LUND
PERFORMANCE SOLUTIONS

WHAT YOU NEED TO KNOW. **WHEN** YOU NEED TO KNOW IT.

De-Frag/X Disk Manager

User Guide

Legal Notices

Lund Performance Solutions makes no warranty of any kind with regard to this manual, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Lund Performance Solutions shall not be held liable for errors contained herein or direct, indirect, special, incidental, or consequential damages in connection with the furnishing, performance, or use of this material.

Restricted Rights Legend

LUND PERFORMANCE SOLUTIONS

240 2nd Avenue SW

Albany, OR 97321

USA

Copyright Notices

Printed in the United States of America.

Trademark Notices

De-Frag/X Disk Manager version C.03

Whitney Olsen, Laura Bryngelson, Rodica Popa 03152002

TABLE OF CONTENTS

1

INTRODUCTION

Welcome to De-Frag/X, an exciting MPE/iX tool from Lund Performance Solutions. De-Frag/X is an extremely helpful and timely product that measures up to the standards you have come to expect from Lund Performance Solutions. De-Frag/X is available for MPE/iX versions 5.5 and beyond.

This brief User's Guide is intended to provide a clear and concise overview of the features and benefits of the De-Frag/X product. On-line help also exists for your convenience. To access the De-Frag/X on-line help sub-system, simply type **H** or a question mark (**?**) at the **De-Frag/X>** prompt. See “Reference Materials” on page 5 for more defragmentation information.

Chapter 1 lays out the advantages of using De-Frag/X, any special precautions you should take before running De-Frag/X, and where to look for product support. “Fragmentation Concepts” on page 7 addresses the types of fragmentation issues commonly found in the MPE/iX world. If you have any doubt that fragmentation issues affect system performance, simply analyze your system's performance, perform defragmentation and re-analyze your system's performance. You will then be able to objectively quantify the effect of disk fragmentation on your system's performance.

Overview

Lund Performance Solutions is in the business of providing software, consulting and training in the Hewlett-Packard system performance arena. As such, we have examined literally hundreds of MPE/iX systems. While it is true that there have been significant strides made in the I/O performance of MPE systems (MPE V systems have significant I/O bottleneck issues), it is not true that disk I/O bottlenecking is a thing of the past. A purist may say that all I/O issues on MPE/iX systems disappear with sufficient main memory resources. This may be true in the theoretical realm but in practical application this concept isn't entirely true. As MPE systems have grown, the application environment has grown enormously. It is not uncommon to see systems with millions and millions of records of application data. We have seen systems with as many as 100 disk devices. As such, the ratio between disk storage and main memory has actually increased significantly. Therefore, it is becoming increasingly important that MPE/iX system managers maintain a "tighter ship" from the perspective of I/O performance.

Advantages

Some of the advantages of using De-Frag/X center around the fact that, to date, Hewlett-Packard does not provide the ability to perform similar functionality for MPE/iX systems. MPE/iX systems, as of release 5.0 of the operating system, will have available a CONTIGVOL command in the VOLUTIL utility program. This command only addresses issues of contiguous free space, and has some limitations at that. It is by no means a comprehensive fragmentation utility product like De-Frag/X.

Although De-Frag/X can perform its functions while users are actively engaged in application work, you may find some files that users have opened for work may very well be those files which require the most maintenance from a defragmentation standpoint. If this is the case, there is no other option than to bring those users off the system, back the files up and then perform the defragmentation maintenance.

Special Precautions

While we at Lund Performance Solutions have developed the De-Frag/X product with the highest regard for the integrity of user data, it is very important to understand that we are performing tasks which are considered privileged in nature and are not available through Hewlett-Packard supported file system interfaces. Because of this, (and we cannot stress it enough,) **be sure to perform system backups before executing any of the data-intrusive De-Frag/X commands**. Additionally, we recommend that you run the Hewlett-Packard provided utility, FSCHECK.MPEXL.TELESUP, both before and after performing any of the data-intrusive commands. This program will analyze and report the status of the file system and any problems associated with file labels, etc.

NOTE You should run FSCHECK when NO other users or jobs are running.

To encourage this practice of running FSCHECK first, De-Frag/X programmatically restricts data-intrusive functions if FSCHECK was not run first. (You may override this restriction by using the RESET FSCHECK statement before executing any of the data intrusive-commands). Please see "FSCHECK - Questions and Answers" on page 25 for more information regarding FSCHECK.

Product Support

Lund Performance Solutions Main Offices

When you purchase support from Lund Performance Solutions, you benefit from the knowledge and experience of our technical support team. We are glad to help you interpret data and resolve performance issues. Our contracted product support entitles you to receive timely updates, bug fixes, documentation and direct technical support.

Postal Address

Lund Performance Solutions

240 2nd Avenue SW

Albany OR 97321 USA

Internet URL

Visit the Lund Performance Solutions website at **http://www.lund.com/**.

Telephone Number

For customer and technical support, call **(541) 812-7600**, Monday through Friday during the hours of 8:00 A.M., to 5:00 P.M., Pacific time, excluding holidays.

Fax Number

Transmit fax messages to **(541) 812-7611**.

E-mail Addresses

Send e-mail messages to:

- Sales Team **info@lund.com**
- Technical Support Team **support@lund.com**
- Documentation Team **documentation@lund.com**
- Certified Trainers **lti@lund.com**
- Consulting Team **lcs@lund.com**

Lund Performance Solutions Sales Team

Lund Performance Solutions' professional sales team is available to answer your sales and customer support questions Monday through Friday during the hours 8:00 A.M., to 5:00 P.M., Pacific time, excluding major holidays.

