LUND
PERFORMANCE SOLUTIONS

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

System Manager's Toolbox

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

System Manager's Toolbox version A.09

Whitney Olsen, Laura Bryngelson, Rodica Popa 03152002

TABLE OF CONTENTS

LIST OF FIGURES

1

SYSTEM MANAGER'S TOOLBOX

Welcome to System Manager's Toolbox

Welcome to the System Manager's Toolbox™ software package by Lund Performance Solutions. System Manager's Toolbox is the industry-standard performance monitoring and management application, devised to streamline, increase performance and help make day-to-day operations and repetitive tasks on the HP e3000 easier and more efficient.

This software consists of fifteen utilities that assist in all aspects of managing the HP e3000 environment: file management, system management, and performance management. All of the utilities that comprise the toolbox were designed by HP e3000 professionals with years of experience. Further, this toolbox was developed with the idea of improving existing MPE utilities and providing solutions that simply have not existed.

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 System Manager's Toolbox 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/iX operating environment.

Online Help System

In the online Help system, you will find explanations of the many features of System Manager's Toolbox, as well as tips to guide you through the program's basic functionality.

2

GETTING STARTED

If you have received an update tape, please install all files shipped in the LPSTOOLS account. During installation, several account-level UDCs are set so that each tool can be run by typing its name. The UDCs are operable by anyone using the MGR logon. If the UDCs are not used, then the user will need to issue a run statement for the tool. All of the tools in each toolbox run out of the LPSTOOLS account.

To familiarize yourself with the on-line edit facility and available function keys for each tool, refer to Appendix E, “Standard Function Keys” on page 301, and Appendix F, “The MODIFY Editor” on page 303. For information on the standard setting you would use for each tool, please see Appendix G, “Setting Options” on page 311.

Viewing Program Version Information

To find out which version of a Tool you are using without running the Tool, issue a RUN statement in the following form:

```
RUN toolname.PUB.LPSTOOLS, VERSION
```

To view the on-line help for a Tool without running the Tool, issue a RUN statement like the one above but replace the word "version" with the word "help" as in the following:

```
RUN toolname.PUB.LPSTOOLS, HELP
```

Conventions

When showing syntax for statement entry, what you type is indented, bold and uppercase (in most cases). Commands or computer statements that are included within the text are in double quotes and bolded or in uppercase.

In the example sections illustrating computer output, ellipsis (...) indicate that lines have been removed in cases where that particular output was judged to be superfluous.

Words in angle brackets (< >) denote user-specified inputs (usually a filename).

Words in square brackets ([ ]) denote optional parameters.

Organization of this Manual

This manual is divided into 15 chapters and 7 appendices. There is a chapter devoted to each tool, and each chapter is organized alphabetically within the toolbox.

Each chapter includes full information for the particular tool, including operations, syntax, commands, examples, and any background topics that may assist you in using the tools.

3

THE ACAP TOOL

The ACAP tool is used to view and alter the capabilities and attributes of both native mode and compatibility mode program files. ACAP also displays informational messages that help you identify program capability sets that seem unusual. For example, ACAP will warn you if a program's NMSTACK is set to zero.

Almost any program file attribute ca be altered or viewed with ACAP, including the status of the OCT flag for compatibility mode programs.

Operation

The most typical use for ACAP is adding a capability to a program file that was omitted during the LINK or PREP stage. This is easily accomplished in a single command line specification (or through an interactive dialogue sequence). Other typical uses include changing the initial value of the stack, heap or testing program operation base of capabilities.

Usually when ACAP is run it will open a program file with read/write access. However, if you don't have write-access to the file, you may choose to use ACAP's "PEEK" command and just display the program's current capability and attribute lists.

When ACAP is used in an interactive dialogue mode, all user changes are written to the program file as soon as the user closes the program file or exits the program. However, if during the course of changing a program file you decide to abandon your changes, you may enter the command "CAP=OLD", and all previously entered changes will be abandoned.

Capabilities

Program capabilities required include IA, BA, DS, MR, PH, PM.

Usage

ACAP can be started from the supplied UDC or from a fully-qualified RUN statement. Another option is to pass commands through the INFO string parameter. For MPE/iX users, progfile can be either a POSIX or MPE file reference.

- UDC

```
:ACAP [<progfile> [<commands>]]
```

- RUN

  ```
  :RUN ACAP.PUB.LPSTOOLS;INFO="[<progfile> [<commands>]]"
  ```

Command Summary

The following list provides a simple description of ACAP commands that you can use to quickly locate the command that suits the task at hand. Detailed information on each command is provided in the next section.

NOTE Portions of the command codes are printed in uppercase to denote the part of the command that ACAP requires in order to distinguish one command from another. However, the commands themselves are not case-sensitive.

Table 3.1 *ACAP Commands*

Command Code	Description
CAPability	Alters program capabilities
noCHECKS	[re]Sets parameter checking for programs
CLose	Closes program file
DLsize	Sets the CM DL size
DUMPWORTHY	Resets or sets the DUMPWORTHY bit for native mode programs
Exit	Terminates program
noFATAL	[re]Sets FATAL error bit for compatibility mode programs
HELP	Invokes ACAP help
Look	Synonym for Peek
MAXDATA	Sets the compatibility mode MAXDATA size
MAXPRIority	Sets the MAXPRIORITY field for native mode programs
nmHEAP	Sets the native mode Heap size (bytes)
nmSTACK	Sets the native mode stack size (bytes)
noNONFATAL	[re]Sets NONFATAL error bit for compatibility mode programs

Command Code	Description
OCTcomp	Turn on/off OCT'ed flag for compatibility mode programs
ODD	Displays unusual (odd) info about prog
Open	Opens a program [edits it]
Peek	Displays information about programfile
noPRIVSEGs	[re]Sets privileged segments bit for compatibility mode programs
PRIority	Sets the PRIORITY field for native mode programs
STACK	Sets the compatibility mode or native mode stack size
noZERODB	[re]Sets the ZERODB flag for compatibility mode programs

NOTE The leading "no" is optional. Commands may be abbreviated to the uppercase characters shown (i.e., LOOK or LOO or LO or L).

Command Definitions

Listed below are detailed descriptions of the ACAP commands.

CAPability

The CAPability command has the following syntax:

```
CAPability [+,-,=] <caplist>
```

This command is used to alter the capabilities of a program. It uses a very flexible syntax so capability modifications can easily be performed. The CAP command can be used to add (+), subtract (-), or set (=) capabilities to a program. For example, to add PM to a program, type "CAP + PM". Similarly, to remove PM from a program type: "CAP - PM".

<caplist>	Can be any of the following: NONE, ALL, OLD, IA, BA, DS, PH, MR, PM separated by commas, + or -. Two non-standard capabilities appear in the caplist above: ALL and OLD. The capability "ALL" tells ACAP to assign all possible capabilities to a program. The "OLD" capability tells ACAP to restore the program's caplist to its previous state. "NONE" specifies no capabilities. **NOTE** The OLD capability is only valid during a continuous session with a single program.
[+,-,=]	Specifying "+" and a capability adds that capability to the open program. "-" will subtract a capability. Specifying "=" will turn off all capabilities and then add the ones following the "=".

CLose

The CLOSE command closes the currently open program file. An implicit CLOSE is done by the EXIT and OPEN commands.

NOTE ACAP does not allow the renaming of the output filename, so you may want to make a backup copy beforehand. For example, you may want to have two versions of the same program, one with PM and one without. In this case you may want to name one "MYPROG" and the other "MYPROGPM" (hint—use REP in the System Manager's Toolbox to make a spare copy of MYPROG and name it MYPROGPM. Then use ACAP to add PM to MYPROGPM).

DLsize

This command has the following syntax:

```
DLsize = #halfwords | *
```

The DL (Data Limit) command changes the size of the DL size your program will have the next time it runs. A value of "*" means: restore the DLsize to the value it was when ACAP opened the program file.

This command is valid only for compatibility mode programs.

DUMPWORTHY

This command has the following syntax:

```
DUMPWORTHY ON | OFF
```

The DUMPWORTHY OFF command resets the DUMPWORTHY flag in the file system for your program file.

The DUMPWORTHY ON command sets the DUMPWORTHY flag in the file system for your program file.

In theory, any program file with the DUMPWORTHY flag set should be included in any memory dump generated by the system.

EXIT (or //)

The EXIT command is used to terminate ACAP. // is accepted as a synonym.

FATAL | NOFATAL

The two forms of this command are used to either set or reset the FATAL bit for compatibility mode programs.

nmHEAP

The HEAP command has the following syntax:

```
nmHEAP=#bytes | *
```

The HEAP command changes the size of the Native Mode Heap your program will have the next time it runs. A value of -1 means that the system default (set via SYSGEN) will be used. A value of "*" means that original value (at ACAP OPEN time) will be restored.

This command is only valid for native mode programs.

HELP or ?

This command has the following syntax:

```
HELP [commandname]
```

The HELP command with no parameters displays general help about ACAP. Help on a specific command is available by typing: HELP commandname.

Example: `HELP CAP`

LOOK

The LOOK command has the following syntax:

```
LOOK programfilename
```

This command is a synonym for **Peek**. The LOOK command opens the specified program file, reports the values of various attributes, and then closes the file.

MAXDATA

The MAXDATA command has the following syntax:

```
MAXDATA=#halfwords | *
```

The MAXDATA command changes the size of the MAXDATA your program will have the next time it runs. A value of -1 means that the system default (set via SYSGEN) will be used. A value of "*" means that original value (at ACAP OPEN time) will be restored.

This command is valid only for compatibility mode programs.

MAXPRIority

The MAXPRIority command has the following syntax:

```
MAXPRIority=# | *
```

This sets the MAXPRIORITY field of an NM Program file to the specified value (in the range 0...255).

A value of "*" means that original value (at ACAP OPEN time) will be restored.

The affect of this field is not yet known.

NONFATAL | NONONFATAL

The two forms of this command are used to either set or reset the NONFATAL bit for compatibility mode programs.

OCTcomp

The OCTcomp command has the following syntax:

```
OCTcomp <ON | OFF | # | OLD | *>
```

OCTcomp OFF tells ACAP to turn off the flag in the header of a CM program file that marks it as having been OCT'ed (i.e.: run through the OCTCOMP "compiler"). If the program had OCT data available, then ACAP will report it as being "CM (OCT'ed/disabled)". If the program had no OCT data, then it will continue to be reported as simply "CM".

OCTcomp ON tells ACAP to turn on the OCT'ed flag. ACAP will refuse to turn on the flag if no OCT information is available in the program file.

OCTcomp # forces the OCT flag word in a CM program file header to the specified value. This command should *not* be used unless you know exactly what you are doing.

OCTcomp OLD restores the OCT setting to it's original value for the current ACAP process. An asterisk ("*") can be used as a synonym for OLD.

ODD

The ODD command has the following syntax:

```
ODD [#] programfilename
```

The ODD command opens the specified program file, looks at the values of various attributes, and reports them in a somewhat fixed-format manner *only* if one or more of them are "odd" (or "unusual"). The file is then closed.

The primary purpose of ODD is to provide the ability to "scan" a group of program files (with multiple ODD commands, one per line), looking for unusual items (i.e.: NMSTACK or NMHEAP with default values).

If a number is specified before the program file name, then it is interpreted as a bitmask defining which unusual items you want to see. If the number is omitted, -1 (all bits on) is used.

Bit 15	1	show NMSTACK or NMHEAP values that are "default".
Bit 14	1	show program files with non-zero MAXPRIority or PRIORITY values.
Bit 13	1	show program files with the POSIX flag.

NOTE Default values for NMSTACK and/or NMHEAP are "bad" because the programmer is relying on the system's particular configuration to provide the NMSTACK and/or NMHEAP size. The system manager generally has *no* idea what that value should be ... it is the responsibility of the programmer to LINK their NM Programs with appropriate NMSTACK and NMHEAP values.

Open

The OPEN command has the following syntax:

```
Open programfilename [editcommands]
```

This command tells ACAP to open the specified NM or CM program file. If [editcommands] are present, they are executed. Editcommands may also be entered on subsequent input lines, until the program file is closed.

Example:

```
OPEN GRANT.PUB.LPSTOOLS
```

Example:

```
O GRANT.PUB.LPSTOOLS - BA
```

The above command opens GRANT.PUB.LPSTOOLS and deletes BA capability.

Peek

The Peek command has the following syntax:

```
Peek programfilename
```

Peek is used to display information about program file when ACAP has READ-only access to that **programfilename**.

PRIority

This command has the following syntax:

```
PRIority # | *
```

This set the PRIORITY field of an NM Program file to the specified value (in the range 0..255).

A value of "*" means that original value (at ACAP OPEN time) will be restored.

The affect of this field is not yet known.

PRIVSEGs | NOPRIVSEGs

The two forms of this command are used to either set or reset the PRIVSEG bit for the compatibility mode programs.

SET | RESET

SET/RESET are used to set/reset global options within ACAP.

Options are:

[NO]CHECKS

If CHECKS is set (default), then ACAP checks that the values of stack, heap, and DL are reasonable.

[NO]DEFaults

DEFaults, if set, causes some ACAP output to be generated in a a fixed-field manner, suppresses all of the capability flags except PM, and reports NM STACK and NM HEAP sizes as "DEFAULT" if they are -1 and 0, respectively. This option is of little use by itself, but can be useful when "LOOK"ing at a large number of program files. NODEFAULTS is set by default.

STACK

The STACK command has the following syntax:

```
STACK=#elements
```

The STACK command changes the size of the Stack your program will have the next time it runs. A value of -1 means that the system default (set via SYSGEN) will be used. The #elements are bytes for a NM program, halfwords for a CM program.

ZERODB | NOZERODB

This sets (or resets) the ZERODB bit in a CM program file. ZERODB tells the loader to initialize the primary DB area of a process to all binary zeros at the start of the process.

It isn't clear if NOZERODB has any affect on MPE/iX.

TOOLBOX STANDARDS

The ToolBox collections from Lund Performance Solutions have a uniform user interface. As a result, in addition to the commands specific to each Toolbox tool, most tools allow the commands described in “TOOLBOX STANDARDS” on page 311.

ACAP Examples

This section provides some examples of the various ACAP commands and their syntax. We have also includes examples of actual ACAP screens to help you see more clearly what your screen should look like.

The screen that follows shows how to add PM capability to a program (called `sosimage`) in a single command line specification.

```
Wolf:/SOSDEV/PUB: run acap.pub.lpstools;info='sosimage+pm'

ACAP [2.5] - LPS Toolbox [A.09f]            (c) 1995 Lund Performance Solutions

For Help at the ACAP prompt enter   ?
This product is licensed to: LPS Wolf
Opened: SOSIMAGE.PUB.SOSDEV
(NM)    old Cap = ba,ia,PM,ph; nmHeap = -1; nmStack = 10,000,000
        Auxiliary flags in SOM: DUMPWORTHY
(NM)    Cap = ba,ia,PM,ph; nmHeap = -1; nmStack = 10,000,000
        Auxiliary flags in SOM: DUMPWORTHY
Closed program file.

END OF PROGRAM
```

Figure 3.1 *Adding PM Capability*

The [<editcommands>] shown on the OPEN command is simply any combination of CAPability commands.

For example:

```
OPEN FOO CAP + PM
```

If ACAP is run with an INFO string, then it is interpreted as the one of the following two scenarios:

```
OPEN infostring
EXIT
```

or:

```
OPEN infostring
```

An EXIT command is implied if the `infostring` contains anything after the program name.

Thus, a UDC could be written that would give **PM** capability to a program with no user interaction:

```
givepm prog
run acap.pub.lpstools;info="!prog cap + pm"
************
```

The automatic EXIT command after the INFO string can be suppressed by running ACAp with PARM=1.

This can be summarized by the following examples:

`run acap;info="FOO"`	Opens FOO, stays alive
`run acap;info="FOO+PM"`	Opens FOO, adds PM, exits
`run acap;info="FOO+PM";parm=1`	Opens FOO, adds PM, stays alive
`run acap`	Stays alive

When ACAP runs it prompts with either "ACAP:" or "[programname] ACAP:" (if a program file has been opened).

NOTE MPE/iX 4.5 users have the option of using POSIX file references.

Figure 3.2 demonstrates the use of ACAP's **Peek** and **OCT** commands.

```
Wolf:/SOSDEV/PUB: run acap.pub.lpstools

ACAP [2.5] - LPS Toolbox [A.09f]            (c) 1995 Lund Performance Solutions

For Help at the ACAP prompt enter   ?
This product is licensed to: LPS Wolf

ACAP: peek /SYS/PUB/SPL
(CM)   Cap = ba,ia,ds,ph; Maxdata = 32,767; DL = 0; Stack = 3,500; OCT'ed

ACAP: open spl.pub.sys
Opened: SPL.PUB.SYS
(CM)   Cap = ba,ia,ds,ph; Maxdata = 32,767; DL = 0; Stack = 3,500; OCT'ed

[SPL.PUB.SYS] ACAP: oct off
Updated to:
(CM)   new Cap = ba,ia,ds,ph; Maxdata = 32,767; DL = 0; Stack = 3,500
      ; OCT'ed/disabled

[SPL.PUB.SYS] ACAP: close
Closed program file.

ACAP: exit
```

Figure 3.2 *Peek and OCT Commands*

In ACAP you have the ability to enter multiple commands on a single line. In this example, the PM and PH capabilities, and NM stack size are all altered in a single command.

```
Wolf:/SOSDEV/PUB: run acap.pub.lpstools

ACAP [2.5] - LPS Toolbox [A.09f]            (c) 1995 Lund Performance Solutions

For Help at the ACAP prompt enter   ?
This product is licensed to: LPS Wolf

ACAP: open sosimage
Opened: SOSIMAGE.PUB.SOSDEV
(NM)   Cap = ba,ia,PM,ph; nmHeap = -1; nmStack = 10,000,000
       Auxiliary flags in SOM: DUMPWORTHY

[SOSIMAGE.PUB.SOSDEV] ACAP: +pm+ph;stack=5000000
Updated to:
(NM)   new Cap = ba,ia,PM,ph; nmHeap = -1; nmStack = 5,000,000
       Auxiliary flags in SOM: DUMPWORTHY

[SOSIMAGE.PUB.SOSDEV] ACAP: close
Closed program file.

ACAP: exit
```

Figure 3.3 *Multiple Commands on a Single Line*

ACAP Error Messages

Each ACAP error message is described in the following table.

Table 3.2 *ACAP Error Messages*

Message	Cause	Action
DL value less than -1 may cause problems	User entered a value for DL that is less than -1.	Currently, values less than -1 have no known value. Since a value of -1 is used to designate the system default, it was decided to allow other negative values, in the event that HP decides to assign meaning to other negative values.
Error reading file system information	ACAP could not obtain file information about the program the user specified.	For an undetermined reason, the '`ffileinfo`' to the program file failed. Exit ACAP and check the condition of the program file. Make sure that it is a valid program, and that you have the capability to alter it.
Error reading record #0 of NMPRG file	ACAP could not obtain the program header information.	For an undetermined reason, the '`freaddir`' to the program file failed. Exit ACAP and check the condition of the program file. Make sure that it is a valid program, and that you have the capability to alter it.
Expected a capability	User entered "CAP=" without giving a new value for CAP.	The valid capabilities are: BA, IA, PM, MR, DS, PH, OLD, ALL, NONE. If the user wants to alter the capabilities one of the above must be selected. The capabilities cannot be abbreviated, although they are not case-sensitive.
Expected DL value	User entered "DL=" without giving a new value for DL.	In order to modify DL, the user must enter a command in the form of "DL=##" where ## is the new value for DL. A value of "-1" is used to designate the system's default value for this variable. **NOTE** DL can only be modified for compatibility mode programs, filecode=PROG.

