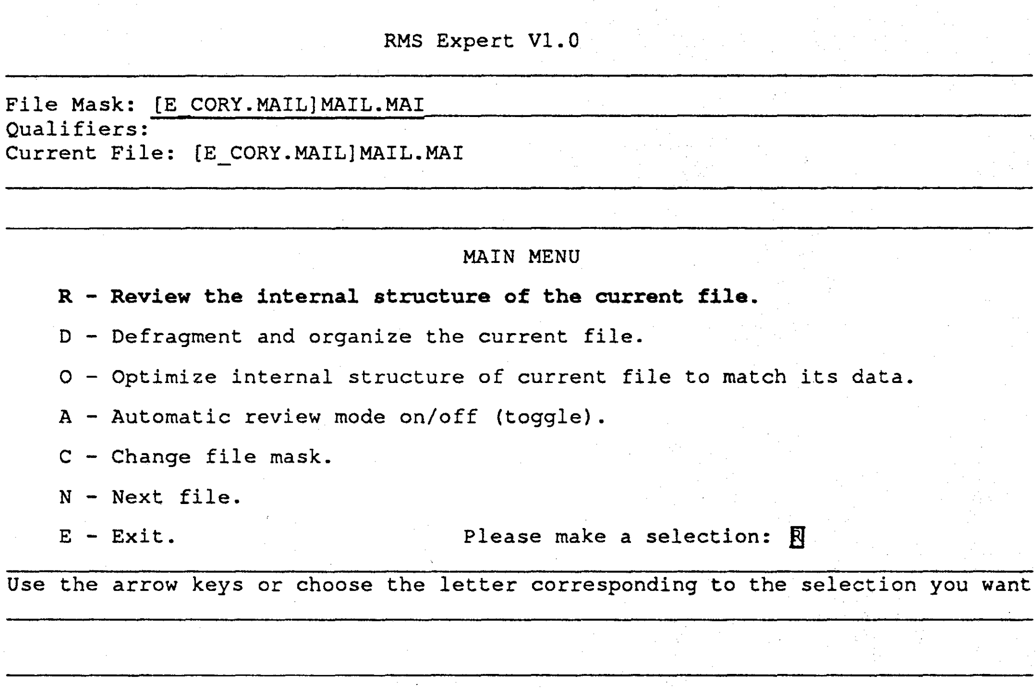
Documentation for RMS Expert is provided in both "online" and "hardcopy" formats. The "online" format is furnished as online-help and BOOKREADER manuals. During the installation, the BOOKREADER version of the manual is copied to DECW$BOOK and optionally added to the BOOKREADER Bookshelf. The online help files may be added to the system help library.

6.4 REVIEW/~~INTERACTIVE~~

RMS Expert is provided with a menu driven user interface. The menu is activated by the DCL command *$ REVIEW/INTERACTIVE*(*file-spec*

When started, REVIEW/INTERACTIVE provides the user with several options, see Figure 6–1, REVIEW/INTERACTIVE Interface.

Figure 6–1 REVIEW/INTERACTIVE Interface



```
                        RMS Expert V1.0

File Mask: [E_CORY.MAIL]MAIL.MAI
Qualifiers:
Current File: [E_CORY.MAIL]MAIL.MAI


                           MAIN MENU

    R - Review the internal structure of the current file.

    D - Defragment and organize the current file.

    O - Optimize internal structure of current file to match its data.

    A - Automatic review mode on/off (toggle).

    C - Change file mask.

    N - Next file.

    E - Exit.                  Please make a selection: R

Use the arrow keys or choose the letter corresponding to the selection you want
```

The display is divided into five regions. The first displays input file specification, command line, and current file information. Below that is an information/status line, that is used if neccessary. The thrid region is for options or reporting information. In Figure 6–1, that section shows the options available. As options are selected, the output from the various utilities will be displayed in this section. Below the options section, is an information/status line that is used to display help on using the options section. The final section is used to display any error or status messages returned after an operation is selected.

The input file is displayed in the the **File Mask:** field, any qualifiiers in the **Qualifiers:** field, the first file found is displayed in the **Current** File: field. The operation desired my be selected by moving the cursor to the desired choice or by entering the letter coresponding to the choice and pressing the Enter, Do, or Return key. The cursor may be moved by using the arrow keys, the Select key, or Tab key.

Help may be obtained on the keys by pressing the Help or PF2 key. Help on the various utilities may be obtained by pressing the Gold/Help (PF1+Help) or Gold/PF2 (PF1+PF2).

After selecting the option, the selected utility is envolked, with its report displayed in the option section of the menu. When the operation is complete, OPTIMIZE or ORGANIZE may be envolked by pressing the PF1 or PF2 keys. When the user returns to the main menu, the next file is selected and the user is presented with the options.

6.5 OPTIMIZE

It is strongly recommended that files be optimized before the ORGANIZE utility is used.

The OPTIMIZE command analyzes the layout of the specified index files and fine tunes the parameters in order to minimize the bucket splits within the file without increasing the index bucket levels of the keys. Once ORGANIZE is being used on a regular basis, the files should be OPTIMIZED, once every other month. If a disk defragmentation utility is ~~being~~ used or files are ~~being~~ restored using BACKUP, then the files will need to be OPTIMIZED less often. On the other hand, if file use is very dynamic (adding and deleting lots of records ~~very~~ frequently), then OPTIMIZE should be used more frequently.

One method to determine if files need to be optimized is to look at the levels of the ROOT_BUCKET of the keys. If the level is greater than 3, even after organizing the file, the OPTIMIZE utility needs to be run on that file. It is recommended that the REVIEW utility be used to determine if a file should be optimized.

Unlike ORGANIZE, OPTIMIZE creates a new copy of the file with new parameters. It also locks the file in exclusive mode during part of the operation. Therefore it is recommended that files be optimized during off-hours when file usage is low. It is important that there is enough disk space to contain another copy of the file temporarily. Once the new copy of the file with new parameters is created, OPTIMIZE checks the new file and makes sure that is contain all the records before deleting the original file and naming the new copy.

After a user issues the command OPTIMIZE with selected qualifiers and the file specification, the file specification is parsed and searched, the OPTIMIZE utility fine tunes all the Prolog 3 index files that match the file specification. System files, those created for VMS internal use only, are not effected. Wildcard characters may be used for any part of the file specification. For example:

```
$  OPTIMIZE   DUA0:[...]*.DAT
$  OPTIMIZE   DUA1:[...]*.*;0
```

9? P?

6.6 ORGANIZE

The ORGANIZE utility defragments ~~the~~ index files inplace. While file organization is taking place, other users are ~~only~~ prevented from accessing the file for a very short time. During this time, the file is write locked. Therefore, file organization should be scheduled during a time when users do not need to update or add records to the file(s). This restriction will be removed in a later release. ORGANIZE may be run at any time of the day without worrying about it using up disk space.