Please contact your sales representative for information about the latest Lund Performance Solutions products, the Lund Software Subscription Plan, upgrade options and prices, and more.

Lund Performance Solutions Technical Support Team

At Lund Performance Solutions, we are working hard to provide you with intuitive software products. Additionally, we try to provide superior online and printed documentation. However, should you find yourself with a technical question that you cannot answer with the tools provided, please contact our technical support team.

NOTE You must be a registered user to access Lund Performance Solutions' support services. Lund Performance Solutions' support services are subject to Lund Performance Solutions' prices, terms, and conditions in place at the time the services are used.

E-mail Tech Support

Ask questions and receive detailed answers from the technical support team by sending an e-mail message to **support@lund.com**. Please include the product serial number with your question. You will receive a reply by e-mail.

Telephone Tech Support

You can reach the technical support team by phone at **(541) 812-7600**, Monday through Friday during the hours 8:00 A.M., to 5:00 P.M., Pacific time, excluding major holidays. Emergency technical support is also available after hours, seven days a week.

When you call, please be at your computer, have the product documentation in hand, and be prepared to provide the following information:

- Product name and version number.
- Type of computer hardware you are using.
- Software version number of your operating system(s).
- Exact wording of any messages that appear on your screen.
- What you were doing when the problem occurred.
- How you tried to solve the problem.

Lund Performance Solutions Documentation Team

Lund Performance Solutions makes every effort to produce the highest quality documentation for our products, and we welcome your feedback. If you have comments or suggestions about our online Help or printed guides, send an e-mail message to **documentation@lund.com** or contact your account manager.

Lund Training Institute Certified Trainers

Lund Training Institute presents system performance training courses at their corporate training center in Oregon and at various locations across the United States and Canada throughout the year. The Certified Trainer Program is designed for trainers from all educational areas, including academia, consulting, and business.

For information about Lund Training Institute or to receive an application, please review our website, send an e-mail message to **lti@lund.com**, or contact your account manager.

Lund Consulting Services IT Consultants

Lund Consulting Services, a division of Lund Performance Solutions, offers strategic IT solutions and expert support to a wide range of businesses. Our team of experienced IT professionals provides onsite consulting, training, and project management services to help businesses optimize their computer resources and achieve long-lasting success.

For information about Lund Consulting Services, please review our website, send an e-mail message to **lcs@lund.com**, or contact your account manager.

Product Documentation

User's Guide

This document accompanies the De-Frag/X Disk Manager software as a guide for the new user and as a quick reference for experienced users. This guide assumes that you have a working knowledge of the MPE operating environment.

Online Help System

In the online Help system, you will find explanations of the many features of De-Frag/X Disk Manager as well as tips to guide you through the program's basic functionality.

Reference Materials

- *Taming the HP3000 - Volume 2*. Copyright © 1992 by Robert Lund, Albany, Oregon, USA
- *SOS/3000 Performance Advisor (for MPE/iX systems) User Manual*. Copyright ©1999-2000 by Lund Performance Solutions, Albany, Oregon, USA
- For additional information regarding fragmentation problems and corrections, go to the Allegro web site: **www.allegro.com/papers/fragmentation.html**.

2

FRAGMENTATION CONCEPTS

There are four types of fragmentation found on HP e3000 MPE/iX systems:

- "File Fragmentation."
- "Disk Fragmentation."
- "System Fragmentation."
- "Database Internal Fragmentation."

File Fragmentation

Within the context of disk fragmentation is the concept of a file extent. File fragmentation occurs when a file's set of extents becomes physically discontinuous on disk. The impact of file fragmentation is determined by the severity of fragmentation, the adequacy of main memory, the speed of disk devices, and the efficiency of MPE/iX's built-in pre-fetch mechanism.

Contiguous Extents

A file on an MPE/iX system can be broken up in to pieces known as "extents." These extents do not need to be next to each other on a disk, nor do they need to be on the same disk drive. This allows the file system to have very large files, (up to four gigabytes,) but not require that all the space be physically contiguous on the disk drive. A file which is contiguous means that all its extents are "next door neighbors" on a particular disk device. In addition, this allows us to build files which exceed the capacity of the largest disk drive available. A performance cost occurs when a file’s extents get spread out over the disk devices. This can present significant performance issues.

For example, let's assume you build a file using the following file build:

BUILD FOO1;REC=-80,,F,ASCII;DISC=100000,1,1

This will build the file "FOO1" with space for 100000 records in one physical extent and allocate all that disk space immediately. The advantage of this is that you are guaranteed to have all of the records that will ultimately go into this file physically contiguous on disk. The disadvantages are that you must have 100000 * 80 bytes of available, contiguous disk on a single disk device, and you must lose the availability of all of that disk space up front, perhaps long before it is needed.

Non-contiguous Extents

Now consider the following file build:

BUILD FOO2;REC=-80,,F,ASCII;DISC=100000,32,1

This will build the same file with a few differences. By specifying the "32" in the DISC parameter, we are telling the operating system to break this file up into multiple extents. (The exact number is not controlled on MPE/iX.) The ",1" in the DISC parameter tells the operating system that we only want a portion of the file allocated up front. In general, MPE will allocate about 1/32 of the file at that time. The advantage of this is that the file system doesn't have to obtain 100000*80 bytes of available, contiguous disk on a single disk drive. Rather, it has only to obtain (100000*80)/32 of available, contiguous bytes of disk initially. Additionally, you only lose the availability of (100000*80)/32 bytes of disk space at build time. You will acquire the rest of the disk space only as required as your application adds records to the file.

As you can see, these are significant advantages. MPE/iX system users must have and cannot live without these advantages, especially in these days of true mainframe-equivalent HP e3000 environments.