Message	Cause	Action
Expected MAXDATA value	User entered "MAXDATA=" without giving a new value for MAXDATA.	In order to modify the compatibility mode maxdata that a program will use, the user must enter a command of the form "MAXDATA=#" where # is the new value for the compatibility mode MAXDATA. A value of "-1" is used to designate the system's default value for this variable. **NOTE** MAXDATA can only be modified for compatibility mode programs, filecode=PROG.
Expected nmHEAP value	User entered "HEAP=" without giving a new value.	In order to modify the native mode heap size that a program will use, the user must enter a command of the form "HEAP=##" where ## is the new value for the native mode HEAP. A value of "-1" is used to designate the system's default value for this variable. **NOTE** HEAP can only be modified for native mode programs, filecode=NMPRG.
Expected OCT FLAGS value	User entered "OCT" without an option specified.	Valid options for the "OCT" command are ON, OFF, #, OLD where # denotes a number.
Expected STACK value	User entered "STACK=" without giving a new value.	In order to modify the stack size (in either native mode or compatibility mode) that a program will use, the user must enter a command of the form "STACK=#" where # is the new value for the stack. A value of "-1" is used to designate the system's default value for this variable. **NOTE** STACK can be modified for both native mode programs and compatibility mode programs.
Expected: -, +, or = after CAP	Used an invalid capability operator.	Valid operators for the CAP command are: "-" (minus sign: used to remove capabilities) "+" (plus sign: used to add capabilities) "=" (equal sign: used to assign capabilities)

Message	Cause	Action
Failed to open	When ACAP tried to open the program specified, it encountered an error.	For an undetermined reason, the 'fopen' to the program file failed. Exit ACAP and check the condition of the program file. Make sure that it is a valid program, and that you have the capability to alter it.
Failed to open ASDF: Nonexistent permanent file	User specified a program file that ACAP was not able to open.	The user issued an "open" command with a program filename that ACAP could not open. The user should make sure that the filename specified was correct. The user should use the MPE command "LIST @,2" to examine the files in the current group.
Failed to post changes to program file:	When ACAP tried to post changes to the program file it failed.	For an undetermined reason, the 'fwritedir' to the program file failed. Exit ACAP and check the condition of the program file. Make sure that it is a valid program, and that you have the capability to alter it.
File is not a NMPRG or PROG file!	User tried to open a non-program file.	ACAP can only modify the attributes of native mode program or compatibility mode program files. Use the MPE command "LIST @,2" to review the files in your directory for 'NMPRG' or 'PROG' filecodes.
HEAP value less than -1 may cause problems.	User entered a value for the HEAP that is less than -1.	Currently, values less than -1 have no known value. Since a value of -1 is used to designate the system default, it was decided to allow other negative values in the event that HP decides to assign meaning to other negative values.
MAXDATA value less than -1 may cause problems.	User entered a value for MAXDATA that is less than -1.	Currently, values less than -1 have no known value. Since a value of -1 is used to designate the system default, it was decided to allow other negative values in the event that HP decides to assign meaning to other negative values.
MAXDATA must be in range -32768...32767:	User entered a value for MAXDATA that is less than 32768 or greater than 32767.	Input a valid value within the range specified.

Message	Cause	Action
Note: program is in use, so we are sharing it!	Selected program is being accessed elsewhere in the system.	This message, if followed by the message "ACAP cannot update this file", indicates that ACAP cannot write to the program file. If the second message does not appear, then ACAP should not have problems updating the selected file.
Oops...this program file has an unexpected header record	The structure of this program's header does not conform to known information.	ACAP can only modify programs that appear to contain valid program header information. Exit ACAP and check the condition of the program file. Make sure that it is a valid program, and that you have the capability to alter it.
Option not available for NMPRG	User tried to alter a NMPRG attribute with a PROG attribute.	Valid native mode attributes for modification are: NMHEAP, NMSTACK, and CAPs.
Program file not open!	User must open program first.	The user must use the "open" command to select a program file to alter before any of its attributes may be modified.
STACK value less than -1 may cause problems.	User has entered a value for STACK that is less than -1.	Currently, values less than -1 have no known value. Since a value of -1 is used to designate the system default, it was decided to allow other negative values in the event that HP decides to assign meaning to other negative values.
Unable to obtain write access to program.	ACAP could not obtain write access to the program file.	Exit ACAP and check the condition of the program file. Make sure that it is a valid program, and that you have the capability to alter it. Also, make sure that it is not being used by another user.
Unknown capability, expected one of:	User entered an invalid capability.	The valid capabilities are: BA, IA, PM, MR, DS, PH, OLD, ALL, NONE. If the user wants to alter the capabilities, one of the above must be selected. The capabilities cannot be abbreviated, although they are not case-sensitive.

Message	Cause	Action
Unknown edit option	User entered unknown command.	Input a valid ACAP command specified on the Command Summary page.

4

THE BETIMES TOOL

BETIMES is a utility program for changing MPE XL's current time or date.

Operation

There is often a need to change the current date and/or time that MPE XL maintains. Sometimes, a system is booted up with the incorrect date (or time). Sometimes, it would be convenient to change the date (or time) to test software that is sensitive to these values. MPE XL does not provide a mechanism to change the date or time. Thus, the need for BETIMES.

BETIMES allows the date and/or time to be changed via a simple user interface. If the new date (or time) is known, it can be entered directly with the DATE or TIME commands. If the offset is known (i.e.: the time is one hour too early), then the date (or time) can be changed by just that small amount (with the ADD and SUBTRACT commands).

When MPE XL boots, the START dialog fetches the date and time from a hardware clock. The hardware clock is supposed to have the correct GMT (Greenwich Mean Time) date and time. Additionally, the operating system maintains an offset which, when added to GMT, produces the local date and time. If the GMT time is incorrect, the CLKUTIL program should be used the next time the machine is at the "ISL>" prompt.

NOTE If the date (or time) is corrected with BETIMES, it will have to be re-entered correctly the next time MPE XL reboots.

Why doesn't BETIMES simply update the internal "offset from GMT", and thus eliminate the need to correct the clock when a reboot is finally done? Because the GMT offset is stored in an EEPROM in the computer. Each location in the EEPROM can be written to approximately 10,000 times before it burns out. BETIMES has no desire to be responsible for burning out an important piece of hardware! Lund cautions users to be very wary of running clock utility programs that purport to update the GMT offset!

Capabilities

BETIMES requires that the user have SM capability.

Usage

BETIMES can be run via the supplied UDC or a fully-qualified RUN statement.

- UDC

  ```
  BETIMES <command>
  ```

- RUN

  ```
  RUN BETIMES.PUB.LPSTOOLS;INFO="<command>"
  ```

If BETIMES is run with an INFO string, then it is interpreted as a single command, followed by an implicit EXIT command.

Thus, a UDC could be written that would set the date back by one year as follows:

```
BACKONEYEAR

RUN BETIMES.PUB.LPSTOOLS;INFO="SUBTRACT 1 YEAR"
```

To see just an overview of BETIMES abilities (and limits), type:

```
HELP OVERVIEW
```

Command Summary

Following is a summary list of BETIMES commands.

Table 4.1 *BETIMES Commands*

Command Code	Description
ADD	Adds to current date and time
DATE	Sets to specified value
EXIT	Terminates BETIMES
GLOBALS	Used for debugging BETIMES
GMTSECS	Reports the number of seconds the clock is offset from GMT time, and the number of seconds that have elapsed since the start of 1970-01-01
HELP	Invokes BETIMES help
NETCLOCK	Fetches the current time from a network time sever via a serial port, and then sets the local machine's clock to that value
NOW	Displays current date and time

Command Code	Description
RESET	Overrides the MPE/iX SETCLOCK command
SET	Used to set the SETCLOCK command
SUBTRACT	Subtracts from the current date and time
SYNCH	Synchronizes the hardware and software clocks
TICKS	Reports some internal time-oriented operating system values.
TIME	Synonym for DATE
TIMER	Tests the MPE/iX time routines, including the TIMER and PAUSE intrinsics
UPTIME	Reports when the system was booted up
//	Synonym for EXIT
?	Synonym for HELP
: <command>	A colon (:) followed by an MPE command or UDC name is passed to the HPCICOMMAND intrinsic

Command Definitions

This section describes BETIMES commands in detail.

ADD

This command has the following syntax:

```
ADD # <quantity> [# <quantity>...]
```

The ADD command adds to the current date and time.

<quantity>	[Years]	0 .. 99
	[MOnths]	0 .. 11
	[Days]	0 .. 31
	[Hours]	0 .. 23
	[MInutes]	0 .. 59
	[Seconds]	0 .. 59

#	a decimal number (with the above limits)

DATE

This command has the following syntax:

```
DATE [<yyyy/mm/dd> | <mm/dd/yy>] [hh:mm[:ss]]
```

The DATE command sets the date and/or time to the specified value. A date, a time, or both (in any order) may be entered.

Dates may be entered in International style (a four digit year, one or two digit month, and one or two digit day-of-month), or in American style (one or two digit month, one or two digit day-of-month, and 2 digit year).

Times must be entered in 24-hour format (i.e., military time). The number of seconds is optional and defaults to 0.

NOTE The TIME command is a synonym for DATE.

EXIT or "//"

The EXIT command is used to terminate BETIMES. // is accepted as a synonym.

GLOBALS

The GLOBALS command is used for debugging BETIMES. It reports global time-oriented data structures within MPE/iX.

GMTSECS

Reports the number of seconds the clock is offset from GMT time, and the number of seconds that have elapsed since the start of 1970-01-01. Note that MPE has two methods of to determining this value. "ticks since 1970" is the value in terms of the local time, and gmt_secs is the value in GMT time. Thus, if you are on Pacific Daylight Time, ticks since 1970 should be 7 hours ahead of gmt_secs.

HELP or ?

The HELP command with no parameters displays general help about BETIMES. Help on a specific command is available by typing:

```
HELP commandname
```

Example:

```
HELP ADD
```

NETCLOCK

The NETCLOCK command has the following syntax:

```
NETCLOCK [options]
```

BETIMES has the ability to fetch the current time from a network time sever via a serial port, and then set the local machine's clock to that value.

The options (if any) specified on the command line will override any related options from the BETIMES.CFG file (if any).

The options are described below.

BAUD [=] ####	Baud rate for the serial port. Default is 2400. Example: BAUD = 9600
END	Causes subsequent lines (if any) in the config file to be ignored.
FAKEINPUT	For internal testing. If specified, will not query the network clock for the current time. Instead, DATELINE is used to construct a fake network clock timestamp.
FAKEOUTPUT	For internal testing. If specified, will not set the machine's clock to the value fetched from the network clock server.
FORMAT #	Specifies which format the network clock provides the time in. Only format 1 is supported: `(...) ddmmmyy hh:mm:ss` Default: 1.
LDEV [=] ldevname_or_number	Specifies the ldev devicename or ldev# for the serial port connected to the network clock. This must be provided, there is no default value.

NOSETCLOCK	Tells BETIMES to use internal time change logic instead of SETCLOCK command to set the machine's time of day. Default: SETCLOCK (if on MPE/iX 5.0 or later)
NOTYPEAHEAD	NOTYPEAHEAD tells BETIMES not to enable type ahead for the netclock serial port. (TYPEAHEAD is on by default)
NOUPDATE	Tells BETIMES that you do NOT want to update the machine's clock to the time found on the network time server. This is useful for testing the BETIMES/NETCLOCK interface, without actually changing the local machine's time.
OUTPUT <Verbose | Quit>	OUTPUT QUIT tells BETIMES to abnormally terminate the BETIMES program if anything goes wrong with a NETCLOCK command. OUTPUT VERBOSE tells BETIMES to simply report any NETCLOCK error, skip the rest of the NETCLOCK command, and return to the command prompt. Default: Verbose
SETCLOCK <Now | Gradual>	Tells BETIMES to use the SETCLOCK command to change the machine's time of day. Either NOW or GRADUAL must be specified. Default: SETCLOCK NOW (if on MPE/iX 5.0 or later)
TRACE	For internal testing. Enables debugging output of the NETCLOCK command's progress.
TYPEAHEAD	TYPEAHEAD tells BETIMES to enable type ahead for the netclock serial port. (TYPEAHEAD is on by default)
UPDATE	UPDATE tells BETIMES that you want to update the machine's clock to the time found on the network time server. Default: UPDATE

BETIMES searches several locations for the configuration file, stopping as soon as it finds a file:

BETIMES.CFG

BETIMESP.CFG

BETIMES.CFG.LPSTOOLS

BETIMESP.CFG.LPSTOOLS

BETIMES.CFG.NUGGETS

BETIMESP.CFG.NUGGETS

Once the options from the command line and the configuration file have been parsed, BETIMES opens a serial port whose formal name is TIMETERM, and whose device name was specified via an LDEV (or PORT) option. The serial port is expected to be connected to a serial network clock.

Sample BETIMES.CFG file:

```
! betimes.cfg 96/01/11
! input is reasonably free form, one option per line
! Comments are indicated by "!" at start of line, or at the end
! of an option (but not in the middle!)
! In case of questions, contact Lund Performance Solutions at
! (541) 926-380 or Stan Sieler (650) 369-2303


LDEV = 255              ! ldev for netclock info
BAUD = 2400             ! baud rate for ldev
FORMAT = 1              ! netclock format (only 1 is supported)
SETCLOCK = NOW          ! NOW or GRADUAL
                        ! Note: if SETCLOCK omitted, uses BETIMES
                        ! inline code instead of SETCLOCK command
OUTPUT = VERBOSE        ! inline code instead of SETCLOCK command
END                     ! stop reading cfg file
```

NOW

This command simply displays the current date and time.

RE[SET]

This command has the following syntax:

```
[RE]SET options
```

Options:

```
[no]SETCLOCK
```

SUBTRACT

This command has the following syntax:

```
SUBTRACT <number> <quantity> [<number> <quantity> ...]
```

The SUBTRACT command subtracts from the current date and time.

<quantity>	[Years]	0 .. 99
	[MOnths]	0 .. 11
	[Days]	0 .. 31
	[Hours]	0 .. 23
	[MInutes]	0 .. 59
	[Seconds]	0 .. 59
<number>	a decimal number (with the above limits)	

SYNCH

This command has the following syntax:

```
SYNCH <HARDware | SOFTware>
```

The SYNCH command tells BETIMES to synchronize the hardware and software clocks, something that MPE/iX normally does every now and then (i.e., at unknown intervals!).

HARDware	Changes the hardware clock to match the software clock.
SOFTware	Changes the software clock to match the hardware clock and is **NOT** recommended.

In general, the use of SYNCH is not recommended, because MPE does a good job of synchronizing the clocks itself.

TICKS

The TICKS command reports some internal time-oriented operating system values.

TIMER

The TIMER command tests the MPE/iX time routines, including the TIMER and PAUSE intrinsics.

NOTE It isn't unusual for the TIMER command to notice that the PAUSE intrinsic is pausing for slightly less time than it should on values like 0.1 seconds. Moral: don't rely on PAUSE to successfully pause for times less than a second or so.

UPTIME

The UPTIME command reports when the system was booted up.

TOOLBOX STANDARDS

The ToolBox collections from Lund Performance Solutions have a uniform user interface. As a result, in addition to the commands specific to each Toolbox tool, most tools allow the commands described in “TOOLBOX STANDARDS” on page 311.

BETIMES Examples

Examples 1 uses the ADD command. To set the clock ahead by one hour, at the BETIMES prompt enter:

```
ADD 1 HOUR
```

To set the clock ahead by one year, two months, and three days, at the BETIMES prompt enter:

```
ADD 1 Y 2 MO 3 D
```

Or:

```
add 1 year, 2 months, 3 days
```

Example 2 uses the SUBTRACT command. To set the clock back by one year, two months and three days, at the BETIMES prompt enter:

```
SUB 1 Y 2 MO 3 D
```

To set the clock back by one year, at the BETIMES prompt enter:

```
SUB 1 year
```

Example 3 uses the GMTSECS command. At the BETIMES prompt enter:

```
gmtsecs
ticks since 1970 -> 864752789 secs  [...]
gmt_secs ()       = 864777989 secs ($338b7705)
local - gmt =          -25200 secs (-7.0 hours)
```

Example 4 uses the TICKS command. At the BETIMES prompt enter:

```
ticks
Ticks:
    since 1970    $c49f47aeabff40 (864753525895 millisecs)
    global_timer  $c49f47ad7c5270 (864753525584 millisecs)
    cpu_0         $c49f47aeabff47
       + CR16     $c49f481f24608f
       +last_cr16 $c49f481f246332
```

BETIMES Error Messages

Each BETIMES error message is described in the following table.

Table 4.2 *BETIMES Error Messages*

Message	Cause	Action
OOPS: TIMER intrinsic no longer works!	BETIMES is not compatible with the version of /iX (or XL) on which it is running.	As HP releases new operating system version, BETIMES needs adjustment to keep in sync with changes that HP implements. To correct this state, warm boot (TC) your machine.
Unable to locate TIMER globals.	Probably due to changes HP made to the operating system between major updates (i.e., patches).	Call Lund Performance Solutions Technical Support. Please provide the Build-id for your operating system, machine series, and BETIMES version when you call.

Message	Cause	Action
You need SM or OP capability to run this program.	User needs SM or OP. Since BETIMES should really only be used by the system manager, it checks to make sure that the user has SM or OP capability.	Have the system manager change the time, or run the GRANT ***System Manager's Toolbox*** to give yourself SM capability.

5

THE BLAZE TOOL

BLAZE is a file management tool that uses a terminal-based windowing technology, WINGSPAN. BLAZE supports a very powerful fileset specification syntax which simplifies file management operations like copying and purging. As you become comfortable with BLAZE, you will want to explore advanced topics like file tagging, mass operations, and file subset management.

Operation

BLAZE is easier to use if you take a few minutes to become familiar with the windows, fileset specification syntax, single letter comma keys, and function key operations. You may have covered some of these topics in the appendices. Basic operations like cursor key support and function key descriptions are also explained in the appendices.

BLAZE Typeahead status is set by the “terminal” option in the **Settings** pull-down menu. With Typeahead enabled, BLAZE single letter command keys require only a single keystroke. With Typeahead disabled, BLAZE single-letter command keys require two keystrokes for the key to be executed. Single-letter commands are discussed in detail later. By default, BLAZE Typeahead is disabled.

Capabilities

Program capabilities required include IA, DS, and PH. No special user capabilities are required to run BLAZE.

BLAZE Screen Layout

The basic BLAZE window contains four sections of interest: the *status line*, *work area*, *single-letter command keys* and *function keys*.