It Is recommended that that ORGANIZE be run once a day. If files are ~~do~~ not ~~get~~ updated frequently, then it may be run less frequently, e.g., every other day or even once a week. On the other hand, if files are very dynamic, ORGANIZE should be run more frequently, e.g., every 6 hours, by using the qualifiers /DELAY and /PASSES. In reality, most users have a combination of files. In such cases, a suggested way to run ORGANIZE on a disk is as follows:

```
$ ORGANIZE~~/AT="5:00:00"/REPEAT="24:00"~~  DUA1:*.*
$ ORGANIZE~~/AT="11:00"~~/SINCE=TODAY~~/PASSES=3/DELAY="6:00:00"~~  DUA1:*.*
```

The first command makes sure that every index level 3 file on the disk DUA1: is organized~~at least once a day.~~ The second command will organizes only those files which were modified today. ~~This organization takes place every 6 hours.~~

After running ORGANIZE a few times on a system, the user will be able to judge how often to ORGANIZE files. This information may be gathered using the /REPORT qualifier during the first few runs. Careful use of a combination of qualifiers will satisfy the needs of many different environments.

6.7 Reporting Features of RMS Expert

The REVIEW utility produces a report that analyzes the internal structure of RMS index files and recommends actions to improve that structure. The OPTIMIZE utility does not generate any reports. It provides informational messages only. ORGANIZE will produce a report upon demand that displays the actions taken by the utility. Both a hardcopy and graphical display are available depending upon the choice of output.

6.7.1 Reporting Features of REVIEW

The REVIEW utility generates a single page report, which describes the internal file structure and predicts performance improvement and disk space reduction achieved by OPTIMIZE and ORGANIZE. The location of the report is controlled by the /OUTPUT[=*file-spec*] qualifier. If the /OUTPUT qualifier is used without a file specification, the output is displayed on the users terminal.

REVIEW concisely informs the user about the extent of optimization and reorganization that is necessary to be performed on file(s). It shows how the space within a file is utilized. It reports the percentage of the data access improvement or disk space saving that should be expected after using OPTIMIZE and/or ORGANIZE. Further, it gives the statistics on the number of blocks to be reclaimed, total number of bytes to be reclaimed in ~~the different areas~~ each key, etc.

The reported information is useful in determining weather a particular file should be resturctured or reorganinzed. It helps in deciding how often the other utilities should be used and how useful it is for the system. This utility should be used to see the level of fragmentation within an index file before actually performing any other operation on that file.

When a user issues the command REVEIW, the utility parses and searches for the specified file(s). It then reorganizes all the Prolog 3 index files that match the file specification. Wildcard characters may be used for any part of the file specification, for example:

```
$ REVIEW/OUTPUT=REVIEW.REPORT DUA0:[...]*.DAT
```

The report shows statistics for each key before reorganization. It also shows the expected improvement in record access and disk utilization after each of the utilites is run. A sample of the report may be seen in Figure 6–2, REVIEW Report.

Figure 6–2 REVIEW Report

```
IDSI --- RMS EXPERT           Index File Review            10-DEC-1990
----------------------------------------------------------------------
       File Mask: []*.DAT
    Current File: TEST_FILE.DAT
----------------------------------------------------------------------
                          Structure Analysis

        Number of Keys: 2,  Number of Areas: 3

    Primary Key:
        Key Size = 32 bytes,  Record Size = 512 bytes
                                        Current        Target
                                        -------        -------
        Index Levels                       4              2   (Levels)
        Bucket Size                        4              8   (Blocks)
        Key Compression          enabled  21%   disabled  0%
        Record Compression       enabled  33%   disabled  0%
        Data Fill                         78%            50%
        Index Fill                        99%            75%
        ----------------------------------------------------
        Estimated Average Access Improvement >200%
        Every 100 I/O operations would take 25 I/O operations
        ----------------------------------------------------

    Alternate Key 1:
        Key Size = 32 bytes,  Record Size = 512 bytes
                                        Current        Target
                                        -------        -------
        Index Levels                       4              2   (Levels)
        Bucket Size                        4              8   (Blocks)
        Key Compression          enabled  21%   disabled  0%
        Data Fill                         78%            50%
        Index Fill                        99%            75%
        ----------------------------------------------------
        Estimated Average Access Improvement >200%
        Every 100 I/O operations would take 25 I/O operations
        ----------------------------------------------------
```

Figure 6–2 Cont'd on next page

Figure 6–2 (Cont.) REVIEW Report

```
Alternate Key 2:
    Key Size = 32 bytes,  Record Size = 512 bytes
                                    Current         Target
                                    -------         ------
    Index Levels                       4               2    (Levels)
    Bucket Size                        4               8    (Blocks)
    Key Compression          enabled  21%   disabled  0%
    Data Fill                         78%             50%
    Index Fill                        99%             75%
    ---------------------------------------------------
    Estimated Average Access Improvement >200%
    Every 100 I/O operations would take 25 I/O operations
----------------------------------------------------------------
Overall Expected Average Improvement    >123%
Overall File Size Change                 +12%
----------------------------------------------------------------

                     File Organization
Primary Key:
                                    Current         Target
                                    -------         ------
    Data  Blocks                       42
    Index Blocks                        3
    Levels                              2
    Number of Data Record:
        Live                          857
        RRV                            39
        Deleted                       318
    Live Records                   138645
    Dead/RRV Records                 7033
    ---------------------------------------------------
    Expected Performance Improvement:
        Defragmentation in key structure       82%
        Data access improvement for this key  302%
    ---------------------------------------------------

Alternate Key 1:
                                    Current         Target
                                    -------         ------
    Data  Blocks                        0
    Index Blocks                        0
    Levels                              1
    Number of Data Record:
        Live                           15
        RRV                             0
        Deleted                        15
    Live Records                     2769
    Dead/RRV Records                12830
    ---------------------------------------------------
    Expected Performance Improvement:
        Defragmentation in key structure        0%
        Data access improvement for this key  205%
    ---------------------------------------------------
```

Figure 6–2 Cont'd on next page

Figure 6–2 (Cont.) REVIEW Report

```
Alternate Key 2:
                                   Current         Target
                                   -------         ------
    Data  Blocks                     42
    Index Blocks                      3
    Levels                            2
    Number of Data Record:
        Live                        857
        RRV                          39
        Deleted                     318
    Live Records                 138645
    Dead/RRV Records               7033
    -----------------------------------------------
    Expected Performance Improvement:
        Defragmentation in key structure       82%
        Data access improvement for this key  302%
-----------------------------------------------------------------
Overall Expected Improvement
    Internal File Fragmentation          12%
    Data Access Improvement             253%
-----------------------------------------------------------------

            Summary For File: TEST_FILE.DAT
                 Expected improvement

                                        Thoughput  Disk Space
    ~~OPTIMIZE/PERFER=SPEED                    800%          3%~~
    ~~OPTIMIZE/PERFER=SIZE                      20%         40%~~
    ORGANIZE                                150%         10%
    OPTIMIZE~~/PERFER=SPEED~~ and ORGANIZE      950%          7%
    OPTIMIZE/PERFER=SIZE and ORGANIZE       ~~170%         50%~~
```

6.7.2 Reporting Features of ORGANIZE

The ORGANIZE utility generates a three page report, which graphically describes the performance improvement achieved by ORGANIZE. The location of the report is controlled by the /OUTPUT[=*file-spec*] qualifier. If the /OUTPUT qualifier is used without a file specification, the output is displayed on the users terminal.

The ORGANIZE reporting facility concisely informs the user about the extent of reorganization that was performed on the file(s). It graphically shows how the space within the file was utilized before and after the reorganization. It also reports the percentage of the data access improvement that should be expected for each key due to the reorganization. Further, it gives the statistics on the number of blocks reclaimed, total number of bytes reclaimed in the different areas, etc.