However, the disadvantage to building FOO2 in this manner can become apparent when subsequent extents are allocated. If the extents are allocated contiguously then there will be no performance degradation. There is some slight overhead associated with multiple extents resulting from more label table activity, but it is negligible in terms of any performance impact. If, however, the extents are allocated non-contiguously, there can be significant performance impact.

Let's examine these ramifications. Taking FOO2 as our example, we'll look at the worst case extent allocation scenario. If the worst possible allocation occurs, the FOO2 file will be located in 32 separate, non-contiguous areas of disk on one disk device. This means that a serial read of this file could result in 32 times the amount of physical disk accesses required than if all extents were contiguous or if the file consisted of a single extent.

If, however, these extents were located on 32 different disk devices, then the impact of the non-contiguous nature of the file distribution would be somewhat decreased by the fact that much of the disk access could be performed simultaneously over the 32 spindles.

Be aware that the FOO1 and FOO2 examples represent a simplified view of the internal operation of MPE/iX systems. There are other issues, particularly multi-page prefetching, that play a role in the overall efficiency of disk I/O on MPE/iX systems.

Disk Fragmentation

Disk fragmentation is best defined as the process by which logically-related data become physically disassociated on disk. Disk fragmentation can be considered a measurable current state of your data, as well as the dynamic process by which such fragmentation gradually occurs.

Disk fragmentation is when the free space on a disk drive is spread throughout the disk in many small pieces. We list this as a separate category because the more fragmented a particular disk device is, the more impact there is on overall system performance. Additionally, the more fragmented a particular disk device is, the harder it is for certain important system functions to be

performed on that device. For example, let's say that LDEV 1 (the system disk) has 1,000,000 sectors of free space, but is so fragmented that the largest free chunk is 13,000 sectors. In the event of an operating system update, MPE/iX requires a certain amount of contiguous free space on LDEV 1. If the required amount of contiguous disk space is not available, you will be unable to perform the update without intervention. That intervention may now take the form of a simple execution of the TRIM and CONDENSE commands. If you still do not have enough free space for the update, you can use the MAKEROOM command to make as much room as necessary.

NOTE The amount needed may change from release to release of the operating system.

Also, it is commonly thought that a lack of free space throughout a system can cause performance problems. There is a role for fragmentation in this same scenario. In other words, there is a fundamental level of free, contiguous free space required on your system, below which you must not go.

System Fragmentation

System fragmentation is yet another extension of file fragmentation. This is the perspective by which you must consider your whole system. This would include a file-level perspective, a disk-level perspective, a volume set-level perspective and a complete system-level perspective.

Database Internal Fragmentation

Database fragmentation is fragmentation within the skeletal structure of a DBMS such as Turbo Image. While specific internal DBMS fragmentation is outside the scope of De-Frag/X, all DBMS's exist on top of the MPE/iX file system. This means that there is still significant impact of defragmenting Turbo Image data sets at the file-level. In other words, given the database TRXDB1, you should still defragment the individual files of TRXDB101, TRXDB102, TRXDB103, etc.

It should still be said that the internal fragmentation of data within a DBMS is still a critical performance issue. Tools such as DBLOADNG (from the INTEREX Contributed Library) or HOWMESSY (from Robelle) were created to measure the internal fragmentation of Turbo Image databases. Included with any Lund Performance Solutions product is a copy of the DBLOADNG program courtesy of INTEREX. Additionally, a product such as Adager, from Adager Corporation, has the ability, via the DETPACK command, to actually fix DBMS-level fragmentation on Turbo Image databases. Information about this product can be obtained from a Lund Performance Solutions representative (see “Product Support” on page 3) or directly from Adager at 1-800-533-7346.

For more information regarding Turbo Image database performance issues, refer to Taming the HP3000 - Volume 2 by Robert Lund, available from Lund Performance Solutions.

3

COMMANDS LIST

The following is an alphabetized listing of the commands available within De-Frag/X. More detailed information about these commands is available in the on-line help facility and in the De-Frag/X User Reference Manual available from Lund Performance Solutions.

NOTE Commands flagged with an asterisk (*) are data intrusive (DI). A backup of your system is recommended before using the data intrusive commands.

DI	Command Name & Syntax	Description
	ANAlyze [ldev] [fileset] [options...] options :: = [NO]SIZE I [NO]SORT I MIN # I LDEV # I [NO]SUMMARY I PAGES < # I PAGES > #I NEEDTRIM I [NO]TRIM Default: @.@.@ SIZE, NOSORT, MIN 1, SUMMARY	Reports file-level fragmentation statistics such as fragmentation %, pages, extents, etc.
*	**BALance** <ldev I [ONVS] vsetname> [options] options ::= EXCEPT ldevlist I [no]CONFIRM I [no]FAST I [no]LOGcopy I [no]Quiet I [no]VERbose I [no]WALKIts I [no]Yes I NO I	Balances set of ldevs in specified volume set
	CHECKFILE <file I fileset> [FIXGID I YES I NO]	Examines the specified files for non-0 GroupID
	CLEARREDO	Clears redo command stack
*	**CLONEdisk** source_ldev# TO destination_ldev# [ALLOWbigger]	Copies every bit of data from one disk to another
*	**CONDense** <ALL I ldev#> [<CONFIRM I NOconfirm I YES>]	Disk level defragmenter