The status line is located at the top of the screen on row 1. Operational status messages are displayed here. The row beneath the status line, row 2, is where the menu bar is located. The menu bar is used to make top-level choices.

The work area is the area in the middle of the screen between the status line and the function keys. Depending on your application you may have up to five windows on the screen.

The single-letter command keys are used to perform operations like file tagging and fileset copying. The object of a single-letter command's operation is determined by which window is active. For example, if the Account Structure window option is active and you issue a TAG (T) command, then all the files associated with the line you are on will be tagged. However, if the File Content window is active when the TAG command is issued, then only the file that is currently selected will be tagged.

The function keys are located at the bottom of the screen. There are eight function keys. Some keys have a standard use assigned to them, while other keys are assigned functionality that is specific to a given operation on an as-needed basis.

The next two pages introduce you to several of the more commonly used screens in BLAZE. The major focus for the next two pages is on understanding what components can be identified on each screen. Information on each component is discussed in later sections.

BLAZE Menu Structure

Following is a detailed discussion of the various menu screens found in the BLAZE tool program.

The Main Menu

This screen lays out the basic structure of the BLAZE screen.

NOTE The status line (row 1); the shaded menu bar (row 2); and the function key locations. The menu bar selections that end in 2 dots (..) indicate that they have associated pull-down menus.

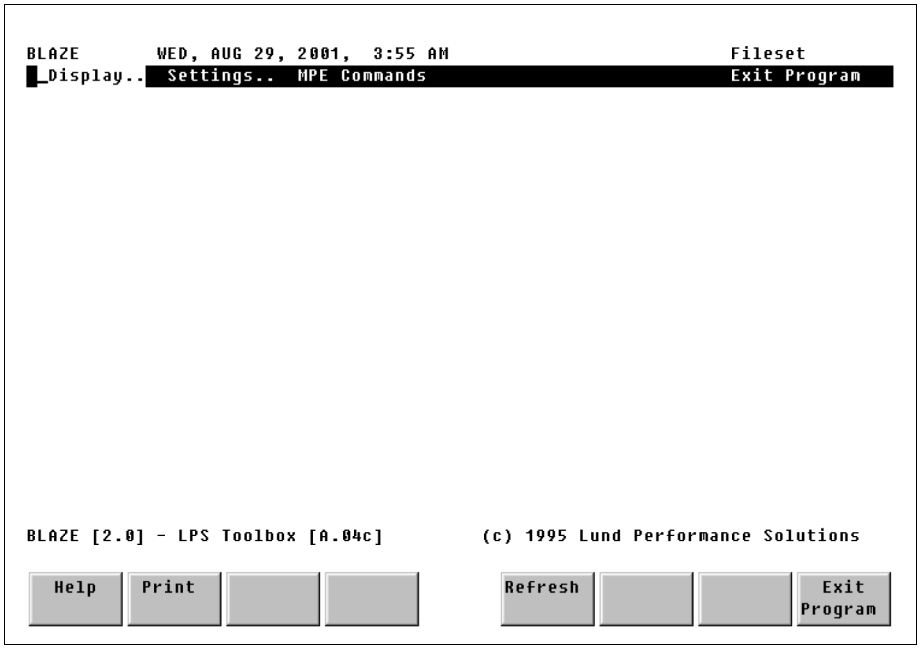

Figure 5.1 *Main Menu*

The Display Menu

The Display menu is the gateway to BALZE's file management windows (Tree and View). Additionally, BLAZE's File Compare (Compare) and Status Report (Profile) windows are accessible through the Display menu.

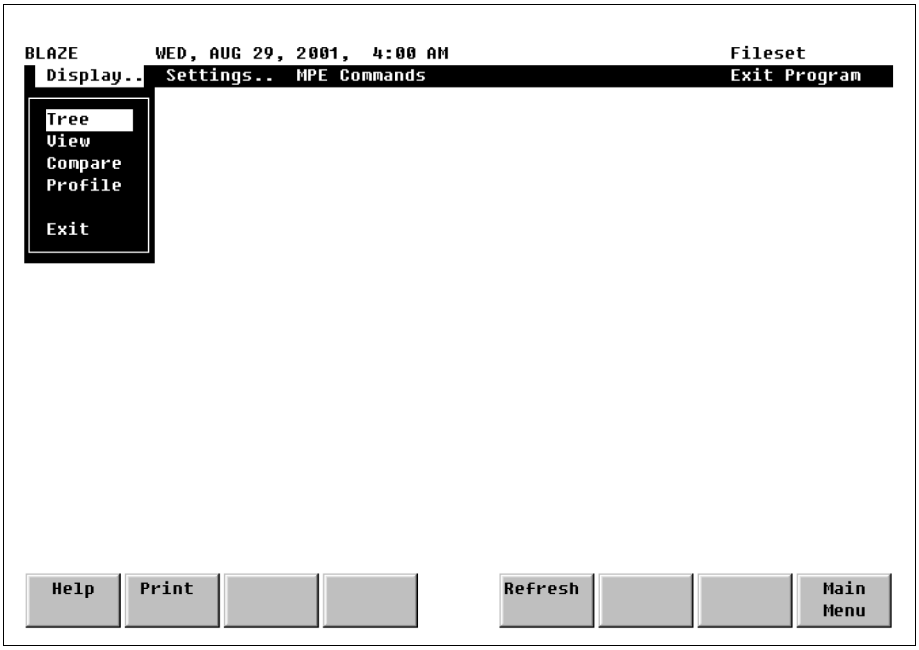

Figure 5.2 *Display Menu*

The Settings Menu

The Settings menu is for user-customizing of the interface as well as fileset specification. Items configured in this section can be saved to a configuration file. The default configuration filename is BLAZECFG, which (if present in the logon group) is loaded automatically. BLAZECFG can be equated to another file.

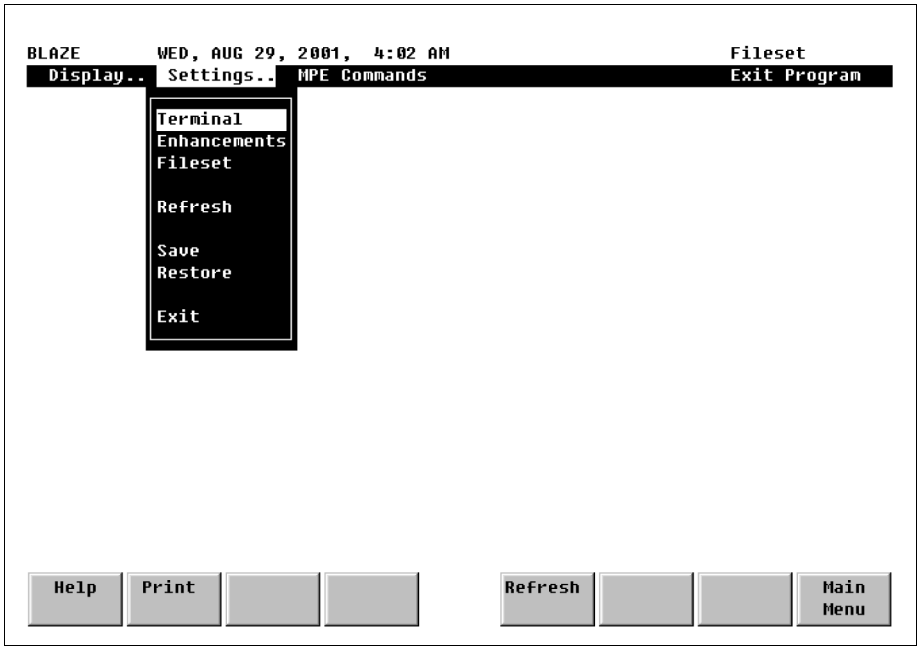

Figure 5.3 *Settings Menu*

MPE Commands/Exit Program Options

These two menu bar selections have no associated pull-down menus. The MPE Commands selection displays a small pop-up window where MPE commands or UDCs may be entered.

The Exit selection terminates BLAZE execution.

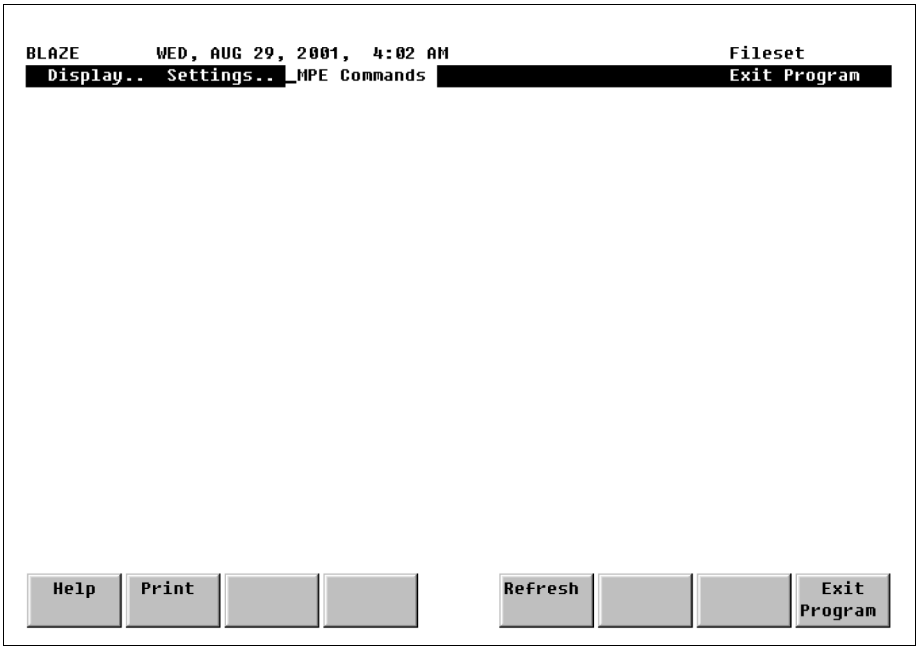

Figure 5.4 *MPE Commands/Exit*

Display Selection Menu

The following screens are accessed through the Display menu.

The Tree Screen

The information on this screen is divided into two panels. The left half of the screen displays account and group information. The right half of the screen displays file lists in a format similar to that of LISTF. File management actions are achieved through several single-letter command keys and the function keys.

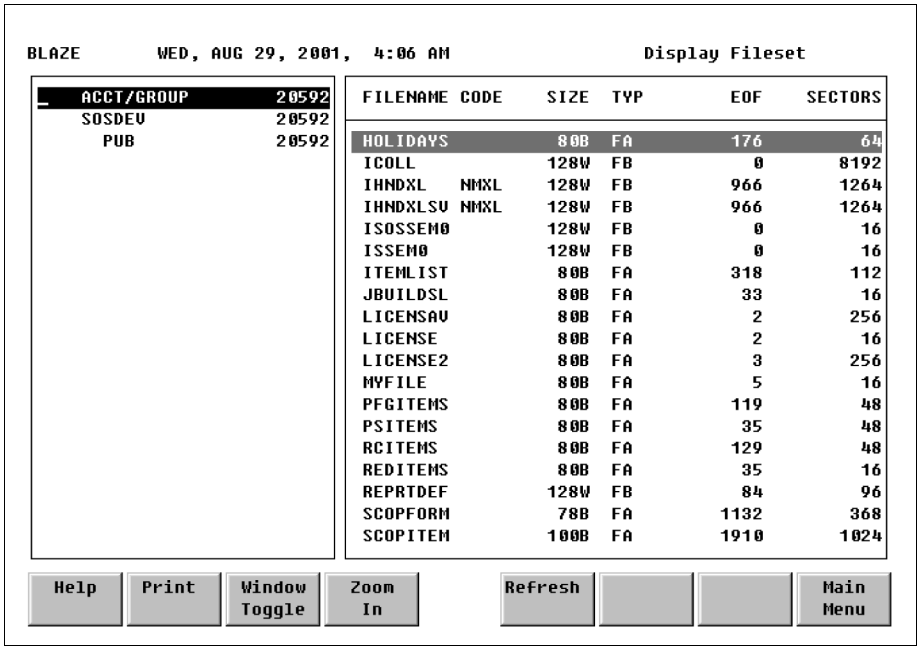

Figure 5.5 *Tree Screen*

The View Screen

This screen is divided into three information regions. The window to the left displays account and group information. The middle window displays a filename list. The window to the right displays file contents. As with the Tree display, file management is handled through single-letter command keys and function key selections.

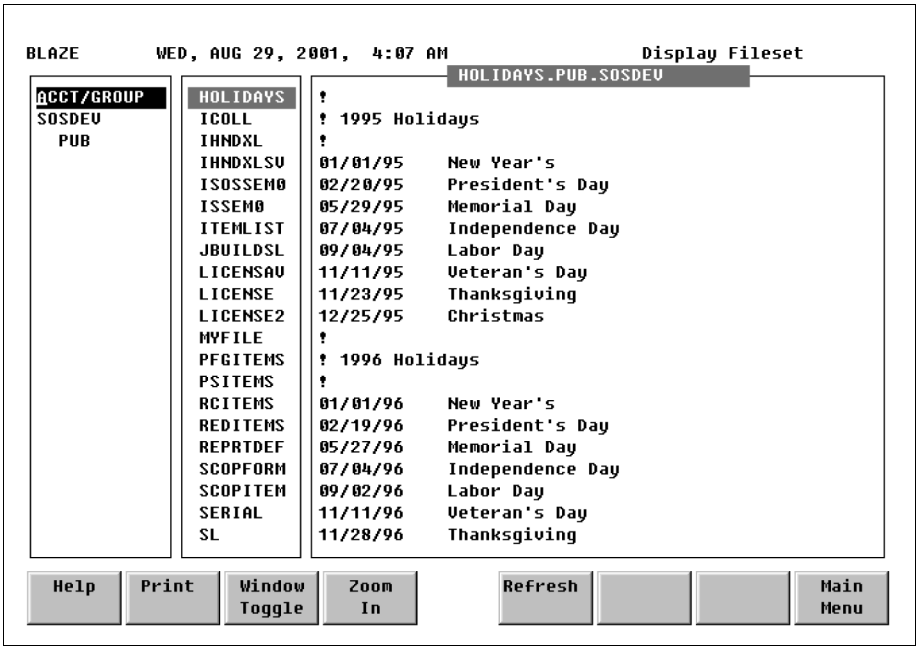

Figure 5.6 *View Screen*

The Compare Screen

This screen displays two windows for viewing the contents of two different files. The function keys provide control over which window is active and in the format of the display. The windows may be scrolled separately or together. ASCII and hexadecimal display formats are available.

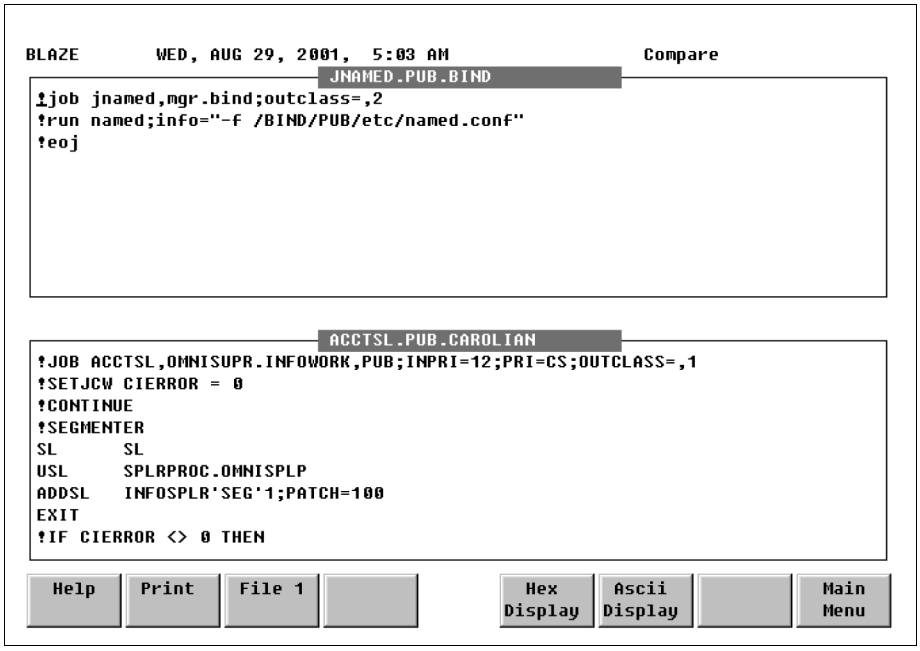

Figure 5.7 *Compare Screen*

The Profile Screen

This screen is divided into three windows. The information provided here includes system information, fileset statistics, and fileset specifications. System information provides information about the environment in which BLAZE operates. Fileset statistics show user-defined filesets. The fileset specifications window displays the current fileset.

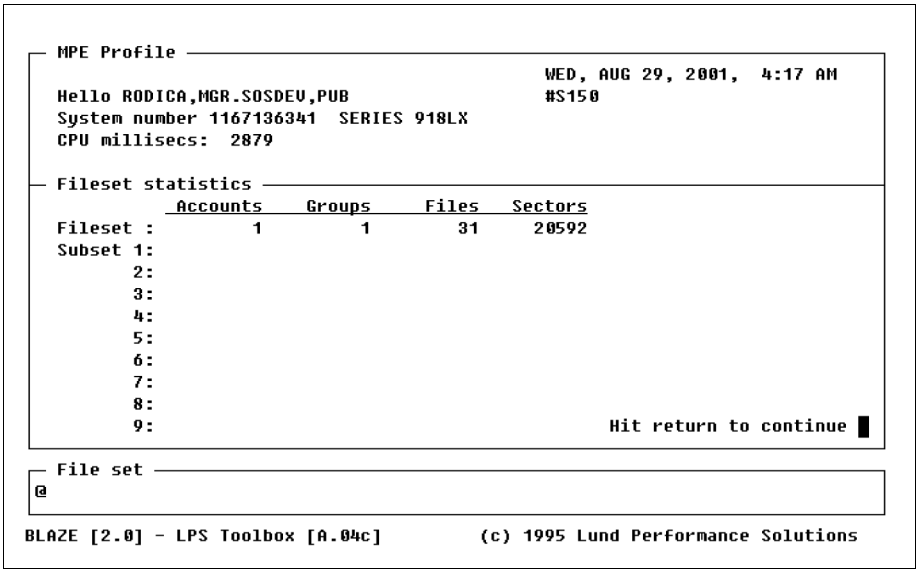

Figure 5.8 *Profile Screen*

Fileset Specification

BLAZE supports a LISTF-style file specification syntax with powerful extensions for creating versatile file descriptions. Filesets can be added or subtracted, and particular characteristics can be used to qualify each fileset.

```
<fileset>
        = <fileset definition>  [ + <fileset definition>] [...]
                                [ - <fileset definition>] [...]
<fileset definition>
        = <MPE fileset> [,<filter>]
<filter>
        = ,<filter descriptor> [,<filter descriptor>] [...]
<filter descriptor>
        = [ "CODE"    <relop> <numeric value> ]
          [ "LABELS"  <relop> <numeric value> ]
          [ "LIMIT"   <relop> <numeric value> ]
          [ "EOF"     <relop> <numeric value> ]
          [ "SECTORS" <relop> <numeric value> ]
          [ "BF"      <relop> <numeric value> ]
          [ "CODE"    <relop> <mnemonic> ]
          [ "CREDATE" <relop> <date  ]
          [ "MODDATE" <relop> <date> ]
          [ "ACCDATE" <relop> <date> ]
          [ "CCTL"    <onoroff> ]
          [ "RIO"     <onoroff> ]
          [ "MSG"     <onoroff> ]
          [ "CIR"     <onoroff> ]
          [ "UNDEFINED" ]
          [ "VARIABLE" ]
          [ "BINARY"   ]
          [ "ASCII"    ]
          [ "FIXED"    ]
          [ "REC"      ]
          [ "TEMP"     ]
Miscellaneous Parameters:
<relop> = "=" | "<" | ">" | "<=" | ">=" | "<>"
<numeric value> = <decimal digits>
<date> = yymmdd | "TODAY"
<onoroff> = [ "ON"  ]  "=ON"
            [ "OFF" ]  "=OFF"
```

Figure 5.9 *Fileset Specification Diagram*

BLAZE Parameters

One of the most powerful features of BLAZE is its fileset specification syntax. The syntax diagram in Figure 5.9 outlines all valid fileset descriptions. The syntax that BLAZE supports is based on the MPE LISTF fileset description. Wildcards are supported and multiple fileset descriptions can be logically connected with the puls (+) and minus (-) operators.

