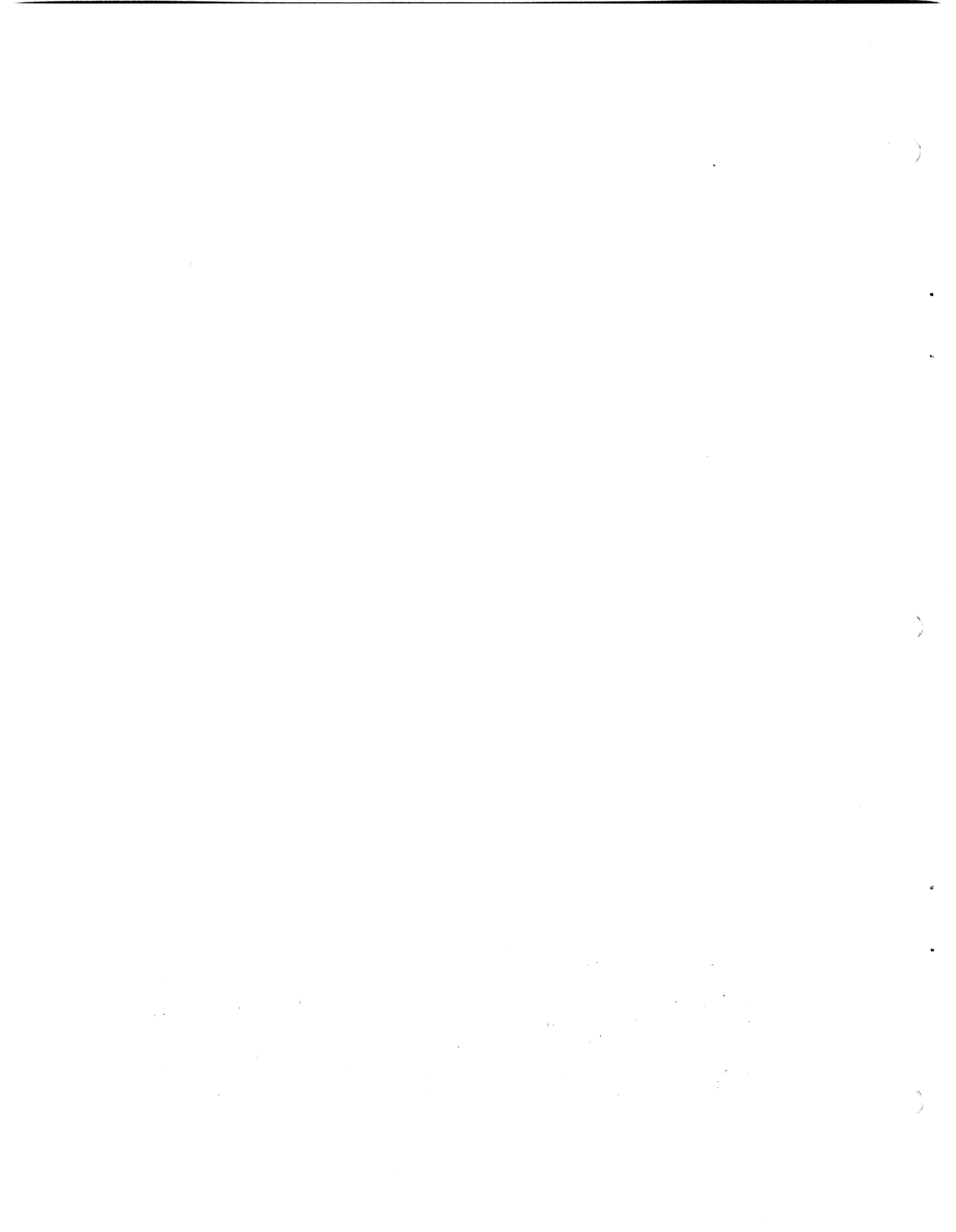


LCM
LOTUS CACHE MEMORY
SOFTWARE MANUAL

POINT 
DATA CORPORATION

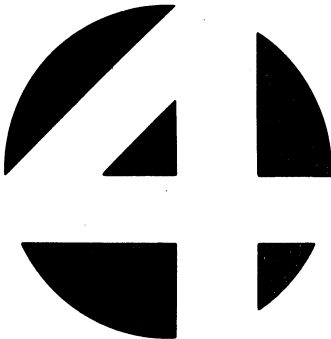


POINT 4 DATA CORPORATION

2569 McCabe Way / Irvine, California 92714

LCM

LOTUS CACHE MEMORY SOFTWARE MANUAL



NOTICE

Every attempt has been made to make this manual complete, accurate and up-to-date. However, all information herein is subject to change due to updates. All inquiries concerning this manual should be directed to POINT 4 Data Corporation.

Copyright © 1981, 1982 by POINT 4 Data Corporation (formerly Educational Data Systems, Inc). Printed in the United States of America. All rights reserved. No part of this work covered by the copyrights hereon may be reproduced or copied in any form or by any means--graphic, electronic, or mechanical, including photocopying, recording, taping, or information and retrieval systems--without the written permission of:

POINT 4 Data Corporation
2569 McCabe Way
Irvine, CA 92714
(714) 754-4114

REVISION RECORD

PUBLICATION NUMBER: SM-160-0025

<u>Revision</u>	<u>Description</u>	<u>Date</u>
01	Preliminary Release	11/16/81
02	Update incorporating Review edits	12/8/81
A	Initial Release	03/15/82

LIST OF EFFECTIVE PAGES

Changes, additions, and deletions to information in this manual are indicated by vertical bars in the margins or by a dot near the page number if the entire page is affected. A vertical bar by the page number indicates pagination rather than content has changed.

<u>Page</u>	<u>Rev</u>	<u>Page</u>	<u>Rev</u>	<u>Page</u>	<u>Rev</u>
Cover	-				
Title	-				
ii thru viii	A				
1-1 thru 1-4	A				
2-1 thru 2-17	A				
3-1 thru 3-3	A				
4-1 thru 4-3	A				
5-1 thru 5-3	A				
A-1 thru A-4	A				
B-1	A				
Comment Sheet	A				
Mailer	-				
Back Cover	-				

PREFACE

This manual is intended for users of LOTUS Cache Memory (LCM), a solid state device designed to improve throughput and response time for POINT 4 and NOVA-type minicomputer systems. Utilizing IRIS (Interactive Real-Time Information System), LCM is used as a cache memory for storage of commonly used disc blocks to increase throughput and reduce response time.

The manual includes basic features, operation and step-by-step installation procedures. Complete LCM configuration information includes specifying ranges, building range tables, and activating the LCM. The manual contains full data on the \$LCM driver and associated software developed for interface to IRIS. It covers LCM maintenance programs and includes inquiry into LCM configuration, its modification, and disc update.