DI	Command Name & Syntax	Description
	COUNTFiles <ldev | ALL | EACH | volumeset>	Counts the files on one or all disks.
*	**DEFRAGment** <filename | fileset> [NO/TRIM] [YES | NO]	File level defragmenter
	DISC [ALL] [Summary | DETAIL] [ONVS vsetname] [BYVS] **DISCfree** ldev [Summary | DETAIL] [ONVS] ldev [Summary | DETAIL] [ONVS] or **DISK** (same as DISC) **DISKFREE** (same as DISCFREE)	Displays miscellaneous info for one or all drives
	DisplayExtents [FILE] filename [matchldev] [options] or **DE** [FLAB] file_label_address [options] [LDEV] ldev# [entry# | ALL] [options]	Displays a list of extents for specified file
	DO [prefix]	Executes the last command without editing
	DSTAT [ALLdisks] [ONVS vsetname] [options] **DSTAT** ldev [ONVS] [options]	Displays an expanded form of MPE's dstat command
	DUMPFI [file#1 [< /file#2 | #files >]] [options]	Displays information about files found during the most recent disk analysis
	DUMPPF [page#1] [<page#2 | #pages>] [UNMOVED] [INUSE] [TRANsient] [FREE] WALKLT] [TEMP] [FILENAME] [FILENUM #]	Displays information about pages on the most recent disk that was analyzed
	ECHO text	Echos the rest of the line to $STDLIST
	ERRor	Translates an error# into a message, using the system error message catalog
	EXIT	Exit or: //
	FILEs <# | MOVED | UNMOVED | UNUSABLE>	Displays file info based on De-Frag/X file number
	FINDPAGE <ANYldev | ldev#> page# [/page#] [options] **FINDSECTOR** <ANYldev | ldev#> sector# [/sector#] [options]	Find a file with an extent that includes the specified page or sector
	FINDSID <ALL | sid# | sid#/sid# | va | va/va> [options]	What objects are associated with the specified virtual addresses

DI	Command Name & Syntax	Description
*	**FRAGMENT** <filename | fileset> [TO=ldevlist]	Spreads file(s) over multiple drives
	HELP < prefix >	Displays help for every command with given prefix
	LISTLT	Lists label table for specified disk
	LISTREDO	Displays list of approximately last 100 executed commands
*	**MAKEROOM** ldev [#MB] [TO= ldevlist] [options...] options are: [NO]CONDENSE[NO]CONFIRMYES	Creates specified mbytes free space on given ldev
	MAP [<ALL | ldev# >]	Displays graphic summary of disk fragmentation
	MEMMAP [< ALL | ldev# | EACH | SUMMARY >].	Shows what pages of the disk are currently in memory
	PAGE #	Reports the name of the file that contains the specified disk page (if any)
	PSCREEN [PARTIAL]	Does a PSCREEN of your terminal
	REDO <prefix>	Executes the last command after prompting for edits
	RESET[Q] [option...]	Disables the option previously set by SET command
	SECTOR #	Reports the name of the file that contains the specified disk sector (if any).
	SET[Q] [option...]	Enables the specified De-Frag/X option
	SSM ldev# page#	Displays the Secondary Storage bitmap for the specified ldev and page
	SUSPEND	Typically used within QEDIT or MPEX to suspend itself and awaken its parent process

DI	Command Name & Syntax	Description
*	**TRIM** <filename | fileset > [NO]TRIM	Releases unused disk space after the EOF marker
	UNMOVED	Lists all files that were not moved by the most recent CONDENSE or MAKEROOM
	USE <filename>	Tells De-Frag/X to read input from specified file
	USEQ <filename>	A "quiet" version of USE which does not echo input
	VERSION	Displays the current version of De-Frag/X
	ZOOM [ROW=]# [COL=]# [[CHUNKS] #] < LAST | NEXT>	Displays detail info about the pages in a particular "chunk" from a De-Frag/X "disk map"

4

GETTING STARTED

Before running De-Frag/X, we recommend running the Hewlett-Packard provided utility, FSCHECK.MPEXL.TELESUP, both before and after performing any of the data-intrusive commands. This program will analyze and report the status of the file system and any problems associated with file labels, etc.

To run this program, do the following:

1 RUN FSCHECK.MPEXL.TELESUP
2 CHECKDIRC ALL IGNORE
3 CHECKLABEL ALL IGNORE
4 CHECKEXTENTS ALL
5 EXIT

Be sure that you are not running De-Frag/X when you execute this program.

NOTE You may get a misleading error message if FSCHECK runs at the same time as De-Frag/X. FSCHECK will also incorrectly report errors on any open files.

Running De-Frag/X

To run the De-Frag/X program, enter the following:

:DEFRAGX.DEFRAGX.LPS

NOTE When using any of the data-intrusive De-Frag/X commands described later in this manual (i.e., CONDENSE, MAKEROOM, DEFRAGMENT, FRAGMENT, CLONEDISK, and TRIM), you will need to be logged on with system manager (SM) capability.

Analysis

The first step in using De-Frag/X is to analyze the current nature and level of fragmentation on your system. To do this, use the MAP, ANALYZE, ZOOM and DISPLAYEXTENTS commands.