There are several possible options for reducing a large fileset into a more specific fileset. This is accomplished using the **filter descriptor**. At this time, there are 21 different filters that can be applied to any fileset.

The syntax for applying filters is:

```
<fileset>,<filter>
```

When multiple filters are applied to the same fileset, the effect is that of a logical "and"

```
@,code=nmprg+@,code=prog
```

In English, this reads: "For all files in this group select the files with the filecode `nmprg` and files with the filecode `prog`."

DATE Filter Definitions

There are three different types of date filters: ACCDATE, CREDATE, and MODDATE. Dates can be specified in two different formats, "yymmdd" and "yy/mm/dd". Also the literal "TODAY" can be used to specify the current date. The relational operators equal (=), less than (<), greater than (>), greater than or equal to (>=), less than or equal to (<=), and not equal to (< >) can be used to create the exact date filter that is required.

ACCDATE

This definition represents "Access Date". It reports the time that this file was last accessed. For example, list all native mode programs that were used today:

```
@.@.@,code=nmprg,accdate=today
```

This example also uses the filter code.

CREDATE

This definition represents "Creation Date". It is the date that a file was created. For example, list all files in this account created after January 15, 1994:

```
@.@,credate>940115
```

MODDATE

This definition represents "Modification Date". It is the date of the last modification that was made to a file. For example, list all files that were modified today:

```
@.@.@,moddate=today
```

NUMERIC Filter Definitions

The filters in the next section accept numeric data as input. The relationship between the filter and numeric data is defined by the relational operator you select. A range can be defined by using the same filter twice, once with an upper limit and again with a lower limit.

BF

This definition represents Blocking Factor. Use this filter to specify a blocking factor size. For example, list all files in this account that have a blocking factor of 16:

```
@.@,bf=16
```

CODE

This definition represents "Filecode". It is the MPE file subsystem filecode. The MPE file subsystem assigns filecodes to all disk files. The filecode is a 16-bit signed number. Negative numbers indicate privileged filecodes.

Many filecodes have predefined meanings. For example, the filecode number 1029 is defined (by MPE) to be used for compatibility mode (CM) program files. System-defined filecodes usually have associated mnemonics. In the case of a CM program, MPE displays the 4-character mnemonic "PROG" when the filecode number is 1029. There are dozens of predefined filecodes. Consult the *MPE Commands Reference Manual* for a complete listing. In addition to system-defined filecodes, there are many others that are commonly used. For example, filecode number 711 indicates a "squished" file, meaning that the file has been compressed via the popular Boeing Computer Services' file compression utility called SQUISHER. Filecode number 111 indicates a QEDIT (a product of Robelle Consulting, Ltd) text file.

When specifying a filecode for the CODE filter, either the numeric value can be used or the mnemonic string. For example, list all files in this group with the filecode equal to 1029:

```
@,code=1029
```

This is equivalent to "@,code=prog"

List all of the native mode executable libraries on the system:

```
@.@.@,code=nmxl
```

EOF

This definition represents End Of File location. this filter lets you specify the size of files to select by specifying an EOF size. For example, list all files in this account that have an EOF equal to 0, and a sector count > 0:

```
@.@,eof=0,sectors>0
```

LABELS

This definition represents "User Labels". This filter lets you limit file selection to just those files having the specified number of user labels. For example, list all files in this account that have user labels:

```
@.@,labels>0
```

LIMIT

This definition represents "File Size Limit". It is the maximum number or records allowed in the file. For example, list all files in the current group except native mode program files, that have a record limit greater than 10000:

```
@,limit>10000,code<>nmprg
```

REC

This definition represents the record size of a file. Use this filter to select files based on record size. For example, list all files in the current group that have a record size equal to 80 bytes:

```
@,rec=80
```

SECTORS

This definition represents the sector size of the file. Use this filter to specify the size of files for selection. Use two SECTORS filters to specify a range. For example, list all files in the current group that have more than 1000 sectors allocated to them:

```
@,sectors>1000
```

List all files in the current group that have more than 1000 sectors but less than 3000 sectors allocated to them:

```
@,sectors>1000,sectors<3000
```

TEMP

This definition represents TEMP files only. Use this filter to specify temporary files only. For example, list all temp files in the current account:

```
@.@,temp
```

NON-PARAMETER Filter Definitions

The following filters have no parameters; you simply include the filter name to select this filter.

ASCII

This definition represents ASCII files only. Limit file selection to ASCII files only. For example, list all ASCII files in this account that are empty.

```
@.@,ascii,eof=0
```

BINARY

This definition represents Binary files only. Limit file selection to binary files only. For example, list all binary files in this account that are not program files:

```
@.@,binary,code<>nmprg,code<>prog
```

FIXED

This definition represents Fixed record length files only. Limit file selection to fixed record length files. For example, list all fixed record files in this account.

```
@.@,fixed
```

UNDEFINED

This definition represents Undefined record length files only. Limit file selection to files whose record length is undefined. For example, list all undefined record length files in this account.

```
@.@,undefined
```

VARIABLE

This definition represents Variable record length files only. Limit file selection to variable record length files. For example, list all variable length files in this account.

```
@.@,variable
```

SINGLE PARAMETER Filter Definitions

The filters in this section only have one parameter, which must be included. It can either be "=ON", or "=OFF".

CCTL

This definition represents "Carriage Control". This filter lets you specify whether to look for files that were/were not written with carriage control. For example, list all fixed record length ASCII files, that were created without the carriage control characters in the current group:

```
@,fixed,ascii,cctl=off
```

CIR

This definition represents Include Circular files. This filter lets you specify whether or not to include CIR files. For example, list all circular files from the current group.

```
@.@,cir=on
```

MSG

This definition represents Include Message files. This filter lets you specify whether or not to include MSG file. For example, list all in this group except message files, and native mode relocatable libraries:

```
@,msg=off,code<>nmrl
```

RIO

This definition represents Include Message files. This filter lets you specify whether or not to include RIO file. For example, include all relative I/O files from the current group:

```
@,rio=on
```

BLAZE Commands

In addition to function keys and menu selections, BLAZE provides single-letter command (SLC) keys that are used to pop-up single-function windows. At this time there are 13 different single-letter commands. As with all BLAZE command entries, the SLCs are not case-sensitive.

SLCs are available when the BLAZE Tree or View screens are active. At other times, the function keys are used to specify selections.

BLAZE Objects

Most SLCs perform a given operation on an object. The object of the command varies, depending on which BLAZE window is active, where the cursor is located, and whether or not any file subsets are defined.

For instance, if the Account Structure window is active, the object of the SLC will be a fileset, an account, or a group. If the File List window is active, the object of the SLC will be the file specified by the cursor's position. In other words, the file name that is highlighted by the cursor is implicitly selected whenever you invoke an SLC.

SLC Key Summary

Many of the SLC keys fall into logical groupings. In the summary that follows, commands are defined according to the type of operation that is invoked.

Table 5.1 *SLC Key Summary*

Task	Letter	Description
Defining Filesets	F	Fileset (define a new fileset)
	M	MAGNET (select fileset based on contents of file)
Choosing Files	T	Tag files
	U	Untag files
File Subset Management	S	Subset (create a new file subset)
	X	eXpand (activate the previous file subset)
	N	Next subset (activate the next file subset)

Task	Letter	Description
BLAZE Object Management Commands	C	Copy files
	P	Purge files
	R	Rename files
	E	Execute MPE command
	Z	Crunch file (Zap) (release wasted disk space)
File Finding Commands	/	Set up find parameters
	>	Find next
	<	Find previous
Help: the BLAZE Single Letter Command Key Summary	H	Help (pop-up command summary)

Defining Filesets

This section discusses various ways of specifying filesets.

F

The F command is used to specify a fileset. Using this SLC will cause a small single-line window to pop-up on top of the current window (see Figure 5.10). In this window you can define a new file specification using the syntax described earlier in the "File Specification Syntax" section. The maximum length of a fileset description is 78 characters. If necessary, use the cursor keys to edit the text.

NOTE Don’t forget, BLAZE will use your input exactly as it appears on the screen.

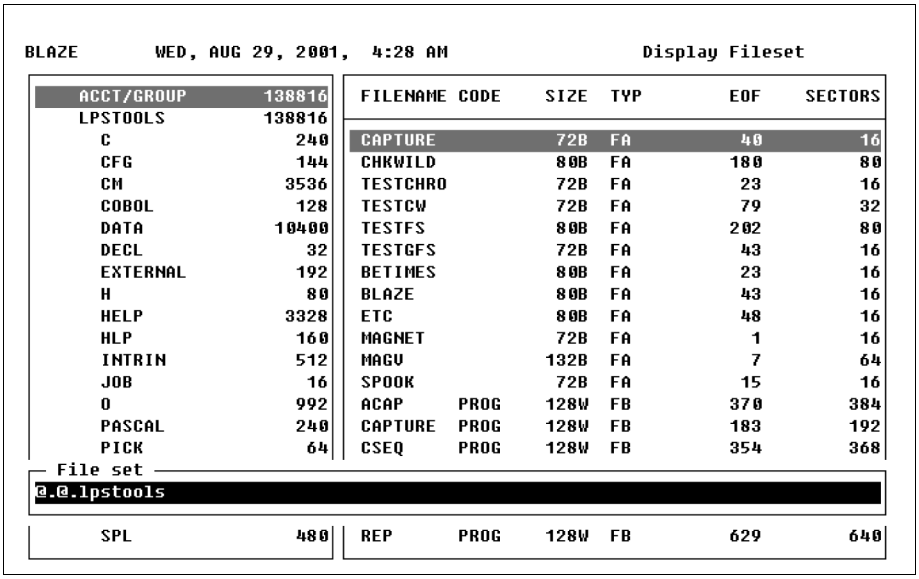

Figure 5.10 *Specify Fileset*

M

The M command pops up a window on top of the current display (see Figure 5.11). This window is titled "Words to search". The parameters entered here are passed programmatically to MAGNET in the ***System Manager's Toolbox*** for file searches.

Inside this window are three search characteristic questions to answer. Each has a default that is initially displayed. User definable search characteristics are:

Case sensitive: Enter "**Y**" for a case-sensitive search; enter "**N**" for a case-insensitive search.

All words must occur: If you select "**Y**" for this entry, then all of the words you specify must be found in a file to be considered a match. By specifying "**N**" if any word from the list is found in any file, that file is included in the fileset.

Whole words: Enter "**Y**" if words must match exactly. Enter "**N**" if the word can be part of another word.

Next, you can specify up to eight text strings (words). The total length of the eight text strings is limited to approximately 180 characters.

After entering all information press F6 (Accept) key to start the search. If the Account Structure window is active, then the object of this command will be determined by the location of the cursor

(i.e., a fileset, account, or group will be searched). If the File List window is active, then the object of the search will be a single file, as specified by the cursor.

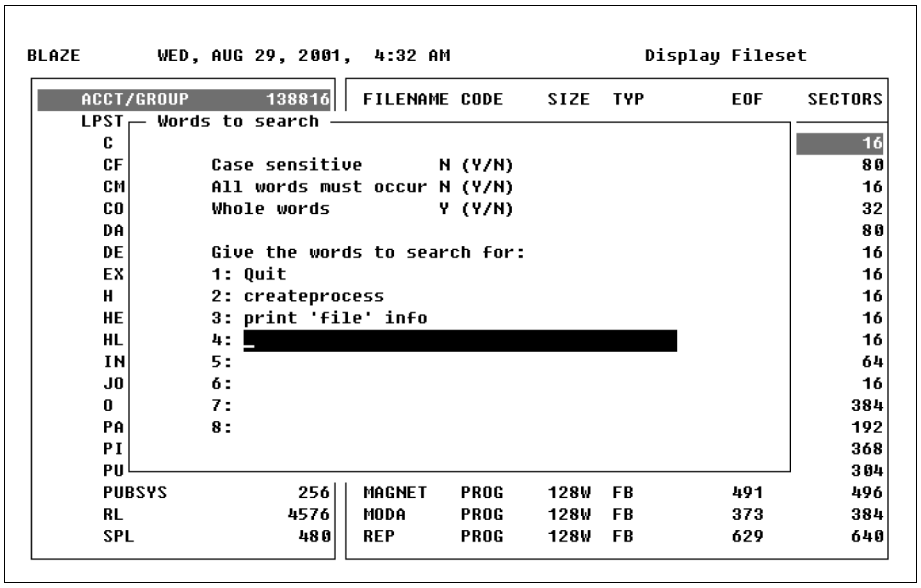

Figure 5.11 *Word Search*

Figure 5.11 illustrates how you can locate the files that use a few common intrinsics. Something like this might be handy when migration to a Spectrum machine and you need an easy way to locate the intrinsic calls requiring modification for compiling in native mode.

Choosing Files

This section describes the two SLCs that are used for selecting files: the **Tag** and **Untag** commands.

T

Selecting this command will result in files being tagged. Determining which files are tagged is easy since BLAZE marks each tagged file with the character curly right bracket (}). As with most SLCs, the object of the Tag command is determined by the active window. Entire file subsets, accounts and groups are tagged while the Account Structure window is active. Individual files are tagged when the File List window is active.

When the Account Structure window is active the following mass tagging is possible.

Warning: Tagging entire file subsets can take several minutes if a large number (i.e, thousands) of files are specified.

Fileset Tagging

To tag all of the files in an entire file subset, move the cursor to the top line of the window. The text that will be highlighted will read "ACCT/GROUP". All of the files in all of the accounts and the groups displayed in this window will be tagged.

Press the letter "**T**" to initiate tagging.

Account Tagging

To tag all of the files in one account, use the arrow keys to move the cursor (highlight bar) to the name of the account that you want to tag.

Press the letter "**T**" to initiate tagging.

Group Tagging

To tag all of the files in a group, use the arrow keys to move the highlight bar to the name of the group that you want to tag.

Press the letter "**T**" to initiate tagging.

File Tagging

When the File List window is active file tagging is possible. Individual file tagging is accomplished by highlighting the file which you want to tag and then pressing the letter "**T**". The tag indicator (}) will be displayed just to the left of the filename to indicate that it has been tagged.

U

Selecting this command will result in tagged files being untagged. This command is used to undo Tag command actions. Functionally, it performs the opposite operation of the tag command. Untagging a file that was not previously tagged is meaningless and has no effect.

In terms of mass untagging operations the U tag command functions the same way that the Tag command functions. Please refer to the Tag command for details on untagging various levels of file subsets.

Untag acts like an "except" operator when it is combined with the Tag command for mass operations. Consider the situation where you want to tag all of files in an account except one or two files. The easiest way to accomplish this would be to “tag” the entire account and then switch to the File List window (use the F3 Window toggle key) to untag the files you want to exclude.

File Subset Management

This section describes the three commands that are used for file subset management.

A file subset is a group of files from the previously defined file subset. Initially, the only file subset defined is the fileset that was specified with the FILESET option of the "Settings.." submenu. To define a file subset, the T and U commands are used to select files which you want to place in a

subset. in addition to these two commands, the M (MAGNET) command can be used to select files for tagging. As file subsets are defined, they become smaller and more focused on a particular characteristic. BLAZE allows up to nine file subsets to be defined.

S

The Subset command is used to create a new subset. The subset is created from the files that have been tagged. When the subset command S is issued, BLAZE assigns a subset number to the newly created file subset and updates the display screen with the new subset information. The subset command performs the same action regardless of which window is active.

Highlighting the top line (i.e., the ACCT/GROUP label) of the Account Structure window and pressing F4 (ZOOM IN) pops up the Profile screen. The Profile screen displays environment information about the BLAZE session, including a list of the currently-defined file subsets.

The initial state for all files in the new subset is "Untagged".

X

The X (expand) SLC switches from the active file subset to the previous file subset definition. Repeated use of X SLC results in restoring all previous file subsets until the original fileset is active again. In effect, X sequentially navigates through each of the file subsets, beginning with the last subset created and ending at the original fileset.

NOTE The status line will always indicate the number of the active file subset.

N

The N SLC is used to select the next file subset to become active. This command is only useful if the X SLC has been used. It navigates in forward sequence, ending at the last file subset created. Repeated use of the N SLC results in the highest numbered file subset being selected as the active file subset.

NOTE The status line displays the number of the active file subset.

BLAZE Object Management Commands

This section describes the five commands that work on BLAZE objects. Remember that the object of these commands will be determined by the active window and/or location of the cursor.

When the **Account Structure** window is active and the cursor is located on the top row of the window, selecting one of these commands will result in the entire fileset being processed. When

the cursor is located on an account name, this command processes the entire account. When the cursor is located in a group name, this command processes the entire group. When the **File List** window is active then file which is highlighted by the cursor will be processed, even if it is not tagged.

C

This SLC is used to copy file subsets. Invoking the C SLC results in a pop-up window being displayed on top of the current window (see Figure 5.12). This window is titled "Copy Files", and contains two copy-related questions. Each question displays an initial default. If the default settings are not satisfactory you can edit them.

Press F6 (Accept) to initiate file copying.