The reported information is useful when using the utility in batch mode or as a background job. It helps in deciding how often the utility should be used and how useful it is for the system. In fact, this utility may be used to see the level of fragmentation within an index file (/REPORT=ONLY) before actually performing the reorganization of the file.

When a user issues the command ORGANIZE, the utility parses and searches for the specified file(s). It then reorganizes all the Prolog 3 index files that match the file specification. Wildcard characters may be used for any part of the file specification, for example:

```
$  ORGANIZE/REPORT/OUTPUT=ORGANIZE.REP  DUA0:[...]*.DAT
```

The first page of the report is a bar chart showing record access time for each key before and after reorganization. It also shows the percentage improvement in record access for each key. A sample of the first page of the report may be seen in Figure 6–3, ORGANIZE Report - Page 1.

Figure 6–3 ORGANIZE Report - Page 1

```
IDSI --- RMS EXPERT          ORGANIZE          10-DEC-1990 Page 1.
----------------------------------------------------------------------

          Record Access Times Before and After Reorganization

                        For file: TEST_FILE.DAT

A 100|    XX       XX       XX       XX
C    |    XX       XX       XX       XX
C  90|    XXOO     XX       XX       XX
E    |    XXOO     XX       XX       XX
S  80|    XXOO     XX       XX       XX
S    |    XXOO     XX       XXOO     XX
   70|    XXOO     XX       XXOO     XX
I    |    XXOO     XX       XXOO     XX
M  60|    XXOO     XX       XXOO     XX
P    |    XXOO     XX       XXOO     XX
R  50|    XXOO     XXOO     XXOO     XX
O    |    XXOO     XXOO     XXOO     XX
V  40|    XXOO     XXOO     XXOO     XX
E    |    XXOO     XXOO     XXOO     XX
M  30|    XXOO     XXOO     XXOO     XXOO
E    |    XXOO     XXOO     XXOO     XXOO
N  20|    XXOO     XXOO     XXOO     XXOO
T    |    XXOO     XXOO     XXOO     XXOO
   10|    XXOO     XXOO     XXOO     XXOO
     |    XXOO     XXOO     XXOO     XXOO
     +----------------------------------------------------------------
          Key0     Key1     Key2     Key3
         13.14%   99.85%   31.08%  179.64%

                                                   Before: XX
                                                    After: OO
```

The second page of the report displays the layout of the file before and after ORGANIZE has been run. It shows the file division, by percentage, for different parts of the file. These parts include those that had live data, parts which had deleted and RRV records, the parts which were empty but usable and did not need to initialized (i.e., parts of the currently used buckets which are not yet filled), and finally, parts which had never been used (empty buckets within the areas). Further, it gives the percentage of space reclaimed by ORGANIZE within the file. A copy of a typical second page of the report may be seen in Figure 6–4.

Figure 6–4 ORGANIZE Report - Page 2

```
IDSI --- RMS EXPERT            ORGANIZE                10-DEC-1990 Page 2.
--------------------------------------------------------------------------

               File Layout Before and After Reorganization

                         For file: TEST_FILE.DAT

                           Percentage of File
                  Live: (l)  Containing actual records.
                  Dead: (d)  Containing dead/RRV records.
                Usable: (u)  Currently empty but usable.
                 Empty: ( )  Which has never been used.

                                  BEFORE
 +--------------------------+-----+-----------------------------------+
 |llllllllllllllllllllllllll|ddddd|uuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuu|
 |llllllllllllllllllllllllll|ddddd|uuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuu|
 |llllllllllllllllllllllllll|ddddd|uuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuu|
 +------+------+------+-----++----+-+------+------+-----+------+------+
 0     10     20     30     40    50     60     70     80     90   100
                             AFTER
 +--------------------------+------------------------------------+----+
 |llllllllllllllllllllllllll|uuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuu|    |
 |llllllllllllllllllllllllll|uuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuu|    |
 |llllllllllllllllllllllllll|uuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuuu|    |
 +------+------+------+-----+-------+------+------+-----+------+-+----+
 0     10     20     30     40    50     60     70     80     90   100

               ***  Percentage of the file reclaimed: 12.60%  ***
```

The third page of the report shows a list of improvements achieved by ORGANIZE. See Figure 6–5.

Figure 6–5 ORGANIZE Report - Page 3

```
IDSI --- RMS EXPERT            ORGANIZE                10-DEC-1990 Page 3.
--------------------------------------------------------------------------

                                   Improvements

                              For file: TEST_FILE.DAT

        Keys Traversed                                     -      4
        Areas Traversed                                    -      3

        Primary Index Blocks Traversed                     -     28
        Primary Data  Blocks Traversed                     -   4107
        Primary Index Blocks Reclaimed                     -      0
        Primary Data  Blocks Reclaimed                     -    342

        Alternate Index Blocks Traversed                   -     36
        Alternate Data  Blocks Traversed                   -   1701
        Alternate Index Blocks Reclaimed                   -      8
        Alternate Data  Blocks Reclaimed                   -      0

        Bytes Reclaimed In Primary   Key Data  Area        - 159519
        Bytes Reclaimed In Alternate Key Data  Area        -      0
        Bytes Reclaimed In Primary   Key Index Area        -      5
        Bytes Reclaimed In Alternate Key Index Area        -   4144

        Number Of Levels Reclaimed In Primary   Tree       -      0
        Number Of Levels Reclaimed In Alternate Tree(s)    -      0
```

6.8 Modes of Operation

FIX

The RMS Expert may be operated in one of three modes. The operational mode is selected by use ~~of the /MODE=*keyword*~~ a qualifier. Each of the modes is discussed in the following sections.

6.8.1 Detached Process Operation

Detached is ~~an ideal~~ the preferred way of using ORGANIZE. When the command

```
$ ORGANIZE/CONCURRENT=n/Detach. file-spec
```

is given, *n* detached processes are created. Each processes a file which matches the file specification. In this mode, users are able to process several files in parallel.

It provides continuous internal defragmentation of files (on the disk) without adversely impacting the normal system activities. Since ORGANIZE does not need to lock any files in an exclusive mode (although it does lock few records of the files from time to time for a very short period which may occasionally cause marginal record access delay) other applications needing the file being reorganized can read it concurrently. Typically System Managers use ORGANIZE in this mode by adding a command to a system startup command procedure that invokes ORGANIZE at particular time of the day. Then, depending on how many passes have been specified, ORGANIZE will start itself after a specified interval. In a single run, it processes all the specified files (or all the index files on the given disk/directory if only disk/directory name is given). It hibernates between runs.

6.8.2 Batch Operation

Future FIX

When it is desirable to schedule operation around other events on the system, or if ORGANIZE is to be run only once a day, then it should be run in batch mode. To invoke ORGANIZE in batch mode, use a command as follows or add it to a system batch command procedure:

```
$ ORGANIZE/MODE=BATCH file-spec
```

6.8.3 Interactive Operation

In order to monitor the reorganization of a particular file, the INTERACTIVE mode should be used. To use this mode, specify the /NOBATCH ~~/MODE=INTERACTIVE~~ qualifier on the command line.