Related manuals include:

<u>Title</u>	<u>Pub. Number</u>
LCM Hardware Manual	HM-160-0026
IRIS Manager Manual	SM-030-0003
IRIS User Manual	SM-030-0004
IRIS R7.5 Updates	



CONTENTS

<u>Section</u>	<u>Title</u>	<u>Page</u>
1	INTRODUCTION	1-1
1.1	General Description	1-1
1.2	LOTUS Cache Memory Theory of Operation	1-3
1.3	User Interface	1-4
2	LCM CONFIGURATION	2-1
2.1	Creating an LCM.PARAMS Text File	2-2
2.1.1	LCM Device Options	2-4
2.1.2	Range Specification	2-5
2.1.2.1	Active Files	2-5
2.1.2.2	File Ranges	2-5
2.1.2.3	User-Specified Absolute RDA Ranges	2-7
2.1.3	Case Studies	2-8
2.1.3.1	Case Study I - Development Environment	2-9
2.1.3.2	Case Study II - Large Development	2-11
2.1.3.3	Case Study III - Heavy Interactive	2-13
2.1.3.4	Case Study IV - Heavy Interactive and Some Data Processing	2-14
2.1.3.5	Case Study V - Compute and Disc-Bound Processing	2-15
2.2	Building Ranges Tables	2-16
2.2.1	Range Table	2-16
2.2.2	Creating the Range Table	2-16
2.3	Activating the LCM	2-17
3	\$LCM DRIVER	3-1
3.1	\$LCM Theory of Operation	3-1
3.2	Troubleshooting	3-2
3.3	Interface to \$LCM	3-3
4	LCM MAINTENANCE PROGRAMS	4-1
4.1	Maintaining LCM Programs	4-1
4.2	Updating the Disc	4-2
4.3	Checking Current LCM Configuration	4-3

5	INSTALLATION PROCEDURE	5-1
5.1	Installation	5-1
5.2	Changes to LCM	5-3

APPENDICES

A	Range Table Layout	A-1
B	LCM.ACTIVE File Structure	B-1

FIGURES

A-1	Range Table Format	A-1
-----	--------------------	-----

TABLES

A-1	Range Table Field Definitions	A-2
-----	-------------------------------	-----

Section 1

INTRODUCTION

1.1 GENERAL DESCRIPTION

The LOTUS Cache Memory (LCM) is a hardware device which can significantly enhance both throughput and response time under IRIS, especially with a large number of users. It is used as a cache memory for storage of commonly used disc blocks. The purpose of this device is to reduce to zero the seek and latency times for access to commonly used disc blocks.

Getting data from disc requires two time delays, known as "seek" and "latency" times. "Seek" is the time required for the heads to move to the cylinder on the disc where the desired data is stored. "Latency" is the time required for the exact data location on the cylinder to revolve around to be under the heads. These delays may seem infinitesimal in human terms, but a millisecond delay is significant to a computer that executes in nanoseconds. When many disc accesses are made, the effect is to slow system throughput and response time. LCM is designed to allow frequently-required disc blocks to be accessed with no seek and latency time and so to speed up system performance.

In addition to accessing data files, IRIS also accesses the disc for processors, programs, and discsubs.

Processors provide language support and other specialist functions. Examples of processors include RUN for executing active BASIC programs, BASIC for the entry and editing of BASIC programs, and EDIT for editing text files.

At any time, only one processor can be in memory. If two users are using different processors, each swap requires another disc access for the new processor to be read into the memory processor area.

IRIS divides memory into user partitions where user programs are held during execution. If there are more system users than partitions, then each time a partition is required by another user, two disc accesses must be made: one to copy the current program out to disc and one to bring the new program into the memory partition.

Discsubs are subroutines which are called regularly and so must be easily accessible, but generally are not used often enough to justify being stored in memory permanently. The System Manager may define some discsubs as core-resident. When a core-resident discsub is called, it is executed in place so no access time is required. When a non-core-resident discsub is called, it is read from a disc into the Subroutine Swap Area, executed, and then thrown away when finished. If one discsub calls another, the first is written out to disc into the Discsubs Nesting Blocks and saved until the second is finished. The first is then read back into the Swap Area from the disc and execution continues.

When the IRIS initialization routine is complete, all memory not required for the system itself, the processor area, and the partition area, is allocated to the buffer pool. This is a dynamically allocated pool of memory buffers designed to maximize disc-file I/O efficiency. All single-block disc transfers are made through the buffer pool, greatly reducing the number of actual disc transfers for frequently-used data.

NOTE

LCM users are advised to allocate memory for only one partition so the buffer pool can be as large as possible. In conjunction with making all user active files LCM-resident, this will help reduce the number of disc accesses and speed up system performance. See the IRIS Manager's Manual (Section 2.4.1) for this procedure.

LCM is designed to allow maximum flexibility. The user controls which areas of the disc will be LCM-resident. The most frequently used processors, programs, and data files are specified via a configuration procedure to the LCM driver. These areas are known as LCM Ranges, and are discussed in Section 2.

The following sections describe the functions and features of the \$LCM driver and all other associated software provided to interface to the IRIS Operating System. This LCM software is available on any IRIS R7.5 installation capable of supporting LCM hardware.

1.2 LOTUS CACHE MEMORY THEORY OF OPERATION

To determine which disc blocks will be resident in the LCM, areas of high disc usage must be identified. These disc areas are known as ranges and will be used as configuration information to be stored in a memory-resident Range Table.

During LCM activation, the blocks specified in these ranges are copied into the LCM. Once activated, operating system software will intercept all transfers and determine whether the required block(s) lie in an LCM range. If so, the transfer will be made between memory and the LCM rather than between memory and the disc. This eliminates disc access time for that transfer and thus improves system performance.

When a logical unit with LCM-resident blocks is REMOVED from the system or the entire system is SHUTDOWN, the system will flush any ranges with updated blocks back to the disc.

NOTE

This may considerably slow down the SHUTDOWN or REMOVE process.

1.3 USER INTERFACE

The average user of an IRIS System with a Lotus Cache Memory can interface with the system normally. However, he should immediately observe an improvement in system performance, with significantly higher throughput and faster response time. If LCM is removed from the system after it has been in use for some time, the difference in performance will be very noticeable.

Section 2

LCM CONFIGURATION

To improve performance to a maximum, key areas of high disc access must be located by testing system operation or by a knowledge of the application programs. Once a number of areas of relatively high disc access are selected, these areas must be specified via a configuration procedure to the LCM driver. These areas are known as LCM Ranges.

Each range consists of specified contiguous disc blocks on a single Logical Unit. When the LCM is activated, all blocks specified as LCM ranges are copied into the LCM and the range is initialized for LCM operation. All further accesses, reads or writes from the operating system will be made to or from the LCM rather than the disc.

The operating system does not require the use of linked lists or chains to access LCM. It checks to see if the needed disc block is within the ranges specified for the LCM.

Before a range can be activated, certain information must be specified in the text file named LCM.PARAMS. This file is used by the program LCMCONFIGURE to determine how the blocks in that range may be handled by the LCM driver.

All configuration information is read from the text file LCM.PARAMS and formatted for Range Table use by the program LCMCONFIGURE. During the next IPL of the system these values will be loaded as part of the LCM activation.

Activating the LCM is a procedure similar to installing a logical unit. The processor LCMACTIVATE performs several checks on the information in the text file LCM.ACTIVE. It then uses this information to down-load the Range Tables into the memory location designated for LCM Range Tables in the \$LCM driver. LCMACTIVATE then proceeds to allocate LCM space and copy the specified range blocks into the LCM.

2.1 CREATING AN LCM.PARAMS TEXT FILE

The LCM.PARAMS text file is created by utilizing the EDIT command (See IRIS - Interactive Real-Time Information System User Manual, Section 3.1). The EDIT command allows the text to be either typed in, or if available on paper tape in ASCII code, to be loaded through a terminal's tape reader. POINT 4's FORGE and other text editors may also be used.