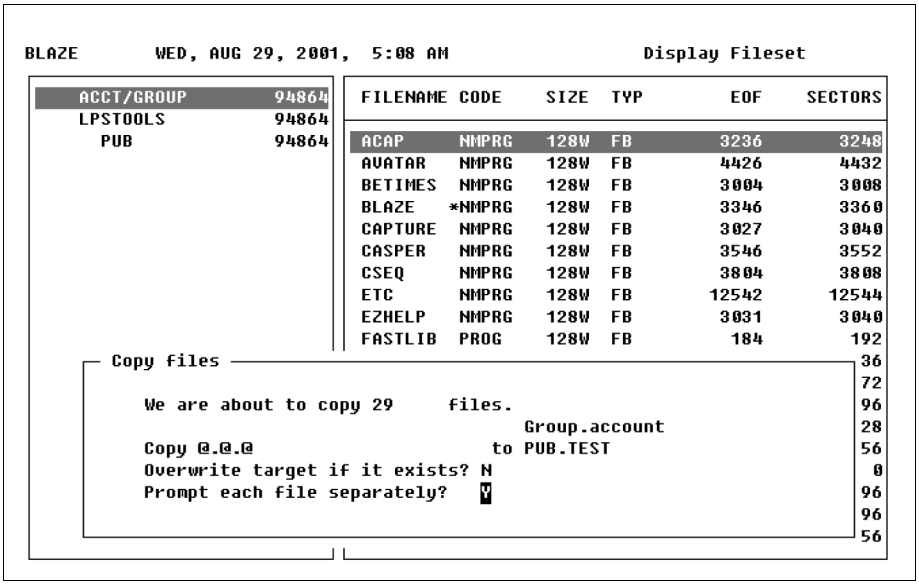

Figure 5.12 *Copy Files*

P

This SLC is used to purge file subsets. Invoking the P SLC results in a pop-up window being displayed on top of the current window (see Figure 5.13). This window is titled "Purge Files". By default you will need to confirm each purge operation. If you do not want to confirm each purge, then answer the confirm question with an "**N**".

Press F6 (Accept) to initiate file purging.

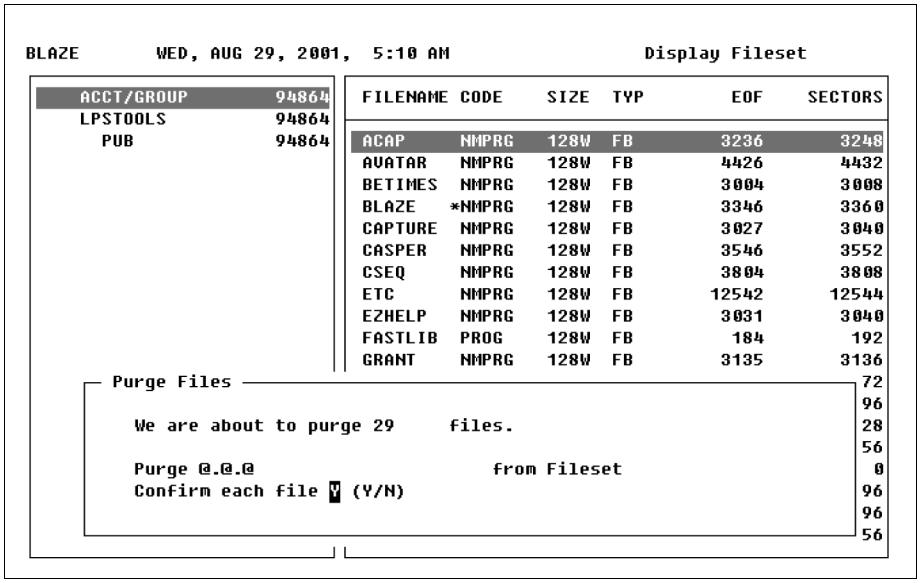

Figure 5.13 *Purge Files*

R

This SLC is used to rename file subsets. Pressing the R SLC results in a pop-up window being displayed in front of the current window (see Figure 5.14). This window is titled "Rename Files". By default the rename command will NOT purge files that exist with the same target name. If you want BLAZE to purge an existing file with the same name, answer "**Y**" to the "Purge existing destinations" question. Otherwise, files with existing names will not be renamed.

Press F6 (Accept) to initiate file renaming.

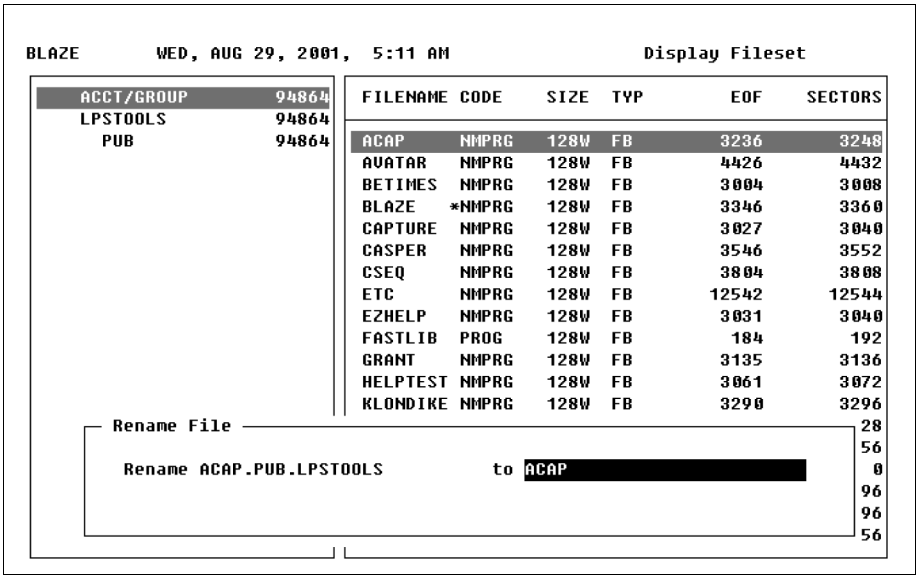

Figure 5.14 *Rename Files*

E

This SLC is used to execute an MPE command against a fileset. Invoking the E SLC results in a pop-up window being displayed on top of the current window (see Figure 5.15). This window is titled "Execute MPE command". Executable MPE commands can be any valid MPE command or a UDC.

This command is most useful when you have to repeat the same basic operation on a number of files. An example of this would be generating hard copies of a number of selected files without having to issue a PRINT command for each file. The PRINT command can be issued once using the BLAZE EXECUTE command to print all of the files listed in the selected subset. The default output destination for the PRINT command is the terminal. A file equation to redirect printed output to the line printer can be issued from within BLAZE using the MPE command option in the main menu. The file equation used for this example was FILE PRN;DEV=LP.

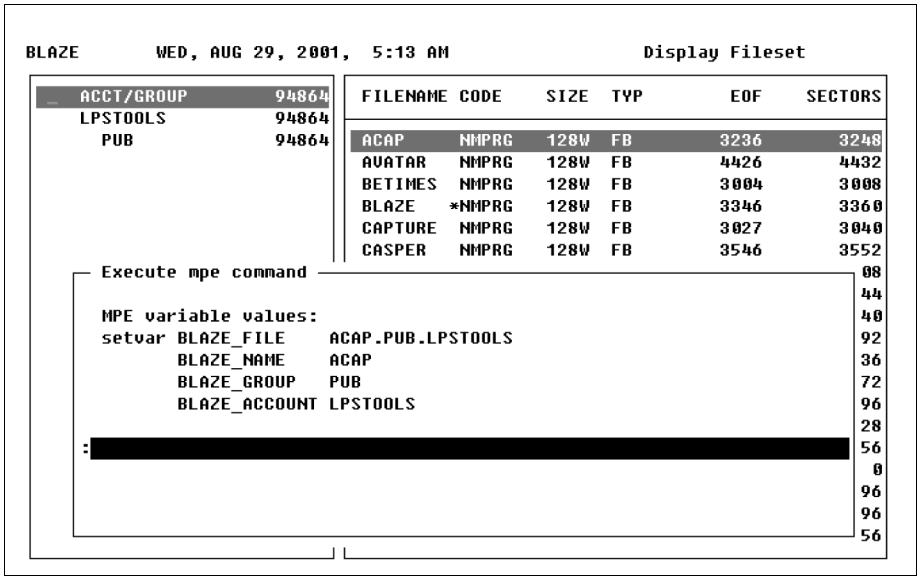

Figure 5.15 *Execute MPE command*

Notice that this example uses a file subset (see the label on the upper right corner of the screen, Display Subset(1)).

There are four environment variables that BLAZE initializes before each invocation of the specified MPE command. These are: **BLAZE_FILE**, **BLAZE_NAME**, **BLAZE_GROUP**, **BLAZE_ACCOUNT**. They are initialized as follows:

BLAZE_FILE	= fully qualified filename
BLAZE_NAME	= only the MPE filename
BLAZE_GROUP	= only the MPE group name
BLAZE_ACCOUNT	= only the MPE account name

When you enter the MPE command that you want executed, simply substitute the appropriate environment variable name where you would have normally entered filename information. Remember to de-reference the variables by preceding each variable name with the de-reference character (**!**). For example, using the COBOL compiler, compile each file into a file with the same name in the **.OBJ-group**:

```
:cob74xl!BLAZE_NAME,!BLAZE_NAME.OBJ,$null
```

Press F6 (ACCEPT) to initiate repeated execution of this command.

NOTE When the File List window is active, only one file will be processed.

Z

The Crunch (Zap) SLC is used to recover wasted disk space. Wasted disk space occurs because of the disk allocation method that MPE/iX uses. Disk space is requested in sectors, however MPE/iX typically does not allocate sectors. rather, MPE/iX allocates disk space in multiple sector blocks. The number of sectors in a block is dependent on many variables. So, unless you happen to create a file whose size is a multiple of the block allocation size, you will end up with wasted disk space.

Invoking the Z SLC results in a pop-up window being displayed on top of the current window (see Figure 5.16). This window is titled "Crunch Files". By default you will need to confirm each crunch operation. If you do not want to confirm each crunch, then answer the confirm question with an "**N**".

Press F6 (Accept) to initiate file crunching.

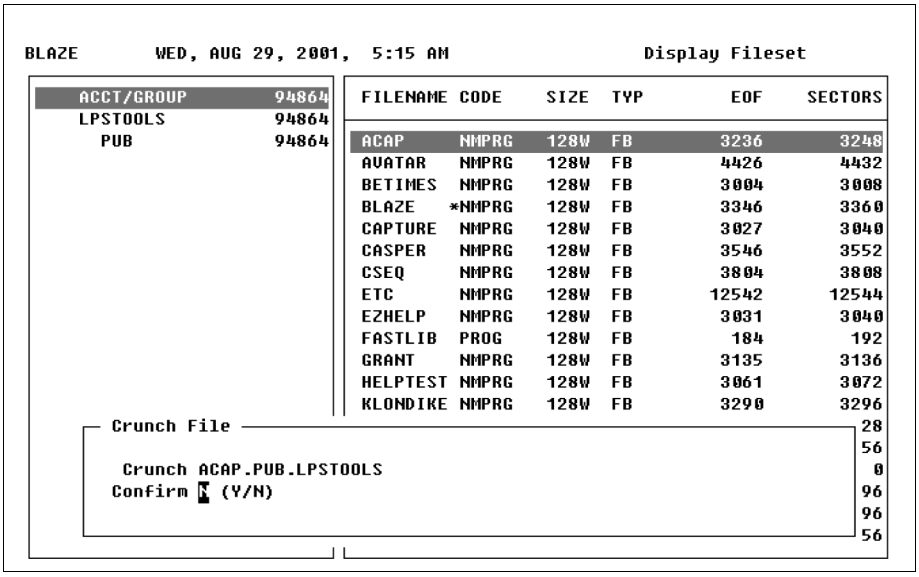

Figure 5.16 *Crunch Files*

It is not uncommon for crunching to recover thousands of sectors of disk space.

File Finding Commands

The SLCs in this section are used for locating files within the active fileset. Anything in the File List window can be the target of a find pattern. Often, there will be hundreds of files in a fileset. Locating a particular file in a fileset this large can be a real chore. These commands speed up the file locating process. Searching takes place whenever a **find text** (>) command or **find previous** (<) command is entered. If found, the File List window is updated so that the "located" file is highlighted.

NOTE Searching can only take place when the File List window is active.

/

This SLC sets up a search string for the Find commands. Using this SLC will cause a small single line window pop-up on the top of the current window on the top of the display screen (see Figure 5.18). In this window, enter a pattern to search for (including embedded spaces if necessary). Character upshifting is automatic. Examples include: "**myfile, 40w fb,nmobj.**"

> | <

Use ">" to begin searching "downward" from the current location in the File List window. Likewise, use "<" to begin searching "upward" from the current location. If no pattern is defined, BLAZE will issue a single beep to indicate the error. The "/" command must be used before directional searches become available.

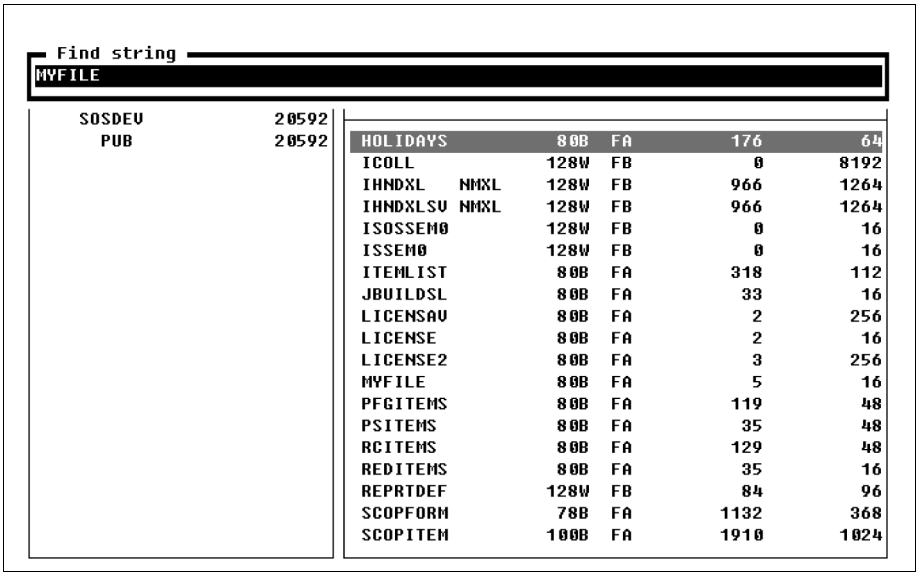

Figure 5.17 *FIND Command*

Help, BLAZE Single Letter Command Key Summary

Selecting this SLC displays a pop-up window that lists all of the single-letter command keys.

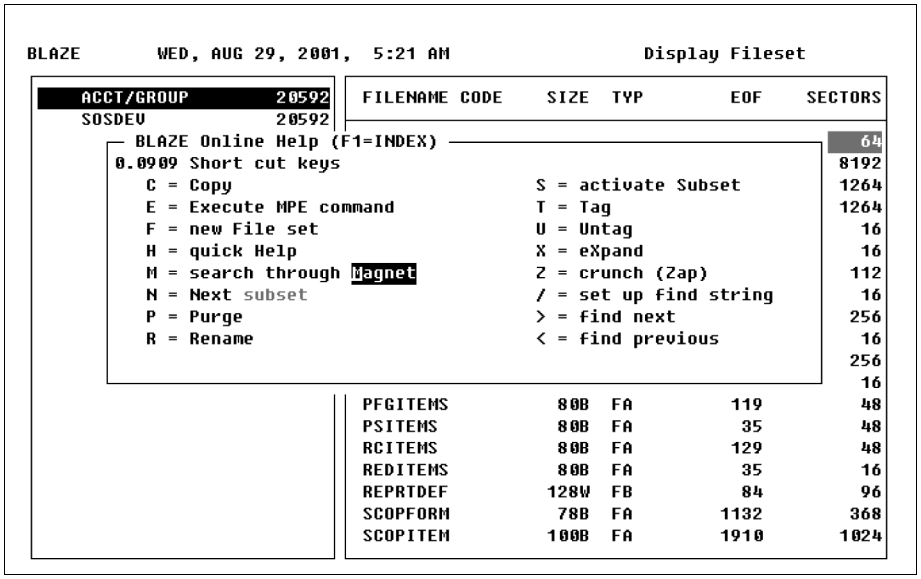

Figure 5.18 *Single Letter Command Keys*

BLAZE Function Keys

Common function keys include Help, Print, Refresh, Exit, and so forth. This section discusses function key operations that are specific to BLAZE.

Depending on which screen is active, some or all of the function keys will be available for you to use. The actual function performed by these keys will vary slightly depending on the context. For example, the CANCEL FUNCTION key is used to return from the Help subsystem.

At other times the function keys are used for navigating through BLAZE windows, traversing the fileset tree, or selecting options based on the task at hand.

WINDOW TOGGLE

This function is not always available. When it is available it is accessed through the F3 function key. This function is used to switch between the various BLAZE windows.

When the Tree screen is active, F3 is used to toggle between the Account Structure window and the File List window. When the View screen is active, F3 is used to toggle between the Account Structure window, the File List window, and the File Content window.

ZOOM

This function is not always available. When ZOOM is available it is accessed through the F4 function key. ZOOM provides two functions: ZOOM IN and ZOOM OUT. The F8 function key is used to ZOOM OUT.

While ZOOM functions are available on most BLAZE screens, the function performed is highly context-sensitive. Logically, ZOOM OUT restores your display to its prior-ZOOM IN state.

In its most common role, ZOOM IN simply enlarges the current window to use the entire display. Other times, using ZOOM IN can redefine function keys or call up other BLAZE menus. If the Tree or View screen is active, zooming in is used to provide more detailed information about specified object. For example, zooming in when the File Content window is active expands the display. Several display formats are accessed through the function keys.

F4	Hex & Ascii (Hexadecimal & Ascii combination)
F5	Hex Display (Hexadecimal only)
F6	Ascii Display (Ascii only)
F7	Ascii Filter
F8	Zoom Out

Example: Zooming in when the Files List window is active will provide general information about the file:

F4	List Security
F5	Listf -1
F6	List File
F8	Zoom Out

Example: Zooming in when the Account Structure window is active will result in 1 of 3 possible displays depending on where the cursor was positioned in that window:

If the ACCT/GROUP row is selected, zooming in will call up the Profile screen display. If an Account is selected, zooming in provides the following information through the function keys:

F4	Listacct
F5	Report
F8	Zoom Out

If a Group is selected, zooming in provides the following information through the function keys:

F4	Listgrp (List Group)

F5	Listf,2
F6	Listf, -2
F8	Zoom Out

6

THE CASPER TOOL

HP dropped the old SPOOK program. CASPER is intended to be a replacement program for SPOOK. CASPER provides access to Native Mode spoolfile in a way that will be familiar to anyone who has ever used SPOOK.

Operation

CASPER has two modes of operation: strict SPOOK emulation and standard mode. Strict SPOOK emulation mode is designed for users who want to continue using the spooler in the manner provided by SPOOK and need output formatted exactly the way SPOOK produced it.

Standard mode is functionally similar to SPOOK but it expands the range of operations to include commands that make use of NM spooler capabilities.

The mode of operation is determined by which program file you run. Two NMPRG files, SPOOK.PUB.LPSTOOLS and CASPER.PUB.LPSTOOLS, are delivered with the **System Manager's Toolbox**. If you choose to run in SPOOK mode, then strict SPOOK emulation is used which provides a SPOOK-duplicate for the NM platform.

Strict SPOOK Emulation Mode

This section describes the operations of Strict SPOOK Emulation, the various settings that can be used to modify operations, and how to put the settings into effect. Following these discussions, standard CASPER mode operations are explained.

Strict SPOOK operations are available through the SPOOK.PUB.LPSTOOLS program. When you run this version, the displays and prompts will, by default, match the A.11.60 ? version of SPOOK.

In order to provide the greatest degree of compatibility with users and applications which used SPOOK on MPE V, a user-definable banner file called SPOOKHDR.DATA.LPSTOOLS is provided. This file contains several examples of banners that can be displayed when CASPER's SPOOK program is run. By default, the A.11.60 ? banner is displayed. However, this is easily changed by replacing the first line in the file with the banner (or text) of your choice.