The MAP command gives a graphic representation of the nature of disk-level fragmentation on your system. Figure 4.1 shows the output of a MAP ALL command on an 18-drive system.

```
De-Frag/X> map all
                                                                    Disk Size of
                                                          Available  Size  chunks
Ldev [ (disk usage map...)                                ] PermMBs   MBs     MBs
   1 [PPPPpppPppPpppppppppppppppppppp***pppp*pp*pp*p***p*]    347   1,292    25.9
   2 [PPPPpppppppppppppppPPPpPpppppppppppppppppppPPpppPPPPp]     49   1,292    25.9
  11 [PPPPpppppppppppppppPppPpPPppPPpppppppPpppppppPppppppp*pp]    119   1,279    25.6
  12 [PpPPpppPpPppPpPpPPpppPPPpPppppPpppppppPpppppppp*pPpp]     93   1,279    25.6
  13 [pPPPppPPPPppPPPPPPpPPPPPPPPpPPppppPpppppppppppppppPp*]    102   1,279    25.6
  14 [pPPPPpppppppppPPPppppPpppPPPPPpPppPpPpppppppppPp**p]    119   1,279    25.6
  15 [pppPpPppPpppppppPpppPPPpPPPpppppppPPpppppppppp*pppp]    112   1,279    25.6
  16 [PPPPppPPpppppppppPpppPpppppppppppppPpppppppppPp**p*]    111   1,279    25.6
  17 [pPPPpPPpppppppppppppppppppppppppppppPpppppppppp*pPPp*]    103   1,279    25.6
  18 [ppPPpppppppppppppppppppppppppppppppppppppppppppppp**]    117   1,279    25.6
MAP_characters:
   Free       *   Permanent  P   Transient  T   unmovable  X
   part Perm  p   part Trans t   same_ldev  x
```

Figure 4.1 *MAP ALL Command Output*

As you can see, the MAP ALL command creates a visual picture of the fragmentation of all the disk drives on your system. This helps to give you an idea of the level of disk fragmentation on a global, system-wide basis. Each character in the display represents a "chunk" of disk space. The chunk size represents 2% of the disk drive and therefore varies based on the size of the disk drive for which it is reported. For example, on a 571 megabyte disk drive, a chunk is equal to approximately 11.4 megabytes. On a 670 megabyte disk drive, a chunk is approximately 13.4 megabytes.

NOTE The word "chunk" is used here for purposes specific to the De-Frag/X product and has no technical meaning or merit outside of a discussion about De-Frag/X. "Chunk" is not a standard HP e3000 term and thus will not be familiar to people outside of this context

The character used to label each chunk identifies the predominant nature of that disk space. The following list shows all the MAP label characters and their significance.

X Displayed if there is any unmovable file present in the chunk. This preempts all other designations.

x Displayed if there are any ldev-specific (i.e., specified for a given drive) files. This pre- empts all other designations except "X."

P Displayed if the entire chunk contains permanent files.

T Displayed if the entire chunk contains temporary (transient) files.

The following three designations are displayed only when none of the above apply:

p Displayed if the chunk contains mostly permanent files.

t Displayed if the chunk contains mostly temporary files.

***** Displayed if predominantly free space.

NOTE Any MAP commands run before CONDENSE or ANALYZE will not display the "x" or "X" types.

Examining Figure 4.1 on page 16, the system displayed appears to be only mildly fragmented. However, a closer examination may be more revealing. To look deeper, a MAP <ldev> command can be used. Figure 4.2 below shows a MAP 15 command issued on ldev 15.

```
De-Frag/X> map 15
------------------------------------------------------------------------
Ldev   15:   (Each chunk represents 394 pages, or 1.5 MB)
   [PPPPPPPpPPPPPPPPPPpppPPPPPpPPPPPPPpPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP]  0
   [PPPPPpPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP]  1
   [PPPPPPPPPPPPPPPPPPPPPPPPPpppppppppPPppPPPpPPpPPPPPPPpPpppPpPPpPPPPP]  2
   [PPPPPPPPPPPPpPPPPPpPPPPPPPPPPpPPPPPPPPPPPpPPPPPPPPPPPpPPPPPPPPPPPPP]  3
   [PPPPPPPPPPPPPPPPpPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP]  4
   [PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP]  5
   [PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP]  6
   [PPPPPPPPPPPpPPPPPPPPPPPpPPPPPPPPPPPpPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP]  7
   [PPPPPPPPPPpPPPPpPPPPpPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP]  8
   [PPPPPPPpPPPPPPPPPPPPPPPPPPPPPPPPPpPpPPPPpPPPPPPPPPPpPpPPpppPP]  9
   [PPPPPPPPPPPPPPPPPPPPpPppppppPPPPPPPPpPpPPPPpppPPppPppppppPPPpPppPPP] 10
   [PPPpppPpp*********pPPPPPPPpPPPPPPPPPPpPPPPPPPPPPPPPp************] 11
   [************************************************************] 12
Col[0....+....10...+....20...+....30...+....40...+....50...+....60..] Row
Available Permanent Disk: 142 MB (582,096 sectors); (disk size: 1,279 MB)
MAP_characters:
   Free       *   Permanent  P   Transient  T   unmovable  X
   part Perm  p   part Trans t   same_ldev  x
```

Figure 4.2 *MAP <ldev> Command Output*

NOTE While the MAP ALL command is limited to representing chunks of disk space no smaller than 11.4 megabytes (i.e., 2% of the disk drive), the MAP 15 command showed 1.5 megabytes for each chunk (0.12%).

As you can see with Figure 4.2, the disk drive is more fragmented than it would appear solely based on the MAP ALL display for that disk drive. You can go one level deeper via the ZOOM command and see what each chunk contains.

Up to this point we have performed only disk-level analysis. For the sake of discussion, let's assume that we have determined that the fragmentation level on ldev 15 is at an unacceptable level. We know only that the disk drive itself is not at its greatest efficiency. We do not as yet know whether the files on that device are fragmented.