Each line of the text file LCM.PARAMS specifies one LCM range. Two types of files may be specified:

- Active files (i.e., swapping files on LU/0)
- Real files

Range information may be specified in Real Disc Addresses (RDA) in octal or by filenames, as appropriate. Several files may be specified to be in the same range by separating filenames with commas (e.g., RUN, RUNMAT, BASIC). Real files and Active Files may not be mixed in the same range.

Each line in the LCM.PARAMS file is limited to 72 characters. If the required information for a range will not fit on a 72-character line, it will be necessary to use a continuation format. The continuation format consists of an ampersand (&) as the final character of the first range line. The ampersand must replace the filename separator (,). The & indicates that the line to follow is to be regarded as a continuation of the present line.

If the files specified for a range (i.e., on the same line) are noncontiguous on disc, the LCMCONFIGURE program allocates a contiguous LCM range starting with the first noncontiguous block and ending with the last noncontiguous block. Other files, or parts of other files which lie within the specified area are included.

If a range is defined as Update (see Section 2.1.2), and also contains blocks of other files, those blocks are made Update also, even if they were not individually specified in LCM.PARAMS.

With random files it is possible that the first block may not have the lowest disc address, and the last block may not have the highest disc address. All blocks between the lowest and highest addresses of file blocks will be included in the range.

It is probable that when a noncontiguous file is specified for a range, part or all of other files will lie between the highest and lowest blocks of the file and will inadvertently be included in the range. When this happens, if even a single block of a file is included in a range, and that file is later specified for inclusion in another range, an error will result.

When this error occurs, it may be corrected by specifying the file giving the error in the same range as the surrounding noncontiguous file.

Having too many ranges will cause excess overhead; placing nonconsecutive files on the same line will waste the intervening LCM memory. If noncontiguous files are specified, it is advisable to run CLEANUP first, and avoid wasting LCM memory. Active files are set up differently, and may be specified on one line or several lines without penalty (see Section 2.2.1).

2.1.1 LCM DEVICE OPTIONS

In addition to entry of all range specifications, the LCM.PARAMS may contain several nonrange-specific features of the device.

- Hamming enabled - Error detection and correction code. Enables detection and correction of any single-bit error and detection of any two-bit error in a 64-bit field (4 words) transferred between LCM and CPU.
- Battery Backup enabled - Not currently available.
- Size - Number of blocks in decimal. LCM boards are available in four sizes:

- 256 blocks (128K bytes)
- 512 blocks (256K bytes)
- 1024 blocks (512K bytes)
- 2048 blocks (1M byte)

Up to 32 boards can be installed in a processor, providing a maximum of 65,536 blocks per device code, limited by chassis size and power supply.

The user may specify to LCM that either errors or errors and transfers may be counted. These counts provide valuable performance statistics but also consume space. They should be included unless space is extremely scarce.

- Error count - Number of errors per range is recorded. This allows you to detect hardware degradation before absolute failure.
- Transfer count - Number of transfers per range is recorded.

The program LCMCHECK may be used to display these parameters.

To select any of these options in the LCM.PARAMS file, enter an asterisk (*) and the parameter name, as shown below:

```
*HAMMING
*BACKUP
*XFERCOUNT
*CERRCOUNT
*SIZE=n
```

where n is the number of blocks. If multiple boards are used, indicate the combined total number of blocks. If "n" is specified as less than the LCM actual size, the specified size will be accepted and the remainder of the LCM will not be used. If "n" is specified as greater than the LCM actual size, LCM will not be able to be activated.

These parameters must be specified at the beginning of the LCM.PARAMS file. The device options may be given in any order.

2.1.2 RANGE SPECIFICATION

2.1.2.1 Active Files

Each user has a unique area on LU/0 for program swapping called the Active File. Active Files are identified by the port number displayed when the user logs on.

To specify Active Files to the LCM, add a line to LCM.PARAMS beginning with an "at" sign (@). For example

```
@1,2,3,7
```

specifies that Active Files for ports 1, 2, 3 and 7 are to be LCM-resident.

A range of port numbers may be specified by separating them with dashes. For example

```
@1-5,7-10
```

specifies that Active Files for ports 1 through 5 and 7 through 10 are to be LCM-resident.

2.1.2.2 File Ranges

Ranges apply only to single logical units, so the logical unit number is specified at the beginning of the line followed by a slash (/).

For each range, range type information is required by the \$LCM driver. The possible types are:

- Copy-thru (C) - Any writes to this range in LCM should also be written to the disc immediately. This feature may be specified where the risk of power-fail is high and the data in the LCM is not protected by battery backup.
- Update (U) - This indicates that the data in this range is expected to change and that it is desirable to copy all data back to the disc before the logical unit is REMOVED from the system or the system is SHUTDOWN.

Specify one of these letters at the end of the range line preceded by a colon (:).

Multiple data files, programs and processors may be specified as a single range by placing the names on a single line, separated by commas.

In the case of a contiguous file, an equal sign (=) may be used to specify the number of blocks to be included in the range. For example

FRED=100

places the first 100 blocks of the file FRED into the LCM range. You can QUERY the file to get the total number of blocks in the file. Note that indexed files are contiguous files with the directory section in the beginning blocks. It is possible to use SEARCH Mode 1 (see IRIS Users Manual, Section 12.9) to ascertain the record number of the first real data record. This is the first data record beyond the directories. The number of this record can be divided by the number of records per block to determine the number of blocks required for the directory. The file directory alone may be included in the range by specifying this number of blocks. If the number of blocks is not specified, the entire file will be included in the range.

For loading Discsub Nesting Blocks to LCM, a special pseudo file exists. This file, DNESTBLKS, lies on Logical Unit 0 and specifies that the Discsub Nesting Blocks be loaded to LCM, using the following format:

0/DNESTBLKS

No range type is specified because this range is LCM-resident only and is never written out to disc.

An example of a complete range specification as it would appear in LCM.PARAMS follows:

2/FILE1,FILE2=100:U

where

2/ - Specifies the files are on Logical Unit 2 of the disc

FILE1 - Specifies the name of a file. The entire contents of this file will be LCM-resident. If file is noncontiguous, all data residing between the first block of this file and the last block will also become LCM-resident.

, - filename separator

FILE2=100 - Specifies the name of a file and the number of blocks to be LCM-resident. Only the first 100 blocks of this file will be LCM-resident. If the file length is less than the specified number of blocks, file length is used. This type of entry is valid only for contiguous files.

:U - Specifies range type information indicating that the entire range will be copied to disc only upon Removal of Logical Unit 2 or SHUTDOWN of the system.

2.1.2.3 User-Specified Absolute RDA Ranges

Absolute Real Disc Address (RDA) ranges may be specified by indicating the logical unit and the first and last RDAs of the range (expressed in octal). For example

5/10-1000:C

where

5/ - Specifies Logical Unit 5

10-1000 - Specifies the RDA range (in octal)

:C - Specifies range type information ("Copy-thru", in this case)

2.1.3 CASE STUDIES

One of the benefits of the LCM is that it allows great user flexibility. Disc use may be assessed for individual environments so that areas of highest use may be made LCM-resident. The precise specification of the LCM.PARAMS text file will depend on the job mix and the size of the LCM.

In determining which areas to make LCM-resident, the user should consider which processors, programs, discsubs, and data files are accessed most often.