6.9 Shutting Down RMS Expert

Under normal conditions, there is no need to shut down any of RMS Expert processes. However both OPTIMIZE and ORGANIZE can be shut down as follows:

```
$ @RMS$EXPERT:ORGANIZE_SHUTDOWN [mode]
$ @RMS$EXPERT:OPTIMIZE_SHUTDOWN [mode]
```

where *mode* is IMMEDIATE or ORDERLY (default)

The IMMEDIATE mode aborts the operation after any pending I/O operations are completed. The ORDERLY mode finishes the reorganization of the current file before exiting.

Warning: Do not stop the RMS Expert by using the DCL command:

```
$ STOP/ID=pid
```

This may cause a file to become corrupted. If this should occur, the file may be returned to its previous state by using the journaling features, see Section 6.11, Journaling.

6.10 Performance Considerations

RMS Expert typically requires only a few seconds of CPU time per file per pass. This time can be reduced even further if a full file specification is given. Also, if the number of passes on the index files (typically one pass for less active files and two or three for more active files per day) is optimally specified, unnecessary processing effort will be saved. This optimization may be determined after using ORGANIZE a few times.

When multiple files are processed by RMS Expert (either OPTIMIZE or ORGANIZE), the qualifier /CONCURRENT=*n* should be chosen such that *n* files will be processed at a time. This is accomplished by creating *n* detached processes for each file that matches the file specification. Each of these detached processes reorganizes the given file in place and if requested, creates a report. This feature ensures that no more than *n* files are being reorganized at a time on the system by a user. By default, if the /CONCURRENT qualifier is chosen without specifying a number, one file is processed at a time.

Note: **ORGANIZE command completely reorganizes the file. That is, if a file is not updated between two passes of ORGANIZE, the second pass will not make any changes to the file even though it takes some processing time. This second pass time will be a lot less than the first pass. Therefore, avoid running ORGANIZE in multiple passes on files which are not updated frequently. This can be accomplish by choosing the qualifiers /BEFORE and /SINCE appropriately.**

The processing time for ORGANIZE depends on the state of the internal fragmentation of the file as well as the size of the file. Therefore typically when ORGANIZE is used for the first time on files, it will take longer time than subsequent runs. Processing time is also a good indication in judging how often ORGANIZE should be run. If an index file of small size takes too long to ORGANIZE, it should probably be ORGANIZE more often. On the other hand if a fairly large file takes a short time to process, ORGANIZE is probably being run too often.

6.11 Journaling

Future

Most of the time there will be no need to recover in the event of a system crash or abnormal image termination. As a safety precaution, RMS EXPERT provides a journaling capability.

While using OPTIMIZE there is no risk of corrupting a file in the event of a system crash or abnormal image termination, since OPTIMIZE ensures that the new version of the file is good and contains valid data before deleting the old version.

In the case of ORGANIZE, it is possible to have a corrupted bucket if the system crashes while the updated bucket is being written to the file. In such an event, after the system reboots, use:

```
$ ORGANIZE/RECOVER  file-spec
```

in order to make sure that no information is lost. ORGANIZE saves the original bucket contents and the key index values before making changes and therefore can recover from such circumstances.

6.12 Restrictions

As a safety precaution, this utility will not operate on the system files if the user specifies the system disk as the target disk for the operation.

Both OPTIMIZE and ORGANIZE change the Record File Address (RFA) values of the records while processing. Therefore files being accessed through RFA values should not be processed by either of these commands.

During a ~~this time is doing~~ file organization, the file is write locked and therefore, other users are prevented from accessing the file for a very short time. This restriction will be removed in a later release.

Part IV RMS EXPERT Commands

This part contains all the commands available to users of the RMS Expert utilities.

OPTIMIZE

This utility optimizes RMS index file(s) in order to allow optimal disk usage and record access.

FORMAT

OPTIMIZE *file-spec[, . . .]*

Command Qualifiers	Defaults
/AFTER[=time]	*None.*
/BEFORE[=time]	*None.*
/CONCURRENT[=n]	*/CONCURRENT=1*
/DETACH	*/NODETACH*
/LOG	*/NOLOG*
/PAGE	*/NOPAGE*
N ← */PREFER[=keyword]*	*/PREFER=SPEED*
/SINCE[=time]	*None.*

restrictions

As a safety precaution, this utility will not operate on the system files if the user specifies the system disk as the target disk for the operation.

OPTIMIZE changes the Record File Address (RFA) values of the records while processing. Therefore files being accessed through RFA values should not be processed by this command.

prompts

_File: file-spec[, . . .]

PARAMETERS

file-spec[, . . .]

Specifies one or more index file(s) that needs to be optimized. The syntax of a file specification determines which files will be processed, as follows:

- If you do not enter a file specification, the OPTIMIZE command processes all versions of the files in the current default directory.
- If you specify only a device name, the OPTIMIZE command uses you default directory specification.
- Whenever the file specification does not include a file name, file type and a version number, all versions of all files in the specified directory are processed.
- If a file specification contains a file name or a file type, or both, and no version number, the OPTIMIZE command processes all versions.
- If a file specification contains only a file name, the OPTIMIZE command processes all files in the current default directory with that file type, regardless of file type and version number.

- If a file specification contains only a file type, the OPTIMIZE command processes all files in the current default directory with that file type, regardless of file name and version number.

Wildcard characters can be used in the directory specification, file name, file type, or version number fields of a file specification to process all files that satisfy the components you specify. Separate multiple file specifications with either commons or plus signs.

Note: **If wildcard characters are given for the file name and/or file type specifications, all the index files that match the rest of the specification will be affected.**

DESCRIPTION

The OPTIMIZE utility analyzes an exiting RMS index file and fine tunes its internal structures and parameters to cause minimum file fragmentation and best access time. It uses utilities provided with VMS to perform some of its functions.

QUALIFIERS

/AFTER=time
/NOAFTER

Requests that the processing be held until after a specific time. If the specified time has already passed, the processing is started immediately.

Time can be specified as either an absolute time or as combination of absolute and delta times. See the *VMS DCL Concepts Manual* for complete information on specifying time values.

You may typically want to OPTIMIZE the files at midnight or during lunch hour when other applications may not need the index files.

/BEFORE[=time]

Selects only those files dated prior to the specified time. You can specify time as an absolute time, a combination of absolute and delta times, or as one of the following keywords: TODAY (default), TOMORROW, or YESTERDAY.

/CONCURRENT[=n]
/NOCONCURRENT

This qualifier indicates how many detached processes should be generated at a time each optimizing one file that matches the file specification. You probably will like to start more than one concurrent processes at a time when you scheduling the OPTIMIZE during the less busy time. Since each detached process works independently, you can start any number of detached processes. But if you are running OPTIMIZE during busy time, you should avoid using *n* greater than 5 since each process takes some CPU time and too many of these processes may affect other applications performances.

By default, when only /CONCURRENT is specified, /CONCURRENT=1 is used.

/DETACH
/NODETACH (default)

When this qualifier is used, a detached process is created. When used in conjunction with the /CONCURRENT=*n* qualifier, *n* detached processes are created. Each processes a file which matches the file specification. In this mode, users are able to process several files in parallel.

/LOG
/NOLOG (default)

Controls if the output of the command is displayed.

/PAGE
/NOPAGE (default)

Controls whether ouput from the OPTIMIZE command is displayed one screen at a time.

N ←

/PREFER[=keyword]

Allows you to select the method in which to optimize files. Select one of the following keywords:

SPACE	Use to select disk usage as the critical factor. By default, if the key/data compression was not found to have saved much disk space, OPTIMIZE disables the compression (i.e., /PREFER=SPEED is used).
SPEED(default)	Use to select the speed of the application as the critical factor.