Other operating issues are controlled through a ICW. Because SPOOK existed in so many flavors, several modes of operation from which you choose are provided for you. In addition to emulating the old SPOOK, CASPER's SPOOK provides some extended modes of operation. These are also controlled through the JCW. The Table 6.1 lists the various JCW settings.

JCW Settings (SPOOKFLAGS)

Table 6.1 *JCW Settings (SPOOKFLAGS)*

Meaning	Bit	Decimal Value
Extended SPOOK operation	15	1
Strict SPOOK emulation	14	2
MPE XL SPOOK emulation (pre 2.1)	13	4
MPE V SPOOK emulation	12	8
Want output paging (extended feature)	11	16
Not suspendable (extended feature)	10	32
Continue after executing INFOSTRING	9	64
Display CASPER banner, instead of SPOOK's banner	8	128

JCW Value Descriptions

The following is a detailed look at the JCW settings.

Extended SPOOK operation: SPOOKFLAGS=1

In this mode of operation the user can use extensions to the SPOOK command set that are available in CASPER.PUB.LPSTOOLS. This includes commands like: **Watch**, and **Text Next**, and **Purge SAVEd**. For more information on these extensions refer to the CASPER documentation which appears in the next section.

Strict SPOOK Emulation: SPOOKFLAGS=2

When this mode is selected, only those commands that were available on classic SPOOK are available. Output also conforms to classic SPOOK conventions. Use this mode when the greatest degree of compatibility is required. This setting can be used with SPOOKFLAGS equal to 4 or 8 to fully specify which version of SPOOK you want to emulate.

MPEXL SPOOK Emulation: SPOOKFLAGS=4

Using this flag creates output identical to that produced by SPOOK A.11.60. This version of SPOOK was distributed with MPE XL prior to XL 2.1.

MPE V SPOOK Emulation: SPOOKFLAGS=8

Use this flag to generate output identical to that of SPOOK A.03.05 ?. This version of SPOOK was distributed with MPE V.

Output Paging: SPOOKFLAGS=16

Use this setting to tell SPOOK that you want it to pause after each page of output, even in strict SPOOK emulation mode.

Non Suspendable: SPOOKFLAGS=32

When this setting is used, SPOOK will interpret the EXIT command to mean quit. By default, the EXIT operation is equivalent to suspend. This gives applications the opportunity to activate SPOOK, eliminating the overhead of restarting SPOOK each time. This is particularly useful while working in environments like QEDIT and MPEX.

Continue After Executing INFO String: SPOOKFLAGS=64

With this setting, SPOOK executes the command entered in the INFO string and then remains active. Normally, SPOOK would execute the command and then terminate.

Display CASPER Banner, Instead of SPOOK's Banner: SPOOKFLAG=128

When selected, SPOOK displays the CASPER startup banner. Actual operations are controlled by the other SPOOKFLAG settings.

How to Set SPOOKFLAGS

Use the SETJCW command and the word SPOOKFLAGS to specify SPOOK JCW values. Then, use the decimal values noted in the JCW Settings Chart to indicate the flags you want to set. For example, Strict SPOOK emulation mode has a decimal value of "2". To run in this mode, type "setjcw spookflags 2" and then run the program. For example:

```
:setjcw spookflags 2

:run spook.pub.lpstools
```

Running SPOOK in strict emulation mode with the CASPER banner would be accomplished using the following settings:

```
:setjcw spookflags 128+2

:run spook.pub.lpstools
```

For a complete discussion of SPOOK commands and operation, refer to the MPE V System Utility Manual.

Standard CASPER Mode

The following sections discuss the standard mode of operation for CASPER. Standard mode is used when you run the CASPER.PUB.LPSTOOLS program. It is the default mode of operation when the ***LPS-Tools*** UDC is set.

In Standard mode, all of the standard SPOOK mode commands a Job Control Word settings can be used. Additionally, several more commands are available that take advantage of the NM spooler capabilities. Global ***LPS-Tools*** commands, like LISTREDO and COPYLP, are also available. Included in the following sections are usage and command syntax for CASPER.

Capabilities

Program capabilities required include IA, BA, PH, DS and PM. SM and OP may be needed for system management tasks.

Usage

Either mode can be started from the supplied UDC or from a RUN statement.

SPOOK Emulation

SPOOK emulation can be started from the supplied UDC or from a RUN statement:

- UDC

  ```
  :SPOOK
  ```

- RUN

  ```
  :RUN SPOOK.PUB.LPSTOOLS
  ```

CASPER

CASPER can be started from the supplied UDC or from a RUN statement.

- UDC

  ```
  :CASPER "[commands]" [parm=#]
  ```

- RUN

  ```
  :RUN CASPER.PUB.LPSTOOLS;info="[commands]";[parm=#]
  ```

Command Summary

The following list provides a simple description of CASPER commands that you can use to quickly locate the command that suits the task at hand.

NOTE Portions of the command codes are printed in uppercase to denote the part of the command that CASPER requires in order to distinguish one command from another. However, the commands themselves are not case-sensitive.

Table 6.2 *CASPER commands*

Command Code	Description
Alter	Alters the characteristics of native mode spool files
Copy	Copies a portion (or all) of a spool file to a flat file
Exit	Terminates program with suspend
Find	Locates character string in TEXTed spoolfile
HELP	Invokes CASPER help
LALL	Synonym for LIST ALL
List	Lists spool file to terminal
LL	Lists last page of TEXTed file
Purge	Deletes one or more output spoolfiles
Quit	Terminates program with nosuspend
RUN	Starts a MPE program
SET/REset	Enables or disables CASPER options
Show	Displays spool files by user.account and/or state
SReady	Displays READY spool files by user.account
Text	Opens an output spoolfile
Watch	Monitor spoolfile creation

Commands (and keywords) may be abbreviated to just the uppercase portion.

At startup, CASPER reads SPOOK.CFG or, if not found, SPOOK.CFG.LPSTOOLS, and executes any commands it finds in the file (i.e., typically SET/RESET commands).

The SAVED Buffer

One of the enhancements made to CASPER that is not available in SPOOK is the concept of a SAVED buffer. The SAVED buffer comes into play when the SHOW command is used. Every time the SHOW command is used, its output goes into the SAVED buffer and is given a relative number to reference each file in the SAVED buffer. For example, if 20 spoolfiles are displayed as a result of the SHOW command, then those 20 files are stored in the saved buffer with relative file number 1-20. The way you tell CASPER that you are referencing a spoolfile with a relative spoolfile number rather than an MPE/iX assigned spoolfile number is by preceding the spoolfile number with a minus (-) sign. For example:

`CASPER: S @.SYS`	(Display all spoolfiles for @.SYS)
`CASPER: T -1`	(Text in relative spoolfile #1)

Each time a SHOW command is issued the SAVED buffer is overwritten, unless the append operator (+) is used:

`CASPER: S+@.SYS`	(Display all spoolfiles for @.SYS).
`CASPER: S+@.HPOFFICE`	(Add @.HPOFFICE to SAVED buffer)

After a SAVED buffer is built, commands like the following can be used:

`CASPER: S SAVED`	(Display SAVED buffer)
`CASPER: P SAVED`	(Purge all spoolfiles in the SAVED buffer. This command is great for cleaning up OUT.HPSPOOL)
`CASPER: T -1`	(TEXT in the first spoolfile from the SAVED buffer)
`CASPER: T N`	(TEXT in the NEXT spoolfile from the SAVED buffer) For example, after TEXTing in a relative spoolfile, the TEXT NEXT (TN or T N) command can be used to TEXT in the next relative spoolfile. CASPER's TEXT command also supports TEXT FIRST (T F), TEXT LAST (T L), and TEXT PRIOR (T P).

Command Definitions

Listed below are detailed descriptions of the CASPER commands:

Alter

The Alter command has the following syntax:

```
Alter <spoolfilelist> <alteroptions>
```

The ALTER command is used to change the characteristics of an output spoolfile.

```
<spoolfilelist> = [user[.account]] [...]
                  [spoolid [,spoolid] [...]]
                  [spoolid [/spoolid]]
                  [*]
```

spoolfilelist defaults to all spoolfiles create by the current **[user[.account]]**. The default **spoolfilelist** for the console user is all the spoolfiles on the system.

Spoolid is the number assigned to the spoolfile by the nativemode spooler (Decimal number with one or more digits. Use the **Show** command to locate spoolfiles.). The **#O** portion of the **spoolid** is not required. Multiple **spoolids** may be specified if they are separated by commas, or a range of **spoolids** may be used. Entering an asterisk (*) is equal to the spoolfile previously selected by the **Text** command.

```
<Alteroptions> = [Copies=# [;DEVice=device[;Priority=#[;DEFER]]]]
```

Copies=#	Specifies the number of copies to print. The valid range is from 1 to 127.
DEVice=ldev	Specifies a new logical device number for the spoolfile's destination. The new logical device must be a spooled device.
Priority=outpri	Used to change the output priority of the specified spoolfiles. The lowest priority is 0 and the highest priority is 14.
DEFER	This option changes the spoolfile's state to DEFER. The spoolfile's priority remains unchanged.

Copy

This command has the following syntax:

```
Copy line1/line2, *file
```

Copies the specified line range to the desired output file.

Exit or "//"

The EXIT command causes CASPER to suspend if its parent is not the top-level command interpreter, otherwise it will terminate.

NOTE This is the same functionality as SPOOK's EXIT command. If you really want CASPER to terminate, use the QUIT command.

Find

This command has the following syntax:

```
Find [start | @] "pattern" [up] [,range]
```

The FIND command locates and displays the specified text string, given by "pattern" in the active spoolfile.

@	Scans entire line
start	Scans leading character only (DEFAULT)
pattern	Test string to search for
up	Upshift pattern and text before compare
range	Range of lines to scan

NOTE FIND can only be used after the TEXT command has been successfully issued.

HELP

This command invokes the CASPER help facility.

LALL

This command is a synonym for "LIST ALL".

List

This command has the following syntax:

```
List  [ <linerange> ] [ "pattern" [UP] ]
```

The LIST command lists the specified lines of the current spoolfile. If a "pattern" is specified (text within quote marks (''), the optional keyword UP may follow it. If a pattern is specified, only the lines within the desired range which contain the pattern will be listed. If UP is specified, the pattern is upshifted and the text is upshifted before the check is made (if the pattern is found the original un-shifted line is listed).

If a **<linerange>** is not given, **ALL** is assumed.

```
<linerange>   <linenumber> | <linenumber> / [ <linenumber> ] | ALL

<linenumber>  # | FIRST | LAST | LAST-#
```

The **<linerange>** allows a single line or multiple lines to be specified. If LAST-# is specified, it refers to # lines before the end of the file.

LL

This command has the following syntax:

```
LL [ "pattern" [UP] ]
```

The LL command lists the last 20 lines of the current spoolfile. A pattern may be specified. Because LL ... is equivalent to LIST LAST-20/LAST, refer to the LIST command for more information.

"pattern" [UP]	Used to specify a text string for which to search within the last "Texted" spoolfile. The text string may contain imbedded blanks. If the "UP" parameter is specified, the pattern and text are upshifted before the check is made. If a match occurs, then the original "unupshifted" line will be displayed.

Examples of using the LL command include the following:

```
L 1/50 "LASER"

L */100 "LASER"

L LAST-100/LAST "LASER"

L First/100 'LASER'

L L-50/L
```

Purge

This command has the following syntax:

```
Purge  <spoolfilerange> [ , <spoolfilerange> ... ]
```

The PURGE command purges the specified spoolfiles. If a range of spoolfiles is specified then interactive users will be asked for confirmation.

```
<spoolfilerange>   # | #/# | SAVED
```

If # is negative, it is treated as a relative spoolfile number (see: SHOW)

SAVED refers to the set of spoolfiles shown by the last Show command.

For example:

```
purge 730/736

purge 730,733,735

purge -1/-3
```

```
purge saved
```

Quit

This command terminates the tool. It is the same as the Exit command.

RUN

This command has the following syntax:

```
RUN <progname>
```

Start an MPE program as a child process.

SET/RESet

This command has the following syntax:

```
SET/RESET  <FAST  | SHOWNUMbers | SHOWCCTL | NUMBERED | UNNumbered
            DEVCL | CONSOLEALL | READYONLY>
```

The FAST option makes the Show command much faster, but at the cost of getting about 10 extra lines of output concerning the overall status of all spool files.

The CONSOLEALL option makes a Show command with no parameters equivalent to SHOW @.@ if you are on the system console. Default: SET CONSOLEALL.

The READYONLY option makes Show commands that don't have a **state** qualification have an implied **;READY**. Default: RESET READONLY.

SHOWCCTL is similar to the MODE CONTROLS=ON of SPOOK, but different. Default: RESET SHOWCCTL.

The SHOWNUMBERS option (synonym: NUMBERED, antonym: UNNUMBERED) tells CASPER you want to see line numbers in LIST output. Default: SET SHOWNUMBERS.

The DEVCL option adds four columns to the SHOW output: DEV/CL, PR (priority), COP (#copies), and RFN (currently blank).

Show and SReady

These commands have the following syntax:

```
Show   [+] [user[.acct]] [;@O] [;state] [;DAYsold #days]
Show   SAVEd                           [;DAYSNew #days]
SReady [+]  [user[.acct]]  [;@O]  [;state]
```

The SHOW command searches for, and displays, spool file IDs that match your criteria.

The **user** and **acct** fields may be wildcarded.

The **;state** option allows you to specify that the spool files reported are in a specified set of states. The states are:

```
OPEN, ACTIVE, CREATE, DEFER, READY, XFER, PRINT, PROBLM,
DELPND, SPSAVE
```

If no **;state** is supplied, and if a SET READYonly isn't in effect, then the SHOW command will not restrict the files shown by state.

DAYSNEW #days selects only those spool files created the specified number of days ago or less. For example, DAYSNEW 0 selects spoolfiles that were created today, DAYSNEW 1 selects spoolfiles created today or yesterday.

DAYSOLD #days selects only those spool files created at least the specified number of days ago. For example, DAYSOLD 2 selects spoolfiles that are 2 or 3 or more days old.

The SREADY command is equivalent to a SHOW command with an implied **;READY** at the end. Extra states may be added if desired (i.e., SREADY @;OPEN).

The SHOW command saves the first 200 or so output lines in a scratch file so that subsequent TEXT and PURGE commands can use "relative" spoolfile numbers. The first spoolfile shown by a SHOW is relative number -1, the second is -2, etc.

Normally, each SHOW command (except SHOW SAVED) resets the saved spoolfile number list to empty at the start. Specifying the "+" option will make the current SHOW command append to the saved spoolfile number list.

The SHOW SAVEd command will display the list of "saved" spoolfile numbers.

Text

This command has the following syntax:

```
Text #
```

If # is a negative number, then it is treated as a relative spoolfile number (see: SHOW).

NOTE 0 is a synonym for -1, simulating the functionality of SPOOK/V prior to V MIT.

Watch

This command has the following syntax:

```
Watch [seconds]
```

After selecting a spoolfile (in create mode) with the Text command the user can "watch" the creation of the spoolfile. By default CASPER looks at the spoolfile every 3 seconds for new output, however, the seconds parameter can be used to change how often CASPER looks. As

information is written to the spoolfile, CASPER will echo it to the screen. When the spoolfile leaves create mode, CASPER will stop watching it.

Xplain

The XPLAIN command results in output identical to HP's SPOOK XPLAIN command:

```
DEBUG
EXIT  <<TERMINATE IF NOT A SON PROCESS>>
XPLAIN
SHOW   [ USER [ .ACCOUNT ] ] [ ; [@] [I] [O] ]
SHOW   DEVICEFILEID [ , DEVICEFILEID ]....
TEXT   DEVICEFILEID
LIST   [ RANGE ]
FIND   [ @ ] [ "STRING" ] [ , FRANGE ]
MODE   [ OPTION [ , OPTION ]...]
       OPTION = WIDTH / CONTROLS
ALTER {DFID [,DFID[,...]]} [ ; OPTION [ , OPTION ]....]
ALTER {USER [.ACCOUNT]   } [ ; OPTION [ , OPTION ]....]
       OPTION = PRI / COPIES / DEV
PURGE  DEVICEFILEID [ , DEVICEFILEID ]....
INPUT  [ USER [ .ACCOUNT ] ] ; TAPEFILE
INPUT  DEVICEFILEID [ , DEVICEFILEID ].. ; TAPEFILE
OUTPUT [ USER [ .ACCOUNT ] ] ; TAPEFILE [; PURGE]
OUTPUT DEVFILEID [, DEVFILEID ] .. ; TAPEFILE [; PURGE]
HELP
RUN    PROGRAMFILENAME [ .GROUP [ .ACCOUNT] ]
KILL  << SON PROCESS >>
QUIT  << TERMINATE >>
COPY   [RANGE [,FILENAME]]
COPY   [DFID [,DFID [,...]] ;] [RANGE [,FILENAME]]
COPY   [USER [.ACCOUNT] ;]  [RANGE [,FILENAME]]
APPEND [RANGE [,FILENAME]]
```

```
APPEND [DFID [,DFID [,...]] ;] [RANGE [,FILENAME]]
APPEND [USER [.ACCOUNT] ;]  [RANGE [,FILENAME]]
       [END  ]
```

TOOLBOX STANDARDS

The ToolBox collections from Lund Performance Solutions have a uniform user interface. As a result, in addition to the commands specific to each Toolbox tool, most tools allow the commands described in “TOOLBOX STANDARDS” on page 311.