Generally, two accesses of Active Files are made each time a partition is required by another user. Therefore assigning Active files to the LCM is recommended. Assigning discsubs and Discsub Nesting Blocks significantly reduces the number of disc accesses and is strongly recommended. Assigning the most frequently-used processors to the LCM is also recommended. Depending on the environment, these may include BASIC, RUN, RUNMAT, EDIT, and SCOPE. In some environments, assigning LU Indexes, ACCOUNTS files, file directories, and frequently-accessed datafiles is advantageous.

The following section contains case studies of some typical LCM/IRIS installations. The user may simply extract the sample PARAMS file for the environment most similar to his own. Or, if he prefers, he may customize a PARAMS file by merging aspects of several samples included here to produce the most advantageous configuration for his environment.

The case studies include a hardware/software configuration, followed by a sample range specification for that configuration. These configurations all assume a BASIC program area of 7774 words. If the program area is smaller, more active files may fit in the LCM.

CLEANUP should be used on all logical units from which sections are to be assigned to the LCM. This ensures 'packing' of the files and minimum use of LCM space.

The size of the LCM in each study has been selected as that most suitable for the application.

2.1.3.1 Case Study I - Development Environment

Configuration:

1. 256K-byte LCM
2. 128K-byte CPU
3. 16-port Mux
4. 2 printers, 14 interactive ports, 5 phantom ports
5. Development of BASIC programs for typical end-user applications.

Sample Range Specification:

Entry	Comments
@1,5-12	Ports 0-4 are phantoms but only #1 is regularly used so it is made LCM-resident. The active files of the busiest users (ports 5-12) are also included in this range. If necessary to conserve space, this range could be omitted.
0/DNESTBLKS	No range type is specified because this range is never written to disc.
0/INDEX,DISCSUBS:C	This range will speed access to discsubs, and access to most processors. This range of blocks will always include the DMAP, NESTING BLOCKS, and ACCOUNTS files. It is set as copy-thru because the 'cost' is small (most of these areas are only read) and the recovery advantage is great.
0/BASIC,RUN,RUNMAT:C	These processors are the most commonly used. They are not usually written, so there is little or no performance penalty for copy-thru.
0/EDIT:C	This range is recommended if EDIT is used for maintaining source files.

0/SCOPE:C

SCOPE is used so frequently.
Processor swapping occurs each
time a user goes to the # sign.

0/BYE

This range could be added if
there is enough space on the LCM.

In this environment it is unlikely that more than three users would be active in a relatively short period of time. Therefore we will maintain three resident partitions so there is no swapping.

2.1.3.2 Case Study II - Large Development

Configuration:

1. 1M-byte LCM
2. 128K-byte CPU
3. 32-port Mux
4. 3 printers, 29 interactive ports, 5 phantom ports
5. Development of BASIC programs for typical end-user applications. Some Assembly language programming.

Sample Range Specification:

Entry	Comments
@0-12,16-36	Active files for all interactive ports. The files for the ports assigned to the printers are not placed in LCM.
0/DNESTBLKS	No range type is specified because this range is never written to disc.
0/INDEX,DISCSUBS:C	Discsub and index access is accelerated. As in Study I the ACCOUNTS file, DMAP and NESTING BLOCKS are included within this range.
0/BASIC,RUN,RUNMAT:C	These processors are the most commonly used. They are not usually written, so there is little or no performance penalty for copy-thru.
0/EDIT,ASM:C	Speeds up swapping to the assembler processor.
0/SCOPE:C	SCOPE is used so frequently.
1/INDEX,ACCOUNTS:C	Speeds up access to INDEX, ACCOUNTS and DMAP on LU/1. (DMAP is included in INDEX).
2/INDEX,ACCOUNTS:C . . .	Repeat for each logical unit to speed up access to INDEX, ACCOUNTS and DMAP on each LU.

Since there are many more users than possible partitions, the number of core-resident partitions should be reduced to one to maximize the buffer pool. Assign all interactive active files to LCM.

ACCCOUNTS files should be placed in LCM since they are accessed every time a file is built or extended.

2.1.3.3 Case Study III - Heavy Interactive

Configuration:

1. 512K-byte LCM
2. 128K-byte CPU
3. 32-port Mux
4. 3 printers, 29 interactive ports, 3 phantom ports
5. Very short interactions, small amount of processing or disc activity per input.

Sample Range Specification:

Entry	Comments
@0-12,16-34	Active files for all interactive ports. The files for the ports assigned to the printers are not placed in LCM.

Since there are many more users than possible partitions, the number of core-resident partitions should be reduced to one to maximize the buffer pool. Assign all interactive active files to LCM.

If apparent response time is to be improved still further, the short time slice may be reduced to 2 or even 1-tenth of a second.

2.1.3.4 Case Study IV - Heavy Interactive and Some Data Processing

Configuration:

1. 1M-byte LCM
2. 128K-byte CPU
3. 32-port Mux
4. 3 printers, 29 interactive ports, 3 phantom ports
5. Very short interactions, small amount of processing or disc activity per input. Some access to an inventory file and customer file for Sales Order Processing.

Sample Range Specification:

Entry	Comments
@0-12,16-34	Active files for all interactive ports. The files for the ports assigned to the printers are not placed in LCM.
0/DISCSUBS:C	Speeds up access to discsubs.
0/RUN,SCOPE:C	These are the only two processors regularly used.
3/STOCKINDEX=500:U	The directory of the stock file.
4/CUSTINDEX=300:U	The directory of the customer file.

Since there are many more users than possible partitions, the number of core-resident partitions should be reduced to one to maximize the buffer pool. Assign all interactive active files to LCM.

If apparent response time is to be improved still further, the short time slice may be reduced to 2 or even 1-tenth of a second.

2.1.3.5 Case Study V - Compute and Disc-Bound Processing

Configuration:

1. 1M-byte LCM
2. 128K-byte CPU
3. 16-port Mux
4. 2 printers, 14 interactive ports, 2 phantom ports
5. The running software is predominantly compute and disc-bound. It makes extensive use of formatted work files.

Sample Range Specification:

Entry	Comments
@0-11,14-17	Active files for all interactive ports. The files for the ports assigned to the printers are not placed in LCM.
0/DNESTBLKS	No range type is specified because this range is never written to disc.
0/DISCSUBS:C	Speeds up access to discsubs.
0/RUN,SCOPE:C	These are the only two processors regularly used.
17/1-1250:U	This space and logical unit is used merely to create and access work files.

Since there are many more users than possible partitions, the number of core-resident partitions should be reduced to one to maximize the buffer pool. Assign all interactive active files to LCM.

In this example the user has created a small logical unit, number 17, to provide an area for relatively fast access to work files. All work files are built on this logical unit, permitting maximum performance advantage to be achieved.

2.2 BUILDING RANGES TABLES

2.2.1 RANGE TABLE

The major interface between \$LCM and the LCM support modules is the Range Table. Almost all configuration and system specific information is included here. The entire table is created by running the LCMCONFIGURE program. Only the error and access counts and the active flags are modified by the \$LCM driver itself. The format of the table and an explanation of Range Table fields is given in Appendix A.

2.2.2 CREATING THE RANGE TABLE

LCMCONFIGURE is a BASIC program used to set up the Range Tables. The program reads the text file LCM.PARAMS which specifies the filenames and/or Real Disc Addresses to be allocated to LCM. This file also includes manager-specified information such as the size of the device.

To create the Range Table, enter the following command at the system prompt (#):

```
LCMCONFIGURE <RETURN>
```

The following paragraphs describe the operation of this program.