/SINCE[=time]

Selects only those files dated after the specified time. You can specify time as an absolute time, a combination of absolute and delta times, or as one of the following keywords: TODAY (default), TOMORROW, or YESTERDAY. Time must be specified using the VMS format of DD-MMM-YYYY HH:MM:SS.CC; international date/time formats will not be accepted by the system. See the *VMS DCL Concepts Manual* for complete information on specifying time values.

EXAMPLES

1

```
$ OPTIMIZE/CONCURRENT=6/SINCE="10-DEC-1990" DUA0:[DBA.PROJECTA]*.DAT
```

In this example, OPTIMIZE searches and optimizes all index files with a file type of DAT in the directory DUA0:[DBA.PROJECTA] that have been created/updated after December 10, 1990.

2

```
$ OPTIMIZE DUA0:[DBA...]*.*
```

Here OPTIMIZE searches and optimizes all the index files under the root directory DUA0:[DBA].

3

```
$ OPTIMIZE DUA0:[*...]
```

In this example, OPTIMIZE searches all the index files on the disk DUA0: For each index file found, it performs a file optimization.

ORGANIZE

This utility reorganizes RMS index file(s) in order to allow optimal disk usage and record access.

FORMAT

ORGANIZE *file-spec[, . . .]*

Command Qualifiers	**Defaults**
/AFTER[=time]	*None.*
/BEFORE[=time]	*None.*
/CONCURRENT[=n]	*/CONCURRENT=1*
/DETACH	*/NODETACH*
/LOG	*/NOLOG*
/MODE[=type]	*/MODE=DETACH*
/OUTPUT[=file-spec]	*None.*
/PAGE	*/NOPAGE*
/REPORT[=type]	*/REPORT=AFTER*
/SINCE[=time]	*None.*

restrictions

As a safety precaution, this utility will not operate on the system files if the user specifies the system disk as the target disk for the operation.

ORGANIZE changes the Record File Address (RFA) values of the records while processing. Therefore files being accessed through RFA values should not be processed by this command.

prompts

_File: file-spec[, . . .]

PARAMETERS

file-spec[, . . .]

Specifies one or more index file(s) that needs to be organized. The syntax of a file specification determines which files will be processed, as follows:

- If you do not enter a file specification, the ORGANIZE command processes all versions of the files in the current default directory.
- If you specify only a device name, the ORGANIZE command uses you default directory specification.
- Whenever the file specification does not include a file name, file type and a version number, all versions of all files in the specified directory are processed.
- If a file specification contains a file name or a file type, or both, and no version number, the ORGANIZE command processes all versions.
- If a file specification contains only a file name, the ORGANIZE command processes all files in the current default directory with that file type, regardless of file type and version number.

- If a file specification contains only a file type, the ORGANIZE command processes all files in the current default directory with that file type, regardless of file name and version number.

Wildcard characters can be used in the directory specification, file name, file type, or version number fields of a file specification to process all files that satisfy the components you specify. Separate multiple file specifications with either commons or plus signs.

Note: **If wildcard characters are given for the file name and/or file type specifications, all the index files that match the rest of the specification will be affected.**

DESCRIPTION

The ORGANIZE utility performs inplace reorganization of RMS Index Files without locking the files in an exclusive mode. It does this by reordering the records within the file buckets. Further, it pushes the maximum number of records into the buckets and then frees up the higher level buckets which are then empty. It also cleans up the data buckets which contain deleted records and releases them to the free bucket list. This allows free space with in the file for more records without having to extend the file.

COMMAND QUALIFIERS

/AFTER[=time]
/NOAFTER

Requests that the processing be held until after a specific time. If the specified time has already passed, the processing is started immediately.

Time can be specified as either an absolute time or as combination of absolute and delta times. See the *VMS DCL Concepts Manual* for complete information on specifying time values.

You may typically want to ORGANIZE the files at midnight or during lunch hour when other applications may not need the index files.

/BEFORE[=time]

Selects only those files dated prior to the specified time. You can specify time as an absolute time, a combination of absolute and delta times, or as one of the following keywords: TODAY (default), TOMORROW, or YESTERDAY.

/CONCURRENT[=n]
/NOCONCURRENT

This qualifier indicates how many detached processes should be generated at a time each optimizing one file that matches the file specification. You probably will like to start more than one concurrent processes at a time when you scheduling the ORGANIZE during the less busy time. Since each detached process works independently, you can start any number of detached processes. But if you are running ORGANIZE during busy time, you should avoid using *n* greater than 5 since each process takes some CPU time and too many of these processes may affect other applications performances.

By default, when only /CONCURRENT is specified, /CONCURRENT=1 is used.

/DETACH
/NODETACH (default)

When this qualifier is used, a detached process is created. When used in conjunction with the /CONCURRENT=*n* qualifier, *n* detached processes are created. Each processes a file which matches the file specification. In this mode, users are able to process several files in parallel.

/LOG
/NOLOG (default)

Controls if the output of the command is displayed.

/OUTPUT[=file-spec]
/NOOUTPUT

Controls where the output of the command is sent. By default, the display is written to the current SYS$OUTPUT device. No wildcard characters are allowed.

If you enter /OUTPUT with a partial file specification (for example, /OUTPUT=[ADAMS]), ORGANIZE is the default file name and REPORT the default file type. If you enter/NOOUTPUT, output is suppressed.

The /REPORT qualifier must be used with the /OUTPUT qualifier.

/PAGE
/NOPAGE (default)

Controls whether ouput from the ORGANIZE command is displayed one screen at a time.

/REPORT[=keyword]
/NOREPORT

Allows you to select the type of report to produce. Select one of the following keywords:

AFTER (default)	This keyword is to produce a report of the file organization after a reorganization is performed.
ALL	This keyword is to produce a report of the file organization before and after a reorganization is performed.
BEFORE	This keyword is to produce a report of the file organization before a reorganization is performed.
ONLY	This keyword is to produce a report of the file organization before a reorganization is performed. When this keyword is specified, no reorganization takes place.

/SINCE[=time]

Selects only those files dated after the specified time. You can specify time as an absolute time, a combination of absolute and delta times, or as a one of the following keywords: TODAY (default), TOMORROW, or YESTERDAY. Time must be specified using the VMS format of DD-MMM-YYYY HH:MM:SS.CC; international date/time formats will not be accepted by the system. See Section 1.4 of the *VMS DCL Concepts Manual* for complete information on specifying time values.

EXAMPLES

1

```
$ ORGANIZE/LOG/SINCE="10-DEC-1990" DUA0:[DBA.PROJECTA]*.DAT
```

In this example, ORGANIZE searches and reorganizes all index files with a file type of DAT in the directory DUA0:[DBA.PROJECTA] that have been created/updated after December 10, 1990. It reorganizes each one inplace and gives a visual report that shows the structure of file before and after the reorganization. It also shows how much space it has made available for reuse and how much access improvement the user should expect because of the reorganization. The information is reported to the user's screen.

2

```
$ ORGANIZE DUA0:[DBA...]*.*
```

Here ORGANIZE searches and reorganizes all the index files under the root directory DUA0:[DBA]. For the each index file found, it analyzes the file and creates the internal fragmentation report. The report is displayed on the users' screen and confirms that the user does in fact want to reorganize the file. Then it does the inplace reorganization and presents the log of its action (on the user's screen). It does this for each index file found under the given directory. You can abort this operation at any time by pressing CTRL/U.