There are two commands used to measure the level of individual file fragmentation: ANALYZE and DISPLAYEXTENTS. The ANALYZE command provides a listing of files, on a system-wide basis or on a drive-by-drive basis, which are fragmented beyond a desired user-defined percentage.

```
Filename                    Frag%  #Pages       #MBs  #Extents   TrimmablP
--------------------------- -----  ---------  ------  ---------  ---------
X00C0038.LOGFL.CAROLIAN       15%        510       1        107
I0007401.LOGSG.CAROLIAN       15%        240                 59
I0008101.LOGSG.CAROLIAN       12%        185                 47
I0006601.LOGSG.CAROLIAN       12%        178                 46
I0008300.LOGSG.CAROLIAN       24%        364       1         80
SOSWKDEF.PUB.LPS              25%          5                  2
TEST29.PGMS.SOSDEV             2%         37                  3
SL001058.LOGFILES.LPS          2%         85                  4
R0009100.LOGSV.CAROLIAN        7%        231                 20
I0008900.LOGSV.CAROLIAN       14%        367       1         89
FRAG12.DATA.DEFTEST           80%        100                100
R0008600.LOGSV.CAROLIAN        8%        547       2         48
I0008601.LOGSG.CAROLIAN       22%        207                 51

Fragmentation by fileset:   Frag%  #Pages       #MBs  #Extents   #Files
             files shown:     11%   192,884     753     26,304       945
             files examined:   0% 2,870,455  11,212     39,518   12,247

Sectors savable by a TRIM:
   files shown:     Space savable by a TRIM: 4,378 pages (17.1 MBs)
   files examined:  180,209 pages (703.9 MBs)
```

Figure 4.3 *ANALYZE Command Output (partial)*

The DISPLAYEXTENTS command (abbreviated as "DE") allows you to see the fragmentation and extent allocation of individual files. Figure 4.4 contains a sample partial DISPLAYEXTENTS output.

```
De-Frag/X> displayextents diskinfo.sample.lps

File: /LPS/SAMPLE/DISKINFO  (2 extents)

   Ldev   Disk Page#   # Pages  File Page #
   ----   ----------   -------  -----------
      18      41,923         4            0
      17      74,981         6            4

   # extents in file:           2
   # pages in file:    10 pages
% fragmented:    8.8
```

Figure 4.4 *DISPLAYEXTENTS Command Output*

Disk Condensing

Let's now assume that you have completed the analysis phase and have determined to perform some maintenance against ldev 15, as well as to some files resident on that device. Now you need to make a decision: do you address the disk-level fragmentation or the file-level fragmentation first?

In a perfect world, it would be desirable to perform file-level defragmentation before performing disk-level defragmentation. This would insure that individual files are as contiguous as possible. Performing disk-level defragmentation first will consolidate the scattered chunks of free space on the drive, but won't necessarily consolidate scattered chunks of the same file together.

In some situations, however, there could be a need to perform disk-level defragmentation first. For example, you may have to perform a disk condense in order to have enough room for efficient defragmentation of a particular file. The variables surrounding the answer to this question are:

- The amount of free space available.
- The size of the disk file you wish to defragment.
- The nature of the disk file you wish to defragment.

For example, one of the rules regarding individual file defragmenting is that the file cannot be more than half the size of the disk drive on which it resides. This is because it is a current requirement of De-Frag/X that any file that is to be defragmented must remain on the same disk drive. For ease of discussion, and to more quickly introduce you to the global capabilities of De-Frag/X, we will begin by addressing file-level defragmentation.

File-level Defragmentation

Before you attempt to perform the CONDENSE command against all selected disk drives, you need to consider the issue of file-level fragmentation. Once again, the ANALYZE command is used to evaluate file-level fragmentation. Once you have identified the files which are fragmented beyond acceptability, you can perform DEFRAGMENT commands against each selected file.

Remember, this is an intrusive operation and should be preceded by a backup of the affected files. Figure 4.5 shows sample output of a DEFRAGMENT command performed against the file illustrated in the DISPLAYEXTENTS command in Figure 4.4.

```
De-Frag/X> defragment diskinfo.sample.lps

File: /LPS/SAMPLE/DISKINFO
   Fragmentation:    8.8%
   # Extents:        2
   # Pages:         10 (160 sectors)

   Will move to 18/ 59,838:   10
   done
```

Figure 4.5 *DEFRAGMENT Command Output*

Figure 4.6 shows the output of a DISPLAYEXTENTS command performed against the DEFRAGMENT'ed file.

```
De-Frag/X> displayextents diskinfo.sample.lps

File: /LPS/SAMPLE/DISKINFO  (1 extent)

   Ldev   Disk Page#   # Pages  File Page #
   ----   ----------   -------  -----------
      18      59,838        10            0
% fragmented:    0
```

Figure 4.6 *DISPLAYEXTENTS Command Output*

The DEFRAGMENT'ed file is now a single extent. This should guarantee optimal I/O performance against that file, provided that the file is accessed serially.

De-Frag/X includes two other commands, TRIM and FRAGMENT, that function on the file level to perform data intrusive functions. The TRIM command causes unused disk space after the EOF marker on a desired file or fileset to be released. When performed system wide this can recover a large amount of free space. We recommend using the TRIM command in conjunction with the CONDENSE command to free up disk space on systems where free space is a concern. By doing a TRIM @.@.@ NOTRIM you will find out how much space will be returned by the system without actually invoking the data intrusive TRIM command.

The FRAGMENT command is the reciprocal of the DEFRAGMENT command. It "intelligently" fragments (i.e., spreads out) a desired file over multiple drives in order to more equally distribute disk I/O for that file.

Disk-level Defragmentation

The command used to perform disk-level defragmentation is the CONDENSE command. You can perform the CONDENSE command on a disk-by-disk basis (i.e.,CONDENSE <ldev>) or on a system-wide basis (i.e., CONDENSE ALL). Keep in mind that except for the DEFRAGMENT, up to this point you have performed only non-intrusive activities. While De-Frag/X can perform its work without a backup and with users on the system, you should always have a backup before you perform data intrusive activities such as the CONDENSE command (or any similar type of commands from products by Hewlett-Packard or third party vendors).