LCMCONFIGURE checks the consistency of the information in LCM.PARAMS and verifies that no Real Disc Address is mapped into LCM twice. It also checks that the amount of disc mapped into the LCM does not exceed the specified size of LCM. At the operator's option, LCMCONFIGURE creates a file to log all configuration errors and actual RDA ranges that were created. By default this log is output to the operator's terminal.

LCMCONFIGURE generates an output file, LCM.ACTIVE, in which the Active File information and range table are created. This file is used by LCMACTIVATE to down-load the required information into the memory-resident copy of the \$LCM driver. The structure of this file is given in Appendix B.

After configuring the Range Table, LCMCONFIGURE chains to the processor LCMMODIFY to alter the size of the Range Table in \$LCM; therefore, the IRIS system must be IPLed after the Range Table size is modified. LCMACTIVATE will refuse to activate the LCM if this is not done.

2.3 ACTIVATING THE LCM

The LCM must be activated, a procedure similar to installing a logical unit. The processor LCMACTIVATE performs several checks on the information in the text file LCM.ACTIVE. It then uses this information to down-load the Range Tables into the memory location designated for LCM Range Tables in the \$LCM driver. LCMACTIVATE then proceeds to allocate LCM space and copy the specified range blocks into the LCM.

To activate the LCM, enter the following command at the system prompt (#):

```
LCMACTIVATE n <RETURN>
```

where "n" is an optional logical unit number to be activated. If this parameter is not supplied, LCMACTIVATE will activate all installed logical units. There are no other input parameters. The program outputs error messages and warnings when appropriate. The following paragraphs describe the operation of this processor.

LCMACTIVATE checks the Range Table to see if \$LCM is active. If it is not active, LCMACTIVATE checks that the size and options in use match those specified in the LCM.ACTIVE configuration table. If the match is not successful, the activation will fail immediately. If there is a successful match, the configuration specified in LCM.ACTIVE is down-loaded to LCM. \$LCM is now activated. However, ranges are not activated yet.

If the optional logical unit parameter has been specified, the processor activates that logical unit only. An error message is issued if the requested logical unit is not installed. If no parameter has been specified, all logical units are activated.

LCMACTIVATE reads the range tables and copies all blocks of the specified logical unit from the disc to the appropriate area in the LCM. It checks the specified range type in the Range Table to see if it is Update or Copy-thru. If the No Dirty Page indicator is set in the system information table (word 606 in CONFIG), LCMACTIVATE causes all Update ranges to be treated as Copy-thru ranges. Once the activation procedure is complete, the range is marked as active in the range flags of that Range Table entry.

A special method is used for Active Files which are allocated to the LCM. The Active Files are copied to new, contiguous files with pseudo-disc addresses numbered sequentially from 100000 octal. These files lie entirely within the LCM, outside the normal RDA range for Logical Unit 0. These Active Files are specified in the file LCM.ACTIVE, which contains a list of the port numbers whose Active Files are to be moved. The partition size is used in determining the size of these files. The entire Active File range is then allocated to LCM, insuring more efficient use of the LCM space.



Section 3

\$LCM DRIVER

3.1 \$LCM THEORY OF OPERATION

The \$LCM driver intercepts IRIS disc and buffer pool transfers when they are directed to LCM-resident blocks. The driver converts Real Disc Addresses into LCM block numbers.

When the \$LCM driver is initialized, instructions in IRIS's REX are replaced with calls to the driver. All disc I/O calls are intercepted by this driver. Each entry to the \$LCM Read/Write routine causes a check for \$LCM active status. Next, the Range Table is checked for LCM-resident blocks. A simple linear search is performed for LCM block RDAs matching read/write block RDAs. If the desired blocks do not lie within one of the LCM ranges, a return is made to REX.

For a read operation involving LCM-resident blocks, the blocks are read and a return is made to the user. In the case of a write operation involving LCM blocks, a check on the type of range is required. Where the range is Copy-thru, the data is written to the disc and then into the LCM. Update ranges are written to the LCM only.

3.2 TROUBLESHOOTING

\$LCM is given its parameters by LCMCONFIGURE and activated by LCMACTIVATE.

Certain LCM failures will cause Trap 3s to be reported to the operator. The accumulators A0, A1, A2 indicate the details of the problem. A0 is the transfer-block count, A1 is the LCM block number, and A2 indicates one of the failure modes shown below:

A2 Failure Mode

- 4 - Permanent I/O error
- 5 - Loss of data due to Power-Fail
- 6 - Data channel late
- 7 - Illegal LCM block number
- 10 - LCM timed out accessing a block

If LCM is used without battery backup, all of the data stored in the LCM will be lost during a power failure. All subsequent accesses to LCM will generate a Trap 3 with A2 equal to 5, indicating the data is not retrievable. All ranges marked "Copy-thru" will be correct on disc and can be restored to the LCM by performing an IPL. All other ranges may be missing data and may need to be restored from a backup.

3.3 INTERFACE TO \$LCM

System software is able to access the \$LCM driver through a single vector provided in page zero. The vector points to the Read/Write routine. A call through this vector is allowed to bypass all checks and the Range Tables. This is a Processor Call and exists for the purpose of activating, removing, modifying or any other system function requiring direct access to the LCM. This interface to the LCM is used by LCMACTIVATE, LCMMODIFY, and the RMLCM and FLLCM discsubs.



Section 4

LCM MAINTENANCE PROGRAMS

4.1 MAINTAINING LCM PROGRAMS

Although the basic modules for LCM operation are LCMCONFIGURE, LCMMODIFY, LCMACTIVATE and \$LCM, in a daily operating environment the operator may be required to perform several other types of LCM-related activities. These activities include inquiring into the current LCM configuration, modifying that configuration, and updating the disc. The following sections describe additional modules which allow the user to perform these functions.

4.2 UPDATING THE DISC

In order to update the disc with any data which has recently been updated in the LCM, two special discsubs are provided. RMLCM and FLLCM flush all blocks from the LCM to the disc for the specified Logical Unit for ranges marked as the Update type. The RMLCM flushes and deactivates ranges while FLLCM only flushes. The RMLCM discsub is automatically called by REMOVE and FLLCM is automatically called by SHUTDOWN when a logical unit is to be REMOVED or when the system is to be SHUTDOWN.

No direct operator interface to these modules is required. All parameters are passed to RMLCM and FLLCM from the IRIS Operating System REMOVE and SHUTDOWN modules.

These discsubs accept as a parameter the Logical Unit whose ranges are to be removed from LCM. All ranges are checked for assignment to the logical unit(s) to be removed. If the range is an Update type, all blocks in that range are copied back to the disc.

During system SHUTDOWN, all logical units are specified to FLLCM and any Update-type range is copied to disc before completion of SHUTDOWN.