CASPER Examples

Figure 6.1 and Figure 6.2 illustrate the latest features implemented since the last release of CASPER. The remaining examples illustrate pre-1.0 operations and other CASPER extensions. For additional examples on how to use strict SPOOK emulation, refer to the MPE V Systems Utility Manual.

```
SPOOK [2.11] - LPS Toolbox A.09f                    (Copyright (c) Lund)

Edit the first line to be your desired "copyright" line for
"SPOOK".  Only the first line is read by SPOOK.PUB.LPSTOOLS.

Copyright lines from some Hewlett Packard versions of SPOOK are:
   Classic Spook: SPOOK G.03.05  (C) HEWLETT-PACKARD CO., 1983
   MPE XL Spook:  SPOOK A.11.60  (C) HEWLETT-PACKARD CO., 1983
```

Figure 6.1 *The Contents of SPOOKHDR.DATA.LPSTOOLS*

```
Wolf:/LPSTOOLS/PUB: setjcw spookflags 16+2
Wolf:/LPSTOOLS/PUB: spook
This product is licensed to: ImageStats Demo
SPOOK [2.11] - LPS Toolbox A.09f                    (Copyright (c) Lund)
> e
Wolf:/LPSTOOLS/PUB:
```

Figure 6.2 *Setting SPOOKFLAGS for STRICT MODE (2) and PAGING (16)*

```
: casper.pub.lpstools

This product is licensed to: lps
SPOOK [2.11] - LPS Toolbox [A.09f]          (c) 1995 Lund Performance Solutions


CASPER: spook

This product is licensed to: lps
SPOOK [2.11] - LPS Toolbox A.09f                    (Copyright (c) Lund)
> s@.sys
#FILE    #JOB     FNAME     STATE   OWNER
#081     #J'1     $STDLIST READY   MANAGER.SYS
#088     #J1      $STDLIST OPEN    MANAGER.SYS
#0100    #J9      $STDLIST READY   MANAGER.SYS
#0101    #J10     $STDLIST READY   MANAGER.SYS
#I1      #J1      $STDIN   OPEN    MANAGER.SYS
> s saved
-### #FILE    #JOB     FNAME     STATE   OWNER
  -1 #081     #J'1     $STDLIST READY   MANAGER.SYS
  -2 #088     #J1      $STDLIST OPEN    MANAGER.SYS
  -3 #0100    #J9      $STDLIST READY   MANAGER.SYS
  -4 #0101    #J10     $STDLIST READY   MANAGER.SYS
  -5 #I1      #J1      $STDIN   OPEN    MANAGER.SYS
> text -4
[#0101]> exit
```

Figure 6.3 *Accessing SAVEd Spoolfile List*

```
CASPER: spook

This product is licensed to: lps
SPOOK [2.11] - LPS Toolbox A.09f                      (Copyright (c) Lund)
> s @.sys;ready
#FILE    #JOB     FNAME     STATE   OWNER
#081     #J'1     $STDLIST READY   MANAGER.SYS
#0100    #J9      $STDLIST READY   MANAGER.SYS
#0101    #J10     $STDLIST READY   MANAGER.SYS
> text last
[#0101]> text first
[#081]> text next
[#0100]> exit
```

Figure 6.4 *TEXT Command Modifiers*

```
CASPER: spook

This product is licensed to: lps
SPOOK [2.11] - LPS Toolbox A.09f                    (Copyright (c) Lund)
> s @.sys;ready
#FILE    #JOB    FNAME    STATE  OWNER
#O81     #J'1    $STDLIST READY  MANAGER.SYS
#O100    #J9     $STDLIST READY  MANAGER.SYS
#O101    #J10    $STDLIST READY  MANAGER.SYS
> s @.sys;@o
#FILE    #JOB    FNAME    STATE  DEV/CL   PR COP RFN OWNER
#O81     #J'1    $STDLIST READY  LP        8   1     MANAGER.SYS
#O88     #J1     $STDLIST OPEN   LP        8   1     MANAGER.SYS
#O100    #J9     $STDLIST READY  LP        1   1     MANAGER.SYS
#O101    #J10    $STDLIST READY  LP        1   1     MANAGER.SYS
#FILE    #LDEV     LABEL ADDR        SECTORS       LINES      TIME
#O81     %1    $00000001 $000a0500    256           50        10:23  9/28/01
#O88     %1    $00000001 $001b1b00    256          100
#O100    %1    $00000001 $0056dc00     32          175        12:38 10/11/01
#O101    %1    $00000001 $00570f00     32          191        12:40 10/11/01
> exit
CASPER: exit
```

Figure 6.5 *Long Output (@O) Format*

CASPER Error Messages

See the MPE V Utilities Manual for information concerning SPOOK error messages.

Table 6.3 *CASPER Error Messages*

Message	Cause	Action
Alter what?	User input needs Alter option.	Valid Alter options are: PRI, DEFER, COPIES, DEV, UNDEFER
Both numbers in range must be relative or both must be positive numbers.	Mixing relative spoolfile numbers and real spoolfile numbers is not allowed.	Do not mix range types when specifying spoolfile numbers.
Didn't find semicolon separating #0 numbers from options.	Bad user input for Alter command.	Review Alter command syntax.

Message	Cause	Action
Failed to alter #0nnnn	CASPER's "ALTSPOOLFILE" command failed.	ALTSPOOLFILE must be ALLOWed for user running CASPER. Review ALLOW command in MPE/iX Command Manual.
Invalid line range	Generally: not standard range given.	Review range syntax for command in question.
No prior spoolfile number remembered.	Relative spoolfile list may be empty.	Use Show SAVEd command to review relative spoolfile list.
No spoolfile ids remembered.	Relative spoolfile list may be empty.	Use Show SAVEd command to review relative spoolfile list.
Only ## spoolfile ids in saved list.	Relative spoolfile list may be empty.	Use Show SAVEd command to review relative spoolfile list.
Relative spoolfile number ## no longer valid.	Entered relative spoolfile number not in list.	Use Show SAVEd command to review relative spoolfile list.
Unknown option.	Bad option for Alter command.	Review Alter command syntax.
Unknown SET/RESET option: <option>, ignored	An invalid RE[SET] option was used.	Review the valid RE[SET] options for your selection.

7

THE ETC TOOL

ETC is a tool which allows a user to view file information for selected processes. One of ETC's most powerful features is its ability to predict when a sequential file access will reach the end of the file (EOF).

The predictive abilities of ETC can be very handy for applications that access data in a sequential manner such as a payroll run or a database migration. Predictions are based on system load, the current record pointer and the known EOF.

In addition to its predictive abilities, ETC provides a way for users to easily view file information that is otherwise not available in a single program.

Using ETC

Using ETC is easy. Begin by choosing a job or session for which you want process-related information. ETC identifies all processes associated with the job or session and, depending on your selection, displays them in a process list. Any process in this list has information associate with it that can be viewed using ETC.

ETC relies heavily on context-sensitive function keys to access its various features. Some function keys invoke pop-up menu boxes containing a list of options. Many of these options lead you through a selection tree where you respond to a series of prompts that define a specific information request.

This document shows you the basic views you will use to obtain process information. Options and features are described as they relate to each view. An alphabetically sorted, comprehensive function key list is included, as well.

Capabilities

SM or OP capabilities are required to run ETC.

Operation

ETC uses a window interface to cleanly manage the job/session, process and file information. All windows support scrolling. Simply use the arrow keys on the keyboard. Window information is

update whenever the **Update** function key or the **Return** key are pressed. For more information on the user interface, including configuration and feature descriptions, refer to the appendices.

Most window information can be adjusted via the Filter pop-up menu which is designed to allow for global filtering of jobs/sessions, processes or files. The Filter menu is accessed through the **Select** or **Edit** function key, which is discussed in detail later.

Each window has several common function keys like **Zoom In/Out** and Help as well as function keys specific to the current view. For example, the **Look At PINs** key is available in the Jobs/Sessions window, but not in the Processes window. In many cases, the window-specific function keys are short-cuts for menu selections.

Running ETC

First, run the program using the UDC, :ETC, or via a RUN statement. When you run the program, ETC assembles a list of all the sessions and jobs currently running on the system. This is the first view, and it always appears when you run ETC.

Whatever method you use to run the program, the first view displayed by ETC will be similar to the one shown next.

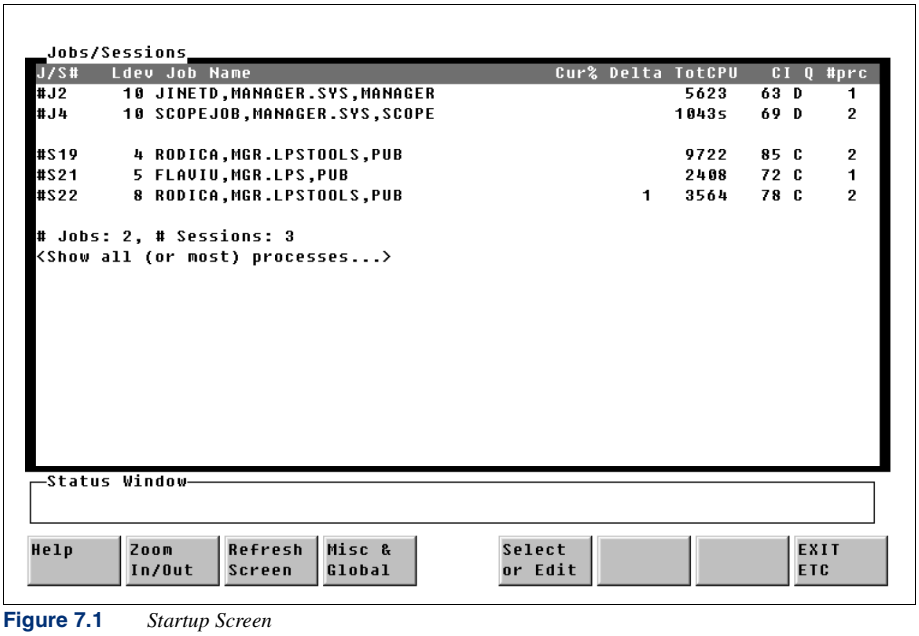

Figure 7.1 *Startup Screen*

Figure 7.1 shows the startup ETC screen. This screen contains the Jobs/Sessions window which displays a list of all jobs and sessions currently executing. Jobs are listed first, followed by sessions. This window is used to select a job or a session.

To select a job or session, use the arrow keys to highlight the job or session and the press **Return**.

Once a job or session is selected, its list of processes is displayed. At the bottom of this window is a job and session count summary followed by a line that says "**Show all (or most) processes**". This last item is useful for displaying a complete listing of all processes active on the system.

Viewing Job/Session Process Information

To view process information associated with a specific job or session, highlight the job or session and press **Return** to display the Processes window.

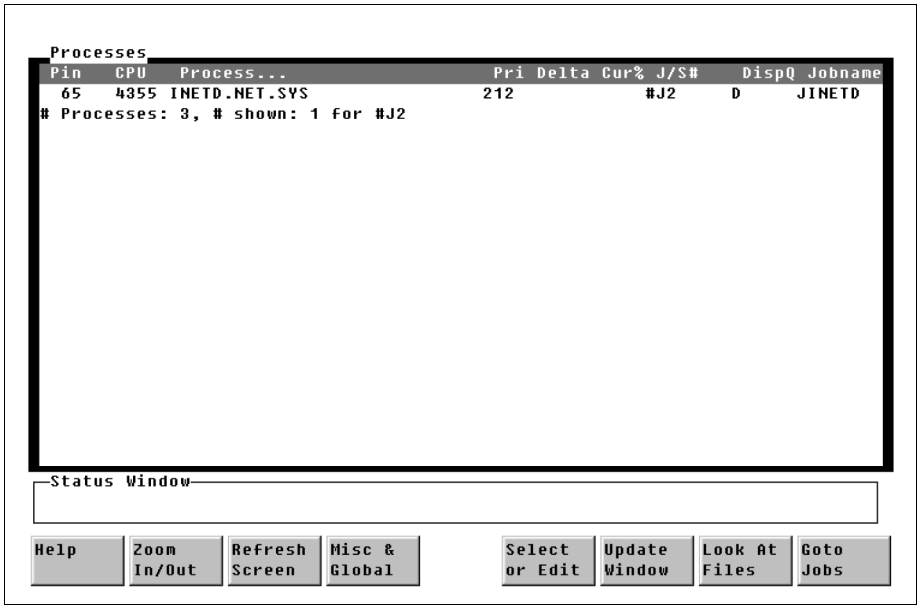

Figure 7.2 *Processes Window*

Figure 7.2 shows a Processes window that has been activated by selecting a job or session. It contains a list of all processes associated with whichever job or session was selected. **Goto** function keys that provide quick access to previous windows are added as new window overlays are added.

Use the Processes window to select a process so that its list of associated files (which are the ultimate source of information) can be displayed. By default, this window displays only user

processes. However, the F5 (**Select or Edit**) function key can be used to pop-up a window that provides access to the "**Filter Processes**" option. This option can be used to display a list of Filter definitions that expand or restrict the list of process types beyond the default setting.

Viewing System Process Information

To view process information that is not associated with specific jobs or sessions, choose "Show all (or most) processes". This selection is always the last entry in the Jobs/Sessions window.

Next, choose the F5 (**Select** or **Edit**) key to display the Processes Action pop-up menu. Now choose F3, Filter Processes option. This displays a list of selectable filter options as shown next. The option entitled "**Show Sys Procs**" is the option used for viewing system process data. In fact, choosing this option displays a listing for all processes on the system.

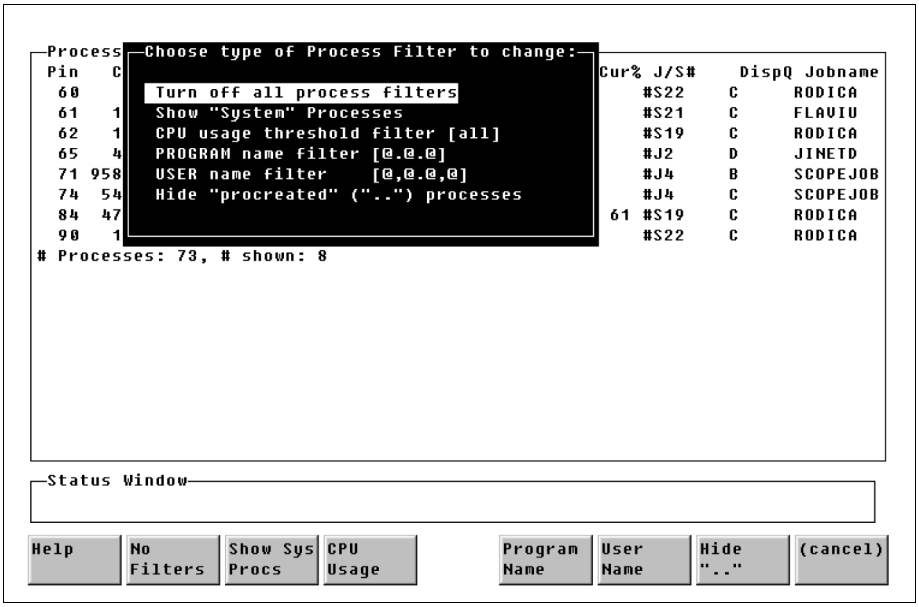

Figure 7.3 *Process Action Pop-up Menu*

Process Filters

Process filters may be employed to limit or expand upon the initial display of process types. For example, filters for the program or user names use wildcard, listf-style definitions (i.e., N@.@.S@ shows program names beginning with "**N**" in accounts beginning in "**S**"). The "**Hide [Show] "procreated" (". .") processes**" option removes (or adds) those processes created by the operating system when it starts up. The "**CPU usage threshold filter**" is used to show processes

that use a given percentage of CPU resource while "**Hide (Show) "System" Processes**" may be used to remove (or add) process types. "**Turn off all process filters**" may be used to return the filter specifications to the default, no-filter state, where the wildcard specifications are global (@.@.@) and the "**Hide**" state is active.

Viewing Process Files

Detailed process information is found in the Files window where all files associated with a given process are listed. Based on files you select, ETC "**reads**" the file for information which is returned to the screen.

To choose a file, start at the Processes window. Use the cursor keys to select the process of interest, then press **Return** or F7 (**Look at Files**) to bring up the Files window.

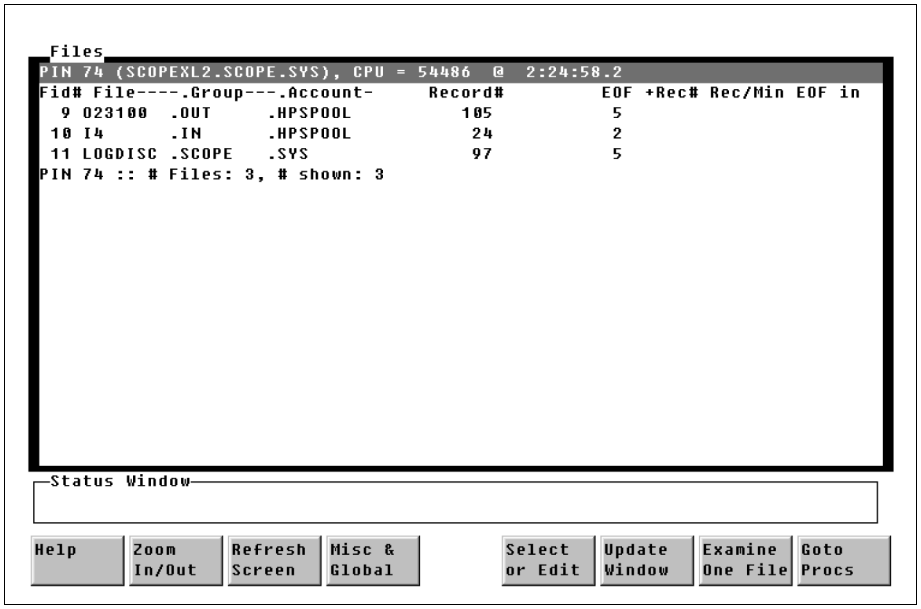

Figure 7.4 *Files Window*

Figure 7.4 shows the Files window. This window displays all of the files associated with a process. Pressing the **Return** key or F6 (**Update Window**) will refresh the data in the window.

Estimating the Time of Process Completion

Knowing when a process is likely to complete can be highly useful information if you are trying to level load or coordinate transaction jobs that involve large amounts of sequentially-read data.

This estimate is available to you via the "EOF in" column located on the far right of the screen. This column will be updated to contain the predicted time that EOF will be hit. The format is hour:minute:second.

Process File Details

More detailed information is available for each file by using the F7 (**Examine One File**) function key. Again, use the cursor keys to select the file of interest.

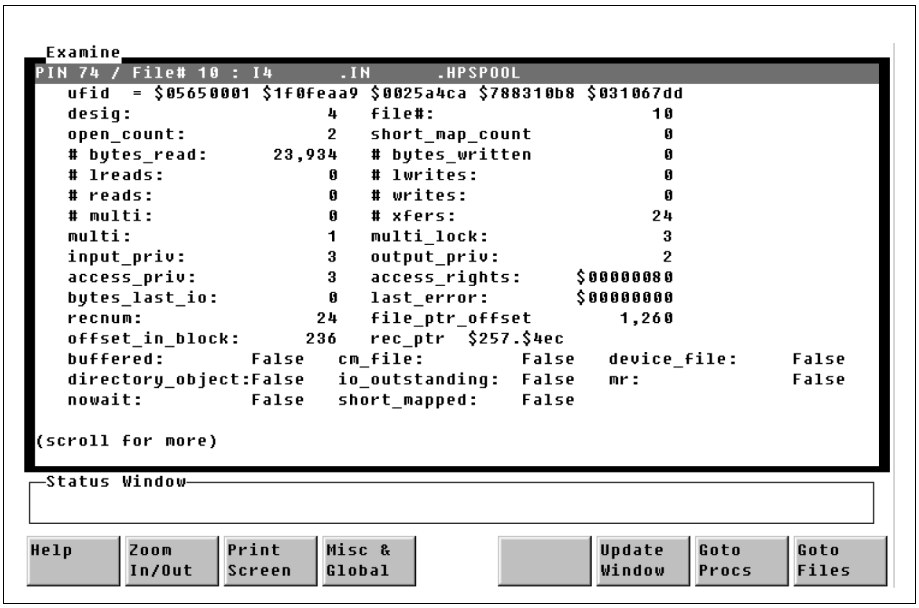

Figure 7.5 *File Examine Window*

Figure 7.5 shows the File Examine window. This window displays many attributes associated with the selected file that can be difficult (if not impossible) to find if you don't have ETC.

The information shown in Figure 7.5 tells you whether or not the file was:

1 opened as a short-mapped file,

2 buffered, or

3 offset in a block.