3

```
$ ORGANIZE DUA0:[*...]
```

In this example, ORGANIZE searches all the index files on the disk DUA0: For each index file found, it does an inplace reorganization.

REVIEW

This utility analyzes RMS index file(s) and reports various internal file statistics. The report shows performance gains that can be expected if one or both of the OPTIMIZE and ORGANIZE utilities were used to restructure and reorganize the file(s).

FORMAT

REVIEW *file-spec[, . . .]*

Command Qualifiers	**Defaults**
/AFTER[=time]	*None.*
/BEFORE[=time]	*None.*
/LOG	*/NOLOG*
/OUTPUT[=file-spec]	*None.*
/PAGE	*/NOPAGE*
/SCREEN_MODE	*/NOSCREEN_MODE*
/SINCE[=time]	*None.*

restrictions

As a safety precaution, this utility will not operate on the system files if the user specifies the system disk as the target disk for the operation.

prompts

_File: file-spec[, . . .]

PARAMETERS

file-spec[, . . .]

Specifies one or more index file(s) that needs to be reviewed. The syntax of a file specification determines which files will be processed, as follows:

- If you do not enter a file specification, the REVIEW command processes all versions of the files in the current default directory.
- If you specify only a device name, the REVIEW command uses you default directory specification.
- Whenever the file specification does not include a file name, file type and a version number, all versions of all files in the specified directory are processed.
- If a file specification contains a file name or a file type, or both, and no version number, the REVIEW command processes all versions.
- If a file specification contains only a file name, the REVIEW command processes all files in the current default directory with that file type, regardless of file type and version number.
- If a file specification contains only a file type, the REVIEW command processes all files in the current default directory with that file type, regardless of file name and version number.

Wildcard characters can be used in the directory specification, file name, file type, or version number fields of a file specification to process all files that satisfy the components you specify. Separate multiple file specifications with either commons or plus signs.

Note: **If wildcard characters are given for the file name and/or file type specifications, all the index files that match the rest of the specification will be affected.**

DESCRIPTION

The REVIEW utility analyzes an exiting RMS index file and reports the expected results of fine tuning its internal structures and parameters to cause minimum file fragmentation and best access time. It uses utilities provided with VMS to perform some of its functions.

QUALIFIERS

/AFTER=time
/NOAFTER

Requests that the processing be held until after a specific time. If the specified time has already passed, the processing is started immediately.

Time can be specified as either an absolute time or as combination of absolute and delta times. See the *VMS DCL Concepts Manual* for complete information on specifying time values.

You may typically want to OPTIMIZE the files at midnight or during lunch hour when other applications may not need the index files.

/BEFORE[=time]

Selects only those files dated prior to the specified time. You can specify time as an absolute time, a combination of absolute and delta times, or as one of the following keywords: TODAY (default), TOMORROW, or YESTERDAY.

/EXCLUDE=file-spec[, . . .]

Excludes the specified files from the REVIEW operation. When using /EXCLUDE in an REVIEW operation on of a different device, use only the file name in the file specification. Wildcard characters are allowed in the file specification. However, you cannot use relative version numbers to exclude a specific version. If you provide only one file specification, you can omit the parentheses.

N

/LOG
/NOLOG (default)

Controls if the output of the command is displayed.

/OUTPUT[=file-spec]
/NOOUTPUT

Controls where the output of the command is sent. By default, the display is written to the current SYS$OUTPUT device. No wildcard characters are allowed.

If you enter /OUTPUT with a partial file specification (for example, /OUTPUT=[ADAMS]), ORGANIZE is the default file name and REPORT the default file type. If you enter/NOOUTPUT, output is suppressed.

The /OUTPUT qualifier is incompatible with the /PAGE qualifier.

/PAGE
/NOPAGE (default)

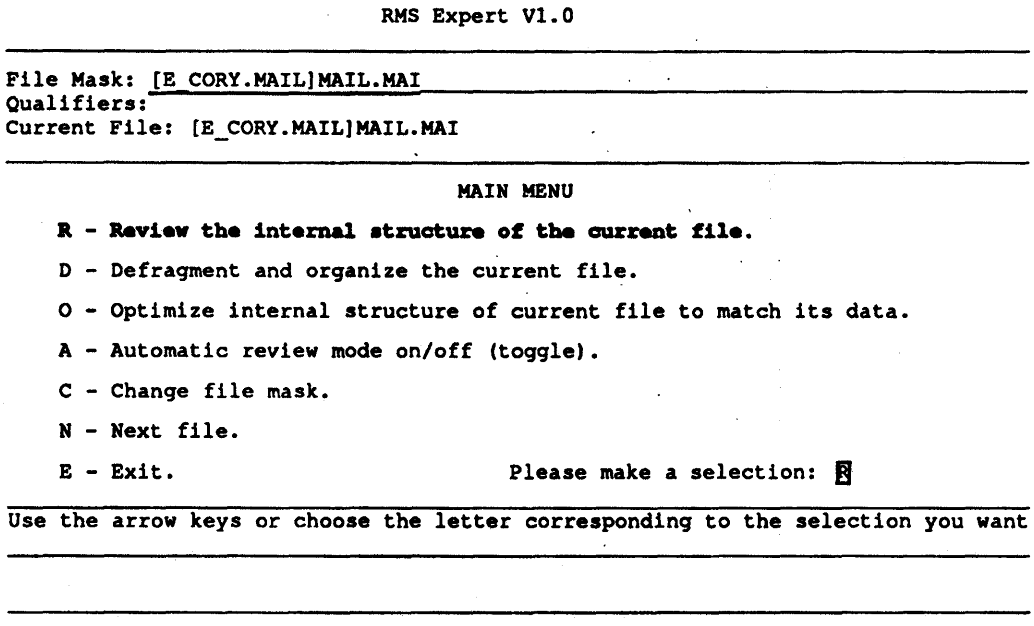
Controls whether ouput from the REVIEW command is displayed one screen at a time.

The /PAGE qualifier is incompatible with the /OUTPUT qualifier.

/SCREEN_MODE
/NOSCREEN_MODE (default)

REVIEW/INTERACTIVE is a menu driven user interface to the RMS Expert utilities. When started, REVIEW/INTERACTIVE provides the user with several options, see Figure RMX–1, REVIEW/INTERACTIVE Screen Mode Menu Interface.

Figure RMX–1 REVIEW/~~INTERACTIVE~~ SCREEN_MODE Screen Mode Menu Interface



```
                    RMS Expert V1.0

File Mask: [E_CORY.MAIL]MAIL.MAI
Qualifiers:
Current File: [E_CORY.MAIL]MAIL.MAI

                              MAIN MENU

    R - Review the internal structure of the current file.

    D - Defragment and organize the current file.

    O - Optimize internal structure of current file to match its data.

    A - Automatic review mode on/off (toggle).

    C - Change file mask.

    N - Next file.

    E - Exit.                      Please make a selection: R

Use the arrow keys or choose the letter corresponding to the selection you want
```

After the file(s) are entered in the the **File:** field, press the Enter, Do, or Return key. After the first file matching the selection criteria is found, the file name is displayed in the **Current File:** field. The operation desired my be selected by moving the cursor to the desired choice and pressing the Enter, Do, or Return key. The cursor may be moved by using the arrow keys, the Select key, or Tab key. Help may be obtained by pressing the Help or PF2 key.