4.3 CHECKING CURRENT LCM CONFIGURATION

LCMCHECK is a BASIC program that can be used by the System Manager to examine the status of the memory-resident \$LCM Driver. Note that LCMCHECK is only valid if run while the LCM is activated. The output includes all information set by LCMCONFIGURE plus any range counts enabled for errors or transfers. Data concerning the following items is displayed:

- Options installed
- LCM size
- Active ranges
- Number of transfers on each range
- Number of corrected errors on each range

To check the current configuration, enter the following command at the system prompt (#):

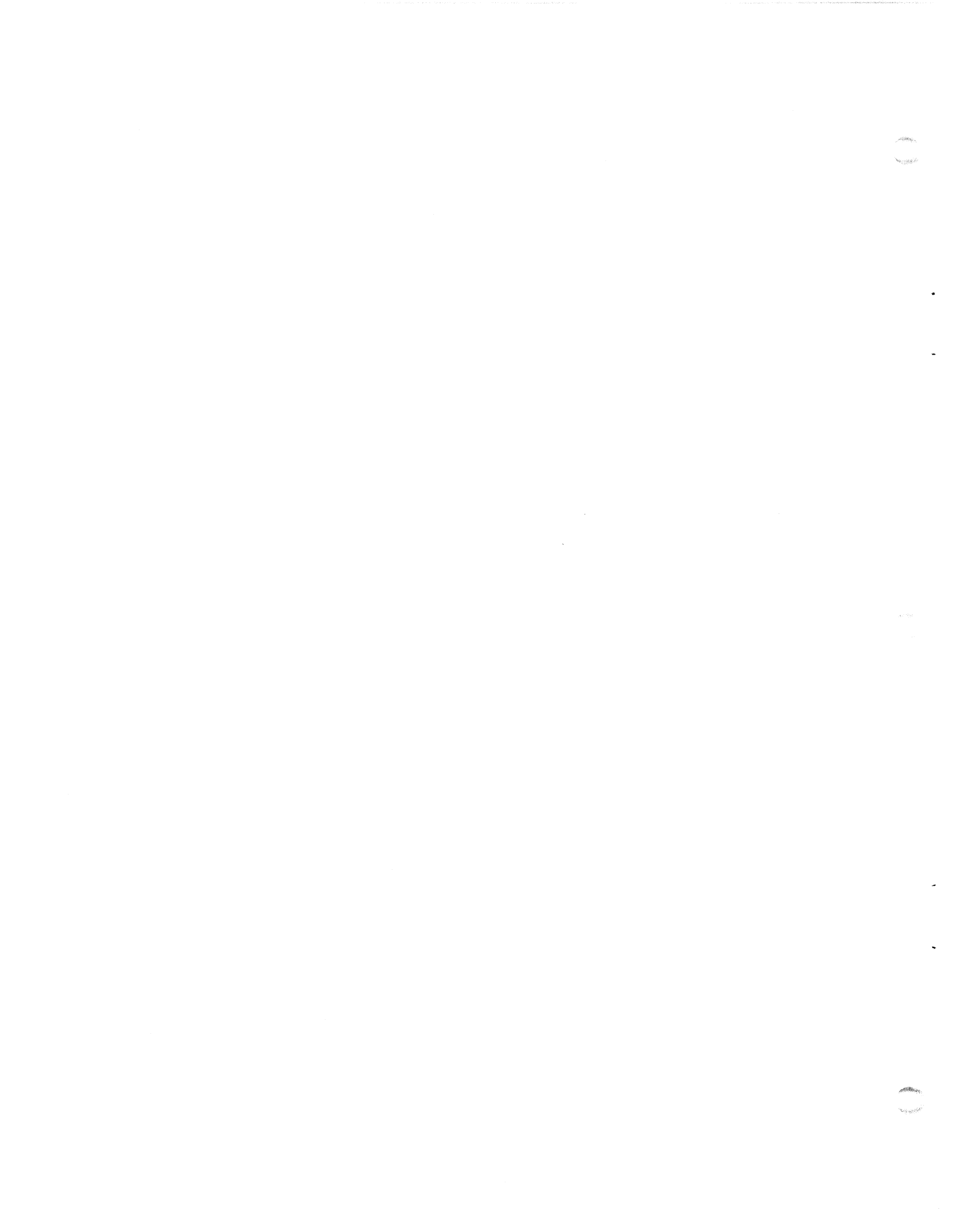
```
LCMCHECK <RETURN>
```

The system requests that the operator specify the device/file to receive output of the inquiry. The following entries are acceptable:

- Carriage return - output to be sent to the operator's terminal
- IRIS filename - output will be formatted into this text file. Contents will be identical to that output on the operator's terminal.
- An IRIS device (any filename starting with \$). The specified device will be opened for reception of output.

The first items output are the parameters set for the entire LCM. This is followed by the entries for the specified ranges. All Real Disc Addresses are specified in octal.

When output is specified to appear on the operator's terminal, a pause occurs after each twenty lines to allow examination of screen contents. To continue to the next twenty lines press <RETURN>.



Section 5

INSTALLATION PROCEDURE

5.1 INSTALLATION

The following is a step-by-step description of the procedure for installing the LCM. User input is underlined to distinguish it from system display.

1. Activate the LCM subsystem under IRIS 7.5 by activating the \$LCM driver. At the system prompt (#), enter

#CHANGE LCM <RETURN>

NEW NAME? \$LCM <RETURN>

NEW COST? <ESCAPE>

2. SHUTDOWN and re-IPL the system to link the loaded modules.
3. Organize Logical Unit #0 for optimum LCM usage by running CLEANUP. From the system command mode on the Manager's account, enter

#CLEANUP <CTRL-E>key<CTRL-E> 0 USING 1 <RETURN>

NOTE

A logical unit other than 1 may be used to CLEANUP LU #0.

4. Re-IPL the system.
5. Create and modify the desired LCM.PARAMS file (see Section 2.1.3).
6. Enter

#LCMCONFIGURE <RETURN>

Listing file (<RET> for CRT) : <RETURN>

In this example, assorted output will follow on the screen (see Section 2.2.2).