Figure 4.7 and Figure 4.8 show sample output for the CONDENSE command used against ldev15 in the above examples. The output of the CONDENSE command is similar to the output of the MAP command. A notable difference between the output of the two commands is the presence of the "x" and "X" characters. These characters differentiate between (1) regular permanent disk files, (2) files which are movable but not off the current ldev, and (3) files not movable at all (primarily operating system files from the System Load Tape (SLT)). The CONDENSE output and any MAP output subsequent to a CONDENSE command will show the movable and unmovable files.

NOTE Any MAP commands run before CONDENSE or ANALYZE will not display the "x" or "X" types.

```
De-Frag/X> condense 15
Wait...analyzing files...
Will condense ldev 15 (first free page @ 3,129)

Largest contiguous free space currently 29,636 pages (474,176 sectors).

Want to continue with Condense?  (def=N) y
```

Figure 4.7 *CONDENSE command prompt*

```
 Unable to move 1 files (1 extents):
    # extents not moved, other:                    1

 # Files moved: 105 (179 extents, 10,600 pages (41 MB))

 Biggest chunk of free space:
                      Sectors     Pages   MBs
    before          :  474,176    29,636   115
    after           :  565,280    35,330   138
```

Figure 4.8 *CONDENSE Command Output*

Figure 4.8 shows sample output of a CONDENSE 15 command.

```
De-Frag/X> map 15
--------------------------------------------------------------------------------
Ldev   15:    (Each chunk represents 394 pages, or 1.5 MB)
   [XXXPPPPPPPPPPPPPPPPPxPPPPPPPPPPPPPxPPPPPPPPPPPPPPPPPPPPPPPPPPP]  0
   [PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPxPppPpppPPPPpppPppPPpPPP]  1
   [PPPPPPPPPPPPPPPPPPPPPPPPppppPppppPPppPPPpPPpPPPPPPpPppppPpPPpPPPPP]  2
   [PPPPPPPPPPPpPPPPPpPPPPPPPPPpPPPPPPPPPPPPPPPPPPPpPPPPPPPPPPPPPPPP]  3
   [PPPPPPPPPPPPPPPPpPPPPPPPPPPPPPPPPpPPPPPPPPPPpPPPPPPPPPPPPPPPPPPpP]  4
   [PPPpPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP]  5
   [PPPPPPPPPPPPPPPPPPPPPPPPPPPPpPPPPPpPPPPpPPPPPPPPPPPPPPPPPPPPPpPPP]  6
   [PpPPPPPPPPPpPPPPPPPPPPPpPPPPPPPPPPpPPPPPPPPPPPPPPPPPPPPPPPPPPPP]  7
   [PPPPPPPPPPpPPPPpPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP]  8
   [PPPPPPPpPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP]  9
   [PPPPPPPPPPPPPPPPPPPPPPPppppPPPPPPPPpPpPPPPPpPPPPPPPPPPPPPPPPPPP] 10
   [PPPPpppPPPPPPPPPPPPpPpppPPppPPPPPPPPPPp************************] 11
   [************************************************************] 12
Col[0....+....10...+....20...+....30...+....40...+....50...+....60..] Row
Available Permanent Disk: 142 MB (582,096 sectors); (disk size: 1,279 MB)
MAP_characters:
   Free       *   Permanent  P   Transient  T   unmovable  X
   part Perm  p   part Trans t   same_ldev  x
```

Figure 4.9 *ldev 15 after CONDENSE command*

Figure 4.9 shows the sample output of a MAP 15 command after the CONDENSE command has been run. Notice the differences between Figure 4.9 and Figure 4.2. As you can tell by examining lines 11 and 12, fragmentation is lessened by performing the CONDENSE command.

Free Space

Every MPE/iX system manager has experienced the problem of not having enough free space on a specific ldev. This particularly concerns free space on ldev 1 prior to an operating system update. Up to this point, the process by which that additional free space was acquired was to

perform tedious, time-consuming moves of files. De-Frag/X has a command (MAKEROOM), which is specifically intended to simplify that process. For example, Figure 4.10 shows the DISCFREE A output of a sample ldev 16.

```
LDEV :     16 -- (U2:MEMBER5)

LARGEST FREE AREA:        262144  TOTAL FREE SPACE:     652672

    0 BLOCK(S) OF       1-     9 CONTIG. SECTORS =          0 FREE SECTORS.  0%
  336 BLOCK(S) OF      10-    99 CONTIG. SECTORS =       7776 FREE SECTORS.  1%
   22 BLOCK(S) OF     100-   999 CONTIG. SECTORS =       7648 FREE SECTORS.  1%
   12 BLOCK(S) OF    1000-  9999 CONTIG. SECTORS =      28240 FREE SECTORS.  4%
    3 BLOCK(S) OF   10000- 99999 CONTIG. SECTORS =     110224 FREE SECTORS. 17%
    3 BLOCK(S) OF  100000-AND UP CONTIG. SECTORS =     498784 FREE SECTORS. 76%
```

Figure 4.10 *DISCFREE A Command Output*

This drive has available 998384 sectors of free space. Let's say an additional 300 megabytes of free space is needed. Figure 4.11 shows output to the MAKEROOM command required to make this available:

```
De-Frag/X> makeroom 16 20
Will use the following ldevs as destinations:
     13    165 MBs avail ( 12% avail)
     18    136 MBs avail ( 10% avail)
     14    117 MBs avail (  9% avail)
     15     80 MBs avail (  6% avail)
     12     77 MBs avail (  6% avail)
     17     71 MBs avail (  5% avail)
     11     69 MBs avail (  5% avail)
Wait...analyzing files...
Will make room on ldev 16

Want to continue with MakeRoom?  (def=N) y
```

Figure 4.11 *MAKEROOM Command Output*

FSCHECK - QUESTIONS AND ANSWERS

Q. What is FSCHECK?

A. FSCHECK is a utility provided by Hewlett-Packard to look for corrupted file labels and other file system problems.

Q. Why do we ask users to run FSCHECK before running De-Frag/X?

A. MPE/iX systems may have corrupted file labels that the system manager may be unaware of. Corrupted file labels can remain undetected for years, especially if no one ever attempts to access those files. Some types of corrupted labels won't even show up if you do access the files.