After selecting the option, the selected utility is envolked, with its report displayed in the option section of the menu. When the operation is complete, the next file is selected and the user is presented with the options.

/SINCE[=time]

Selects only those files dated after the specified time. You can specify time as an absolute time, a combination of absolute and delta times, or as one of the following keywords: TODAY (default), TOMORROW, or YESTERDAY. Time must be specified using the VMS format of DD-MMM-YYYY HH:MM:SS.CC; international date/time formats will not be accepted by the system. See the *VMS DCL Concepts Manual* for complete information on specifying time values.

EXAMPLES

1

```
$ REVIEW DUA0:[DBA...]*.*
```

Here REVIEW searches and REVIEWs all the index files under the root directory DUA0:[DBA].

2

```
$ REVIEW DUA0:[*...]
```

In this example, REVIEW searches all the index files on the disk DUA0: For each index file found, it produces a report.

Part V RMS EXPERT Messages

This part contains a list of messages and actions to take when a message is received.

ASTFAIL, **AST routine did not complete successfully.**

Facility: RMX, ORGANIZE Utility

Severity: Warning

Explanation:

User Action:

CHANASS, **Error assigning channel to mailbox !AS.**

Facility: RMX, ORGANIZE Utility

Severity: Warning

Explanation:

User Action:

CLOSEFILE, **Error closing file !AS.**

Facility: RMX, ORGANIZE Utility

Severity: Error

Explanation:

User Action:

CONNFILE, **Error assigning channel to file !AS.**

Facility: RMX, ORGANIZE Utility

Severity: Error

Explanation:

User Action:

CREATFILE, **Error creating file !AS.**

Facility: RMX, ORGANIZE Utility

Severity: Warning

Explanation:

User Action:

CREGRLOG, **Error creating group logical name for mailbox !AS.**

Facility: RMX, ORGANIZE Utility

Severity: Warning

Explanation:

User Action:

CREPRC, **Error creating detached process.**

Facility: RMX, ORGANIZE Utility

Severity: Warning

Explanation:

User Action:

ENQCREXLCK, Error creating Exclusive lock on VBN !ZL.

Facility: RMX, ORGANIZE Utility

Severity: Warning

Explanation:

User Action:

ENQCRNLLCK, Error creating NULL lock on VBN !ZL.

Facility: RMX, ORGANIZE Utility

Severity: Warning

Explanation:

User Action:

ENQMODEXLCK, Error modifying lock to Exclusive VBN !ZL.

Facility: RMX, ORGANIZE Utility

Severity: Warning

Explanation:

User Action:

ENQMODNLLCK, Error modifying lock to NULL on VBN !ZL.

Facility: RMX, ORGANIZE Utility

Severity: Warning

Explanation:

User Action:

ENQMODPWLCK, Error modifying file lock to Protected write.

Facility: RMX, ORGANIZE Utility

Severity: Warning

Explanation:

User Action:

FAIL, Routine failure.

Facility: RMX, ORGANIZE Utility

Severity: Fatal

Explanation:

User Action:

FATAL, Fatal error occurred during processing.

Facility: RMX, ORGANIZE Utility

Severity: Fatal

Explanation:

User Action:

GETLOCK, Error accessing lock information on file !AS.

Facility: RMX, ORGANIZE Utility

Severity: Warning

Explanation:

User Action:

GETPID, Unable to obtain PID of current process.

Facility: RMX, ORGANIZE Utility

Severity: Warning

Explanation:

User Action:

GETUNIT, Unable to obtain unit number for mailbox !AS.

Facility: RMX, ORGANIZE Utility

Severity: Warning

Explanation:

User Action:

INVBEFORE, Invalid date/time specification for BEFORE. Option ignored.

Facility: RMX, ORGANIZE Utility

Severity: Warning

Explanation:

User Action:

INVCONCUR, **Invalid value for CONCURRENT: !ZW. Value must be positive. Option ignored.**

Facility: RMX, ORGANIZE Utility

Severity: Warning

Explanation:

User Action:

INVSINCE, Invalid date/time specification for SINCE. Option ignored.

Facility: RMX, ORGANIZE Utility

Severity: Warning

Explanation:

User Action:

MBXCR, Unable to create mailbox !AS.

Facility: RMX, ORGANIZE Utility

Severity: Warning

Explanation:

User Action:

NOCMEXEC, CMEXEC privilege required to run utility.

Facility: RMX, ORGANIZE Utility

Severity: Warning

Explanation:

User Action:

NOCOMMAND, Function will be implemented in a later release.

Facility: RMX, ORGANIZE Utility

Severity: Success

Explanation:

User Action:

NOLOCK, Insufficient memory allocated for lock database. Internal error.

Facility: RMX, ORGANIZE Utility

Severity: Warning

Explanation:

User Action:

NOMEM, **Unable to allocate virtual memory.**

Facility: RMX, ORGANIZE Utility

Severity: Warning

Explanation:

User Action:

NORMAL, **Normal Completion.**

Facility: RMX, ORGANIZE Utility

Severity: Success

Explanation:

User Action:

OPENFILE, **Error opening file !AS**

Facility: RMX, ORGANIZE Utility

Severity: Error

Explanation:

User Action:

PARSFILE, **Error parsing filename !AS. Invalid file specification.**

Facility: RMX, ORGANIZE Utility

Severity: Error

Explanation:

User Action:

READADSCR, **Unable to read area descriptor block at VBN !ZL.**

Facility: RMX, ORGANIZE Utility

Severity: Warning

Explanation:

User Action:

READBKT, **Unable to read bucket at VBN !ZL.**

Facility: RMX, ORGANIZE Utility

Severity: Warning

Explanation:

User Action:

READKDSCR, **Unable to read key descriptor for key !ZW.**

Facility: RMX, ORGANIZE Utility

Severity: Warning

Explanation:

User Action:

READVBLK, **Error reading virtual block on mailbox !AS.**

Facility: RMX, ORGANIZE Utility

Severity: Warning

Explanation:

User Action:

ROUTINE, **From routine !AS.**

Facility: RMX, ORGANIZE Utility

Severity: Informational

Explanation:

User Action:

SEARCHFILE, **Unable to find given file !AS**

Facility: RMX, ORGANIZE Utility

Severity: Error

Explanation:

User Action:

WRITBKT, **Unable to read bucket at VBN !ZL.**

Facility: RMX, ORGANIZE Utility

Severity: Warning

Explanation:

User Action:

WRITDSCR, **Unable to update key/area descriptor block at VBN !ZW.**

Facility: RMX, ORGANIZE Utility

Severity: Warning

Explanation:

User Action:

WRITVBLK, **Error writing virtual block on mailbox !AS.**

Facility: RMX, ORGANIZE Utility

Severity: Warning

Explanation:

User Action:

WRTATTN, **Error setting write attention for mailbox !AS.**

Facility: RMX, ORGANIZE Utility

Severity: Warning

Explanation:

User Action:

Index

```
!*+
!-----------------------------------------------------------------------
!
!                Name: RMS Expert Module Classification File
!
!               Ident: RMX-C001C01-01-00
!
!                Date: 15-JAN-1991
!              Author: Earl S. Cory
!
!-----------------------------------------------------------------------

!-----------------------------------------------------------------------
!
!                          History of Development
!
!  Version      Date      Programmer    Description of work done and impact
! --------- ---------- ------------- ----------------------------------
!
!  01-00    15-JAN-1991 Earl S. Cory    Initial Release
!
! [Change_entry]
!
!----------------------------------------------------------------------
!
!-----------------------------------------------------------------------
!
!      Functional Description
!
!          This file documents Classification Number Assignments for RMS Expert
!          modules.
!
!      End Functional Description
!
!-----------------------------------------------------------------------
```

RMS Expert Module Classification File

```
!--------------------------------------------------------------------------
!
!                        Module Identifier Format
!
!                           RMX-ccssuut-vv-rr
!
!                 The following letters may not be used:
!
!                              I O Q S V Z
!
!                             Module Classes (cc)
!
!             01  ORGANIZE Utility
!             20  OPTIMIZE Utility
!             30  REVIEW Utility
!             40  RMX_SHUTDOWN Utility
!             50
!             60
!             70
!             80
!             90  RMS Expert Test Routines
!             99  General RMS Expert Routines
!             A0  RMS Expert Help Files
!             B0  RMX Expert Kit Files
!             C0  RMS Expert Documentation Files
!             D0  RMS Expert Build Files
!             E0  RMS Expert Message Files
!
!
!                              Unit Numbers (uu)
!
!              01-1Y Source Files
!              20-3Y Include Files
!              41-5Y Assembly Command Procedures
!              60-6Y Link Command Procedures
!              70-7Y Build Command Procedures
!              80-8Y Command Language Definitions
!              90-9Y Message Files
!              A0-BY Help Modules
!              C0-DY Documentation Files
!              E0-FY Command Procedures
!
!------------------------------------------------------------------------
```

                RMS Expert Module Classification File

```
!------------------------------------------------------------------------
!
!                            Module Types (t)
!
!              Source Code Files:
!                  A - Main program module
!                  B - Sub program module
!                  C - Function definition
!                  D - Include files
!                  E -
!                  F -
!
!              Compiled Files
!                  G - Data Definition Tables
!                  H - Display Files
!                  J - Message files
!                  K - Program Design Language files
!                  L -
!                  M -
!
!              Definition Files:
!                  N - Command Language Definitions
!                  P - Schema File Definitions
!                  R - File Definition
!                  S -
!
!              Procedural Files:
!                  T - Command procedures
!                  U - Make and MMS descriptor file
!                  W - Other procedural files
!                  X -
!                  Y -
!
!              Informational Files:
!                  1 - Documentation file
!                  2 - Text/Memo/Information file
!                  3 - Help library module
!                  4 -
!                  5 -
!
!              Data Files:
!                  6 - Data Files
!                  7 -
!                  8 -
!                  9 -
!                  0 -
!
!--------------------------------------------------------------------------
!*-
```

RMS Expert Module Classification File

```
Class 01  ORGANIZE Utility

        01 ORGANIZE
           01
           02 ORGANIZE_COMMON.C
           50 ORGANIZE.H
           51 ORGANIZE_CLI.H
           52 ORGANIZE_COMMON.H
           53 ORGANIZE_INPLACE.H
           54 ORGANIZE_SINGLE.H
           55 ORG_MSG.H
           56 ORG_REPORT.H
           A1 ORGANIZE.CLD             Command Line Definition
           A2
           A3
           A4
           A5
           A6

        02 ORGANIZE Main Routines
           01 ORGANIZE_MAIN.C
           A1

        02 ORGANIZE Common Routines
           01
           A1
```

```
Class 20  OPTIMIZE Utility

        01
           01
           A1 OPTIMIZE.CLD              Command line definition

        02
           01
           02
           A1
           A2
```

RMS Expert Module Classification File

Class 30 REVIEW Utility

```
     01
        01
        02
        A1  REVIEW.CLD                Command line definition

     02
        01
        A1
```

```
          RMS Expert Module Classification File

Class 40  RMX_SHUTDOWN Utility

        01 Shutdown RMS Expert Utilities
           01 RMX_SHUTDOWN.C           C MAIN Source Module
           02 RMX_STOP.C               C Function Module
           50 RMX_SHUTDOWN.H           C MAIN header Module
           A1 RMX_SHUTDOWN.CLD         Command language Definition
```

```
              RMS Expert Module Classification File

Class 90  RMS Expert Test Utilities

        01 OGANIZE Test Modules
           01
           A1 ORGANIZE_COMMON.CLD              Command defininition language

        02 OPTIMIZE Test Modules
           01
           80
           A1

        03 REVIEW Test Modules
           01
           A1

        04 RMX_SHUTDOWN Test Modules
           01
           A1
```

RMS Expert Module Classification File

```
Class 99  General RMS Expert Routines

        01 HELP Routines
           01 RMX_HELP.C

        02
           01
           80
           A1

        03
           01
           A1

        04 RMS Expert Startup Moudles
           01 RMX_STARTUP.COM          Startup command procedure
           02 RMX_COMMANDS.COM         Assign commands command procedure
```

```
            RMS Expert Module Classification File

Class A0  RMS Expert Help Files

        01 RMS Expert Help Files
           01 RMX010.HLP               Help Library Main Module


Class B0  RMS Expert Kit Files

        01 Full Kit
           01 KITINSTAL.COM            Command Procedure

        02 Demo Kit
           01 KITINSTAL.COM            Command Procedure


Class C0  RMS Expert Documentation Files

        01 RMS Expert Module Identifier Modules
           C0 RMX_MODULES.IDENTS

        02 RMS Expert to VDD Cross References Documents
           C0 RMX_TO_VDD.RMX011     Cross Reference File

        10 RMS Expert User Manual
           01  RMX.SDML                Bookbuild profile
           02  RMX_SYMS.SDML           Symbol file
           03  RMX_FRONT.SDML          Front matter
           04  RMX_PART_1.SDML
           05  RMX_INTRODUCTION.SDML
           06  RMX_DESCRIPTION.SDML
           07  RMX_DESIGN.SDML
           08  RMX_THEORY.SDML
           09  RMX_PART_2.SDML
           10  RMX_INSTALLATION.SDML
           11  RMX_PART_3.SDML
           12  RMX_USING.SDML
           13  RMX_PART_4.SDML
           14  RMX_COMMANDS.SDML
           15  RMX_PART_5.SDML
           16  RMX_MESSAGES.SDML
           20  LOGO.PS                 IDSI logo
           21  LOGO_ICOM.SDML          IDSI logo icon file
           A0  RMX.COM                 Document build command procedure
           B0  RMS_EXPERT.DECW$BOOK    Manual in bookreader format
           B1  RMX.LINE                Manual in lineprinter format
           B2  RMX.PS                  Manual in postscript format
```

RMS Expert Module Classification File

Class D0 RMS Expert Build Files

01 Utilty Build Data Files

01 RMX_ASM.DAT	RMS Expert Assembly Data File
02 RMX_LNK.DAT	RMS Expert Link Data File

02 RMS Expert Build Command Procedures

E0 RMX_ASSIGN.COM	Command Procedure

Class E0 RMS Expert Message Files

01 RMS Expert Message Source Files

01 RMX_MSG.MSG	Message Source File
02 RMX_MSG.H	Message C Header File
03 RMX_ ~~APP~~ MESSAGE_APP.COM	