7. Upon return to the system command mode, Re-IPL the system.

8. Log onto the Manager's account and INSTALL all logical units; then activate the LCM by entering

#LCMACTIVATE <RETURN>

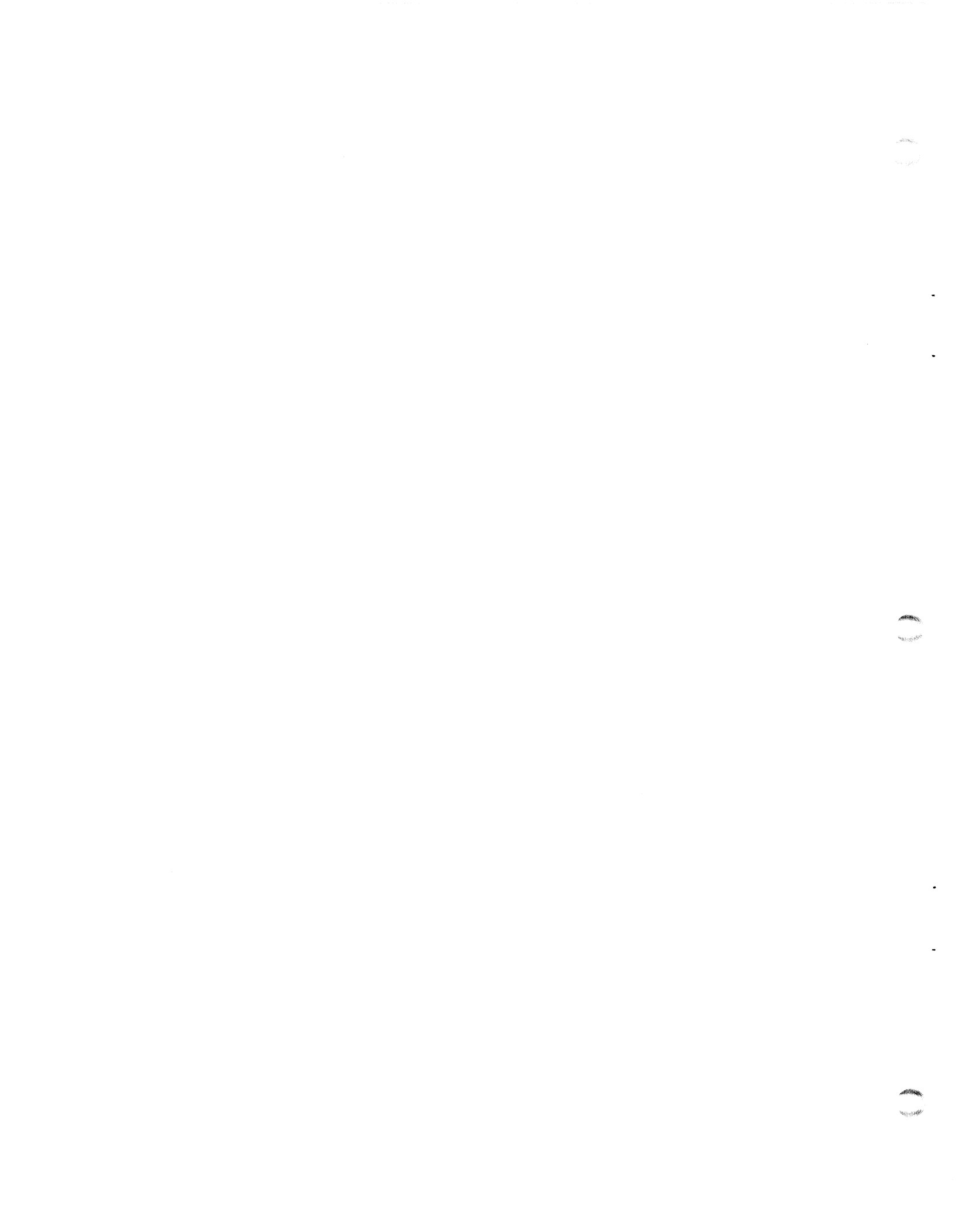
NOTE

LCMACTIVATE takes some time to complete; other users will run more slowly until it is finished. Once completed, LCM is activated, and all system users should see performance improvement.

5.2 CHANGES TO LCM

At various times during normal system operation it may be necessary to repeat some steps in the installation procedure for changes to the system. For example:

- LCMACTIVATE must be run after running INSTALL.
- LCMCONFIGURE must be run any time that files designated for the LCM are moved on disc. This will happen after running CLEANUP, replacing a data file which is listed in LCM.PARAMS, or re-SAVING a BASIC program listed in LCM.PARAMS. CHANGE does not move a file so it is not necessary to re-run LCMCONFIGURE after changing a filename.
- CLEANUP cannot be run while LCM is active. The system must be SHUTDOWN before running CLEANUP and then LCM must be activated again using the procedure given in Section 5.1.

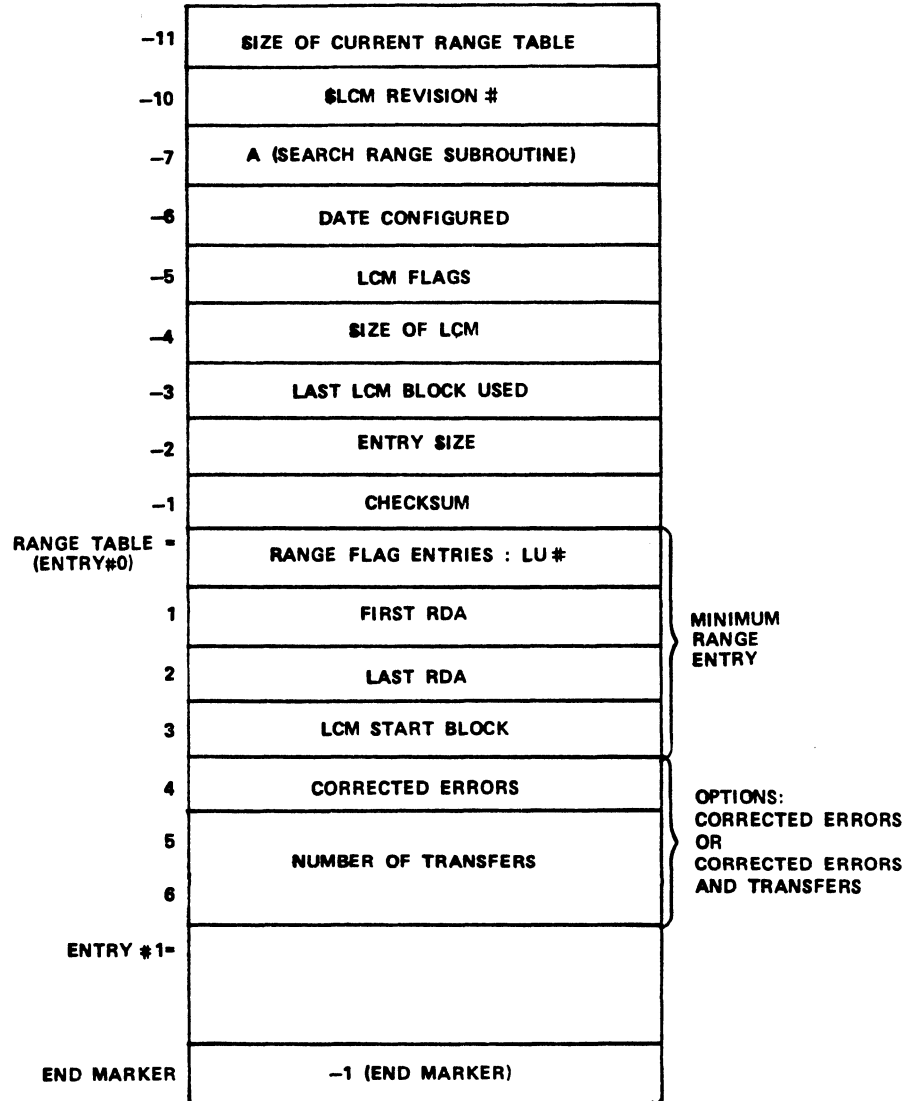


APPENDICES



Appendix A

RANGE TABLE LAYOUT



160-006

Figure A-1. Range Table Format

TABLE A-1. RANGE TABLE FIELD DEFINITIONS

Field	Definition
Size of Range Table	Size of current Range Table calculated by LCMCONFIGURE
\$LCM Revision #	\$LCM Driver assembly parameter
A Search Range Sub-routine	Address of Search Range subroutine in memory
Date Last Configured	Set by LCMCONFIGURE to current date. This reflects assembly date until the first LCMCONFIGURE.
LCM Flags	Bit Mask 1 octal - Hamming enabled Bit Mask 2 octal - Battery Backup installed Bit Mask 4 octal - \$LCM active Bit Mask 10 octal - Battery Backup functional Bit Mask 20 octal - \$LCM did not survive power-fail Bit Mask 40 octal - REX hooks not inserted* Bit Mask 100 octal - \$LCM driver initialized Bit Mask 200 octal - \$LCM file has been modified**
Size of LCM	Set by LCMCONFIGURE, measured in blocks.
Last LCM Block Used	Set by LCMCONFIGURE. The difference between this value and Size of LCM equals the number of LCM blocks available for other use.
<p>*IRIS Real-time Executive (REX) modifications are accomplished through LCM "REX hooks". This bit is set when modifications have been made.</p> <p>**LCMMODIFY sets this bit to force an IPL before the next LCM activation.</p>	