De-Frag/X is extremely cautious with your data. If it detects anything wrong with a file, it will skip over that file and report the error. In fact, De-Frag/X checks for these errors (and more), and using a Hewlett-Packard supported utility will satisfy the Response Center requirements for proof of a problem.

Q. FSCHECK reported several (or hundreds, or thousands, or....) of errors that look like this:

"WARNING: FILE LABEL CORRUPTION DETECTED FOR:..(153,-91)
File label virtual address of corrupted file is $000000D1.$00061200.

What's wrong?

A. FSCHECK reports this particular error (153,-91) when it encounters any open files. You can safely ignore this error. Note that you really should run FSCHECK when NO other users or jobs are running.

Q. What should I do if FSCHECK encounters any other kinds of errors?

A. Save the output from FSCHECK and contact the HP Response Center for advice.

Q. Should I use the FIX option within FSCHECK to correct any errors reported?

A. NO! The HP Response Center says that you are in great danger of damaging your system if you try to use FSCHECK to do repairs. As stated above, save the output from FSCHECK and contact the HP Response Center for advice.

Q. Can I run FSCHECK in batch?

A. Absolutely not! There are situations when running FSCHECK in batch can cause system failures.

Q. Is running FSCHECK mandatory before running De-Frag/X?

A. No, we recommend running FSCHECK for customers' confidence only. However, we believe that it would be irresponsible for us not to recommend running FSCHECK.

B

RUNNING DE-FRAG/X IN BATCH

Included with De-Frag/X is a jobstream (JDEFRAGX.DEFRAGX.LPS) which can be used to run De-Frag/X in batch mode. This jobstream is set up to perform a MAP command on each ldev before and after an execution of the TRIM command on all files and the CONDENSE commands on all drives. You should modify this jobstream to reflect the particular drives on your system that you wish to MAP and/or CONDENSE.

The following sequence of commands demonstrates the JDEFRAGX jobstream:

```
!job jdefragx,manager.sys; outclass=,1
!
!tellop Starting JDEFRAGX...
!
!showtime
!run defragx.defragx.lps
set permyes
set log
reset fscheck
analyze
map all
map each
trim @.@.@
cond all
map all
map each
exit
!
!showtime
!tellop ...JDEFRAGX all done!
!
!eoj
```

C

QUICK REFERENCE TASKS GUIDE

Following are three tables of common tasks performed using De-Frag/X. These three tables consist of Analysis Tasks, Data Manipulative Tasks and Utility Tasks. More detailed information about all of these commands is available in the Commands List in Chapter 3, in the on-line help facility within De-Frag/X, or in the De-Frag/X User Reference Manual available from Lund Performance Solutions.

Analysis Tasks Description	De-Frag/X Command Sequence
Measure fragmentation on all disk drives	**MAP ALL**
Measure fragmentation on one disk drive	**MAP<ldev>** where ldev is desired logical device # or **MAP EACH** will map each ldev
Measure fragmentation on one "chunk" of a MAP output	**ZOOM ROW=<row> COLUMN=<column>** where row and column dictate the chunk's position in the MAP command output
Find fragmented files on all disk drives	**ANAlyze MIN#** where # is the minimum fragmentation percentage to report
Find fragmented files on one disk drive	**ANAlyze <ldev>** where ldev is desired device #
Find fragmentation within a particular file	**DisplayExtents <filename>**
Counts the files on each disk	**COUNTFiles <EACH>**

Data Manipulative Tasks Description	De-Frag/X Command Sequence
Measure fragmentation on all disk drives	**MAP ALL**
Remove disk level fragmentation on all disk drives	**CONDense ALL**
Remove disk level fragmentation on one disk drive	**CONDense <ldev>** where ldev is desired device #
Create a desired amount of additional free space on a particular disk drive	**MAKEROOM <ldev> <mb>** where ldev is the desired device # and mb is megabytes of free space at add

Data Manipulative Tasks Description	De-Frag/X Command Sequence
Remove fragmentation within a particular file/fileset	**DEFRAGment <filename | fileset>**
Release unused space after the EOF on a particular file/ fileset	**TRIM <filename | fileset>**
Balance the set of ldevs in the volume set specified	**BALance <ldev>** where ldev is the device to be balanced
Fragments the specified file(s)	**FRAGment <filename | fileset>**

Utility Tasks Description	De-Frag/X Command Sequence
Measure fragmentation on all disk drives	**MAP ALL**
Measure fragmentation on one disk drive	**MAP<ldev>** where ldev is desired logical device # or **MAP EACH** will map each ldev
Measure fragmentation on one "chunk" of a MAP output	**ZOOM ROW=<row> COLUMN=<column>** where row and column dictate the chunk's position in the MAP command output
Find fragmented files on all disk drives	**ANAlyze MIN#** where # is the minimum fragmentation percentage to report
Find fragmented files on one disk drive	**ANAlyze <ldev>** where ldev is desired device #
Find fragmentation within a particular file	**DisplayExtents <filename>**
Counts the files on each disk	**COUNTFiles <EACH>**

INDEX

N

R

S

T