TABLE A-1. RANGE TABLE FIELD DEFINITIONS (Cont)

Field	Definition
Entry Size	<p>Specifies the size of each entry in the Range Table.</p> <p>Size = 4 indicates that no access counts or corrected error counts will be recorded.</p> <p>Size = 5 indicates that accesses only will be recorded.</p> <p>Size = 7 indicates that both are required.</p> <p>Size > 7 are for future use.</p>
Checksum	Not used.
Range Entry/ Flags	<p>Bit Mask 200 octal - This range is inactive</p> <p>Bit Mask 400 octal - Copy all writes to disc</p> <p>Bit Mask 1000 octal - Update; copy all writes to disc on SHUTDOWN or REMOVE</p> <p>Bit Mask 2000 octal - Do not load this range during LCMACTIVATE</p>
LU#	Logical Unit # for this range.
First RDA	First Real Disc Address included in this range. Set by LCMCONFIGURE.
Last RDA	Last Real Disc Address included in this range. Set by LCMCONFIGURE.
LCM Start	Generated by LCMCONFIGURE. Indicates the first LCM block in this range.

TABLE A-1. RANGE TABLE FIELD DEFINITIONS (Cont)

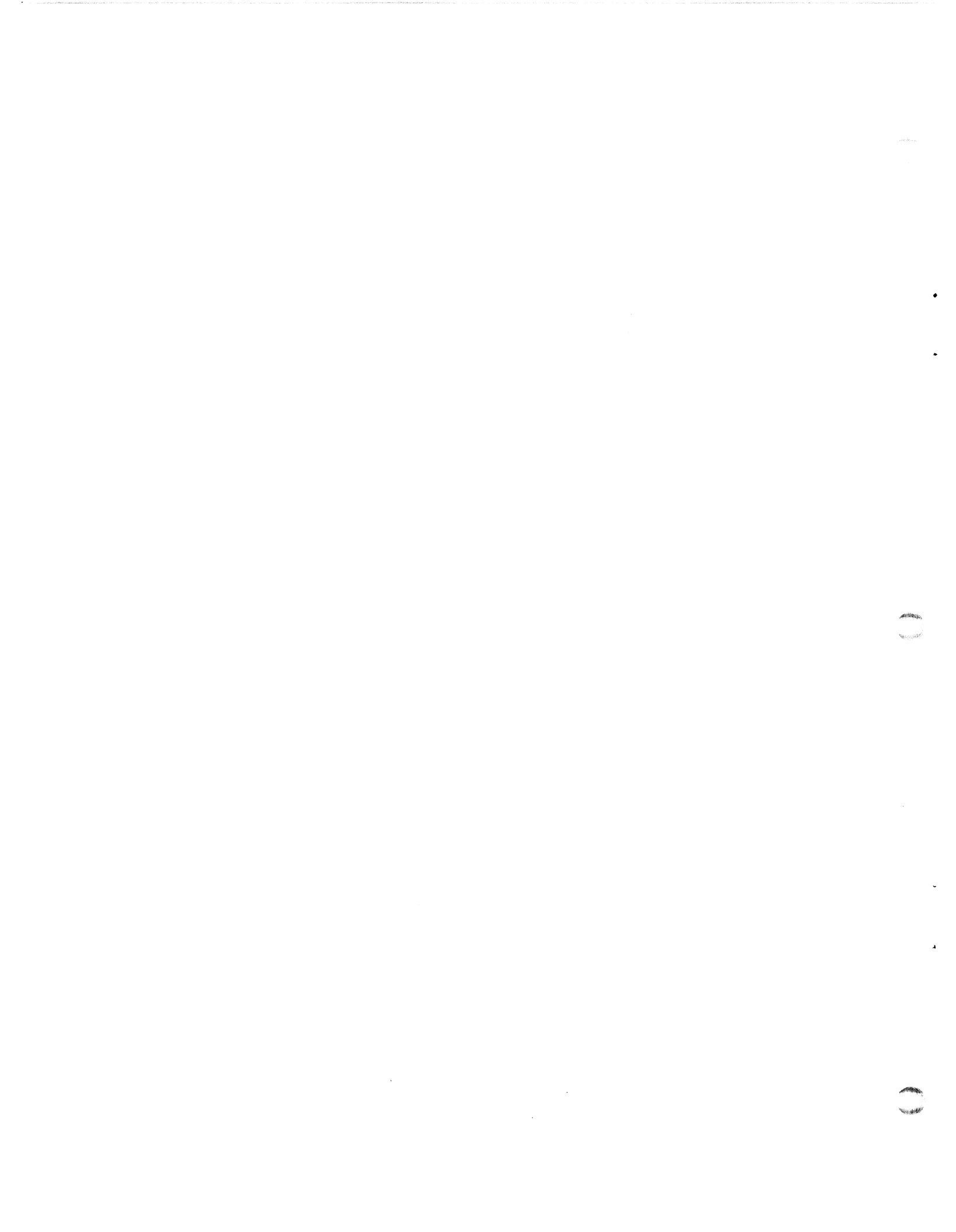
Field	Definition
Corrected Error Count	Incremented each time a corrected error is detected (optional field as determined by entry-size parameter).
Number of Transfers	Records the total number of accesses to this range. This field can be used to determine the most beneficial ranges to optimize system operation (optional field as determined by entry-size parameter).

Appendix B

LCM.ACTIVE FILE STRUCTURE

LCM.ACTIVE is structured as follows:

- Record 0 - Revision number of the format.
- Record 1 - Pointer to Range Table section of this file.
- Record 2 - Pointer to Active File section of this file.
- Range Table Section (begins at record number specified by Record 1) - A word-by-word representation of the Range Table to be copied to the real range table. Terminated with a -1.
- Active File Section (begins at the record number specified by Record 2). The first record pointed to by Record 2 indicates the size of Active Files. The following records list the port numbers whose Active Files are to be copied into the device. Terminated with a -1.



COMMENT SHEET

MANUAL TITLE LCM (LOTUS Cache Memory) Software Manual

PUBLICATION NO. SM-160-0025 REVISION A

FROM: NAME/COMPANY: _____

BUSINESS ADDRESS: _____

CITY/STATE/ZIP: _____

COMMENTS: Your evaluation of this manual will be appreciated by POINT 4 Data Corporation. Notation of any errors, suggested additions or deletions, or general comments may be made below. Please include page number references where appropriate.



BUSINESS REPLY MAIL

FIRST CLASS

PERMIT NO. 5755

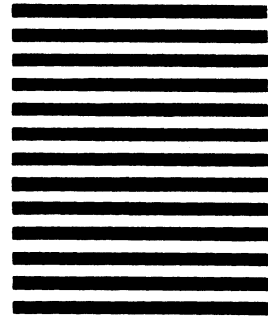
SANTA ANA, CA.

POSTAGE WILL BE PAID BY ADDRESSEE:

POINT 4 Data Corporation
PUBLICATIONS DEPARTMENT

2569 McCabe Way
Irvine, CA 92714

NO POSTAGE
NECESSARY
IF MAILED IN
UNITED STATES



CUT ON THIS LINE



POINT 4 DATA CORPORATION

2569 McCabe Way / Irvine, California 92714 / (714) 754-4114