It also identifies the Unique File ID (UFID), multi-read data, the number of readers and users, and the privilege level.

These highly-specific, highly-technical information items are described in HP's *Architected Interface (AIF) Manual* under the intrinsic *AIFFILELGET*.

Function Key List

Following is an alphabetical list of all function keys. The second column contains the name of the window or previously selected function key from which the function was chosen. This information is provided to assist you in locating a specific key in the software.

Table 7.1 *Function Key List*

Function Key	Option/ Window*	Description
EXIT ETC	Jobs/Sessions	Terminates the program.
Examine One File	Look At Files	A comprehensive File information display for files associated with selected processes. This option is available from the "**Look At Files**" key located in the active Process window.
Filter Jobs	Select or Edit	Contains the filter options used to define jobs or sessions of interest. This key is found in the version of the "**Select or Edit**" key associated with the Jobs/Session window.
Filter Files	Files Action	Contains the filter options used to define a fileset. This key is available in an active Files Action window which is found by choosing the "**Select or Edit**" key associated with the Files window.
Filter Procs	Select or Edit	Contains the filter options used to define processes of interest. This key is found in the version of the "**Select or Edit**" key associated with the Process window.
Goto Procs	Files	Displays the Process window.
Goto Jobs	Process	Displays the Jobs/Session window.
Help	Jobs/Sessions	Displays context-sensitive help.
Look At Files	Select or Edit	Displays the files associated with a previously selected process. This key is found in the version of the "**Select or Edit**" key associated with the Process window.
Look At PINs	Select or Edit	Displays the process information numbers associated with a previously selected job or session. This key is found in the version of the "**Select or Edit**" key associated with the Job/ Session window.

Function Key	Option/ Window*	Description
Misc & Global	Jobs/Sessions	Configure settings, modify window sizes, shell out to MPE, and add optional display fields.
MPE Command	Select or Edit	A shell to MPE.
Print Screen	Examine One File	Prints the screen display to the default print device on your system.
Refresh Screen	Select or Edit	Repaints the screen, displaying any changes that have transpired.
Select or Edit	Files	Displays a Files Action window where Files-related information may be specified for the information view. **NOTE:** The context-sensitive function keys associated with the Files Action window provide a second avenue for selecting Files Action options.
Select or Edit	Jobs/Sessions	Displays a Jobs Action window where Jobs-related information may be specified for the information view. **NOTE:** The context-sensitive function keys associated with the Jobs Action window provide a second avenue for selecting Jobs Action options.
Select or Edit	Processes	Displays a Process Action window where Process-related information may be specified for the information view. **NOTE:** The context-sensitive function keys associated with the Process Action window provide a second avenue for selecting Process Action options.
Update Window	Files	Repaints the screen, displaying any changes that have transpired.
Zoom In/Out	Jobs/Sessions	Expands/contracts the active window.

*The Option/Window column contains the reference to the context in which the function key is active. For example, the "**Look At PINs**" key is available only when the "**Select or Edit**" key has already been chosen for the Jobs/Session window.

8

THE GRANT TOOL

The GRANT tool "grants" all possible capabilities to the user. This tool should be secured with a **lockword** to prevent access by unauthorized users. The capabilities granted remain in effect for the duration of the session (or job).

Operation

The reason behind the statement that GRANT gives all possible capabilities is that if you use the WHO intrinsic to inspect the capabilities word (32 bits) of a give process after you have used GRANT, you will see that all of the bits have been set. Setting all of the bits corresponds to granting all possible capabilities.

GRANT should always be secured with a lockword to prevent unauthorized access. When the **LPS-Tools/System Managers Toolbox** is installed, GRANT will have the lockword "QUARTZ". Use the MPE RENAME command to change this to the lockword of your choice.

Capabilities

Program capabilities required include IA, BA, PM, DS, and PH. No special user capabilities are required to run GRANT.

Usage

Run GRANT using the supplied UDC or a fully-qualified RUN statement.

- UDC

  ```
  :GRANT
  ```

- RUN

  ```
  :RUN GRANT.PUB.LPSTOOLS
  ```

GRANT Examples

Following is an example of how to run GRANT.

```
Wolf:/LPSTOOLS/PUB: run grant.pub.lpstools
LOCKWORD: GRANT.PUB.LPSTOOLS?


GRANT [2.5] - LPS Toolbox [A.09f]           (c) 1995 Lund Performance Solutions

For Help, :RUN GRANT.PUB.LPSTOOLS,HELP
This product is licensed to: ImageStats Demo

granted.

END OF PROGRAM
Wolf:/LPSTOOLS/PUB:
```

Figure 8.1 *Running GRANT*

GRANT Error Messages

Each GRANT error message is described in the following table.

Table 8.1 *GRANT Error Messages*

Message	Cause	Action
GRANT cannot run on this version of MPE, PCB size=xxx.	Because GRANT works directly with MPE data-structures, there are checks built into it to prevent incorrect operation due to operating system changes.	Provide the following information for Lund Performance Solutions Technical Support: your HP3000 series, operation system version, and the version of GRANT that you are using.

9

THE KLONDIKE TOOL

The KLONDIKE tool is used to load (i.e., fetch) data files into memory. Once loaded in memory, KLONDIKE can "freeze" that file into memory so that it is immediately accessible to the programs that require that information. Later, KLONDIKE's "thaw" command can be used to unload the file from memory. Freezing commonly used files can have a large impact on performance.

NOTE In the text of the documentation for KLONDIKE, references made to "page" refer to a logical page of 4,096 bytes.

Operation

KLONDIKE is a tool that can be used to improve access time to objects that are loaded into memory. It does this in several ways.

You can use KLONDIKE to fetch a file into memory. What this is really doing is "pre-fetching" a file into memory. The reason for doing this is that if you know that a given operation would benefit from having its data in memory, the you can achieve performance gains by prefetching it with KLONDIKE. For example, when you compile a program, if you fetch the source into memory, then it will save the operating system the task of checking and possibly loading the source into memory.

NOTE Fetching a file into memory does not guarantee that it will stay there. On a busy system or a system without much "user memory" (see "The PAGES Tool" on page 143 for a description of user memory) the likelihood is high that your data will at least be partially swapped out before you access it. Also, on systems with low amounts of memory, it may not be possible to fetch an entire file into memory.

You can use KLONDIKE to freeze a file into memory. This is similar to fetching except a frozen file will not be swapped out like a fetched file would. KLONDIKE's "unfreeze" or "thaw" commands will free up that frozen file. Freezing a file into memory guarantees better access to the file. Keep in mind though that you could actually reduce system performance, depending on the amount of memory in your machine, since you have in effect taken away (frozen) some of the memory that is normally available. This can lead to a situation where thrashing occurs.

KLONDIKE and PAGES are closely related in terminology. So, you may find it helpful to review the section on PAGES.

Capabilities

Program capabilities required include IA, BA, PM, DS and PH. User SM capability is required for the FREEZE and THAW commands.

Usage

To run KLONDIKE either use the supplied UDCs or use a fully-qualified RUN command.

- UDC

`:KLONDIKE`	Starts KLONDIKE, no parameters
`:COUNT <filename>`	Report on % of file in memory
`:FETCH <filename>`	Fetches a file into memory

- RUN

```
:RUN KLONDIKE.PUB.LPSTOOLS;INFO="[commands]"
```

NOTE **filename** can be either an MPE or POSIX filename.

Command Summary

The following list provides a summary description of KLONDIKE commands that you can use to quickly locate the command that suits the task at hand. Detailed information on each command is provided in the next section.

NOTE Portions of the command codes are printed in uppercase to denote the part of the command that KLONDIKE requires in order to distinguish one command from another. However, the commands themselves are not case-sensitive.

Table 9.1 *KLONDIKE commands*

Command Code	Description
COUNT	Displays percentage of file in memory
Exit	Terminates the program
FETCH	Loads file into memory
FREEZE	Freezes file into memory
Help	Displays context-sensitive Help text
Open	Tests access to a file
POST	Writes file to disk
SET/REset	Enables or disables KLONDIKE options
THAW	Unfreeze file from memory (same as UNFREEZE)
UNFREEZE	Synonym for THAW

Command Definitions

Listed below are detailed descriptions of each KLONDIKE commands. Syntax is provided in some cases.

COUNT

The COUNT command has the following syntax:

```
COUNT <filename>
```

This command is used to count the number of pages of a file that are currently in memory and then displays the results on the screen. It shows the percentage of a file in memory, as well as the number of logical pages in memory. More information is displayed if the VERBOSE option is selected.

FETCH

This command has the following syntax:

```
FETCH <filename> [[# <Pages | Bytes> | ALL]] [ <WAIT | NOWAIT> ]
```

The FETCH command opens the specified file and attempts to fetch the specified number of pages (or bytes) into memory, starting at the beginning of the file.

WAIT	Instructs KLONDIKE to wait until all of the fetched pages are in memory before continuing.
NOWAIT	(default) Instructs KLONDIKE not to wait for all of the pages to be read into memory before continuing.

NOTE Attempts to fetch more than 200 pages at a time will be broken into multiple smaller fetches automatically. This is done to prevent overloading the system with large requests.

FREEZE

This command has the following syntax:

```
FREEZE <filename> [ALL | [# <Pages | Bytes>]]
```

The FREEZE command opens the specified file and attempts to freeze the file into memory. The entire file will be frozen. A file frozen with the FREEZE command can be unfrozen with the THAW command (or its synonym, UNFREEZE).

If a page of a file is frozen, and then updated, the data will NOT be posted to disk until sometime after the file is THAWed or until a POST command is done.

OPEN

The OPEN command has the following syntax:

```
OPEN <filename>
```

This command is used to test the access to a file.

POST

This command has the following syntax:

```
POST <filename> [# <Pages | Bytes> | ALL] [<WAIT | NOWAIT>]
```

The POST command opens the specified file and attempts to post (force a write to disk) the specified number of pages (or bytes) from memory, starting at the beginning of the file.

WAIT	(default) Instructs KLONDIKE to wait until all of the posted pages have been written to disk before continuing.
NOWAIT	Instructs KLONDIKE not to wait for all the disk writes to finish.

SET | REset

The SET/RESET commands are used to specify the following options. Use these options to enable or disable information-reporting features.

PARTIAL	SET sets the specified options to "true". RESET sets the standard options to "false". If PARTIAL is true, KLONDIKE will allow partial file specification for some commands like: POST, FREEZE, and FETCH.
TIMES	SET sets the specified options to "true". RESET sets the standard options to "false". If TIMES is true, KLONDIKE will report on information relating to how much cpu time & wall time were required to perform a task.
VERBOSE	SET sets the specified options to "true". RESET sets the standard options to "false". If VERBOSE is true, KLONDIKE will report more information about a file in memory than it would if VERBOSE is false.

THAW | UNFREEZE

These commands have the following syntax:

```
THAW <filename>
```

or

```
UNFREEZE <filename>
```

These two commands perform the exact same operation. What they do is open the specified file and unfreeze the file from memory. The entire file will be unfrozen.

TOOLBOX STANDARDS

The ToolBox collections from Lund Performance Solutions have a uniform user interface. As a result, in addition to the commands specific to each Toolbox tool, most tools allow the commands described in “TOOLBOX STANDARDS” on page 311.

KLONDIKE Examples

Following are examples of the KLONDIKE tool.

Figure 9.1 shows how the COUNT command is used and what the output looks like:

```
Wolf:/LPSTOOLS/PUB: run klondike

KLONDIKE [2.8] - LPS Toolbox [A.09f]        (c) 1995 Lund Performance Solutions

For Help at the KLONDIKE prompt enter   ?
This product is licensed to: ImageStats Demo

KLONDIKE: count qedit.pub.robelle

qedit.pub.robelle @ $657.0 ... opened ok.
File: 869 pages; InMem: 0 (0.0% of file, 0.00% of memory)

KLONDIKE: exit

END OF PROGRAM
```

Figure 9.1 *COUNT Command*

Figure 9.2 shows the effect of the **Verbose** and **Times** options when used with FETCH:

```
Wolf:/LPSTOOLS/PUB:    run klondike

KLONDIKE [2.8] - LPS Toolbox [A.09f]        (c) 1995 Lund Performance Solutions

For Help at the KLONDIKE prompt enter   ?
This product is licensed to: ImageStats Demo

KLONDIKE: fetch catalog.pub.sys

catalog.pub.sys @ $c4.100 ... opened ok.
forcing WAIT option
.
File: 275 pages; InMem: 275 (100.0% of file, 0.67% of memory); 17 Referenced

KLONDIKE: set verbose
ok

KLONDIKE: set times
ok
CPU = 2, elapsed = 3 milliseconds.

KLONDIKE: fetch catalog.pub.sys

catalog.pub.sys @ $c4.100 ... opened ok.
forcing WAIT option
.
File: 275 pages; InMem: 275 (100.0% of file, 0.67% of memory); 0 Frozen;
   0 Coming In; 0 Going Out; 0 Dirty; 275 Referenced; 0 ROC; 0 Resident
CPU = 276, elapsed = 288 milliseconds.

KLONDIKE: exit

END OF PROGRAM
```

Figure 9.2 *Verbose and Times Options*

Figure 9.3 shows how to freeze a source file into memory:

```
Wolf:/LPSTOOLS/PUB: run klondike

KLONDIKE [2.8] - LPS Toolbox [A.09f]        (c) 1995 Lund Performance Solutions

For Help at the KLONDIKE prompt enter   ?
This product is licensed to: ImageStats Demo

KLONDIKE: freeze udc

udc @ $1ab.0 ... opened ok.
File: 2 pages; InMem: 2 (100.0% of file, 0.00% of memory); 2 Frozen

KLONDIKE:
```

Figure 9.3 *Freezing a Source File into Memory*

Figure 9.4 uses the CPUNT command to show that, indeed, the file has been frozen into memory. Then, the THAW command is used to unfreeze the file. Finally, the COUNT command is used to verify that the file was unfrozen:

```
Wolf:/LPSTOOLS/PUB: run klondike

KLONDIKE [2.8] - LPS Toolbox [A.09f]        (c) 1995 Lund Performance Solutions

For Help at the KLONDIKE prompt enter   ?
This product is licensed to: ImageStats Demo

KLONDIKE: count udc

udc @ $1ab.0 ... opened ok.
File: 2 pages; InMem: 2 (100.0% of file, 0.00% of memory); 2 Frozen

KLONDIKE: thaw udc

udc @ $1ab.0 ... opened ok.
File: 2 pages; InMem: 2 (100.0% of file, 0.00% of memory)

KLONDIKE: exit

END OF PROGRAM
```

Figure 9.4 *COUNT and THAW Commands*

KLONDIKE Error Messages

In the error messages that follow, "xxxx" refers to a number that is filled in at runtime by KLONDIKE.

Table 9.2 *KLONDIKE Error Messages*

Message	Cause	Action
Attempted to fetch "xxxx" logical pages, which is illegal.	User tried to fetch ZERO or less pages.	KLONDIKE can only fetch a positive number of pages.
Cannot freeze...file already has "xxxx" frozen pages.	At least one page of the specified file is already frozen (KLONDIKE does not support partial file freezes).	Determine who or what has frozen the file, unfreeze it, and then use KLONDIKE to freeze the file.
File is already frozen.	User tried to freeze a file that was previously frozen.	KLONDIKE can only freeze a file once.
File won't fit in memory...won't freeze it. Memory is only "xxxx" pages of 4,096 bytes.	An attempt was made at freezing a file into memory that was larger than the amount of memory available on the system.	Reduce the size of the file before freezing it.
KLONDIKE will not freeze/ thaw file in spaces $A or $B due to possible undesirable consequences.	KLONDIKE was instructed to perform a freeze or thaw command into memory currently used by the operating system.	No action is required, this is a warning only.

Message	Cause	Action
Limiting fetch to "xxxx" logical pages (size of memory).	User tried to fetch a file that is larger than the amount of memory on the machine.	No action is required, this is a warning only.
The file is not frozen...cannot unfreeze.	User tried to unfreeze a file that was not frozen in the first place.	Only previously frozen files can be unfrozen.

10

THE KNOCKOUT TOOL

KNOCKOUT provides a way of keeping inactive sessions from tying up all of your HP3000 terminal and MODEM port resources. It does this by monitoring all of the sessions on the system, and then aborting those which are inactive. KNOCKOUT is also distributed in a compatibility mode form, so both your MPE and MPE/iX machines can use the same mechanism for controlling inactive sessions.

NOTE The criteria by which KNOCKOUT determines inactivity status is specified entirely by you.

Operation

Typically, KNOCKOUT runs as a background process where it monitors all sessions. When a session becomes inactive (idle) for a user-definable amount of time, the KNOCKOUT aborts that session via the MPE abortjob command.

NOTE The MPE **abortjob** command must be allowed through the MPE **allow** command for the user running the KNOCKOUT Tool.

In order to tell KNOCKOUT how to determine idleness, you will need to develop a script which describes to KNOCKOUT how you want to manage your system. KNOCKOUT provides a number of options which can be specified in your script that provide for both global and individual criteria for determining idleness.

A KNOCKOUT script is constructed with one or more IDLE commands, EXCLUDE commands, and global options. All of this is typically followed with an END command which defines the end of input. Also the LOOP command can be used to modify the "sample" rate at which KNOCKOUT runs. The sample rate is used to determine how often KNOCKOUT checks sessions for idleness.

Idle Checking Algorithm

When KNOCKOUT fits and idle session, it checks the logical device of the session against the list of EXCLUDEd LDEVs (if any). If the session is in the exclude list, then it is left alone. Otherwise, the next set of steps occurs.

First, the idle sessions job name is checked against the list of users specified in the IDLE commands. If the session matches one of the IDLE command users, and the idle-time is greater than the value specified in the IDLE command, then KNOCKOUT checks to see if this user should be warned first or just aborted. This is determine by either the WARN command or through the WARN option of the IDLE command.

If the idle-session does not match any of the IDLE patterns, then it is checked against the GLOBAL idleness setting. If the idle-session has been idle longer than the GLOBAL setting, then it is either issued a warning or it is aborted.

Review the files KNOCKOUT.DATA.LPSTOOLS and KNOCKOUT.JOB.LPSTOOLS to see how a script and job are put together for KNOCKOUT. Also, see the examples in this section. KNOCKOUT can support up to 40 separate IDLE patterns and up to 1,023 excluded logical devices. An idleness limit of 0 or 32767 means: infinite. Don't ever abort matching sessions for idleness!

A session is considered "idle" if it has not used any processor time *anywhere* in its process tree since the last time it was checked. Whenever a change in CPU utilization is noticed, all flags and counters associated with a given session are updated. The implication of the *warn-then-abort* policy is that a "warned" session must be idle for twice the amount specified by the matching IDLE pattern before it will be aborted.

Capabilities

Program capabilities required include IA, BA, DS, PM and PH. User SM capability is required to run KNOCKOUT. Also, the ABORTJOB command must be ALLOWed for the session or job running KNOCKOUT.

Usage

KNOCKOUT can be run either by the supplied UDC or with a fully-qualified RUN statement.

- UDC

  ```
  :KNOCKOUT
  ```

- RUN

  ```
  :RUN KNOCKOUT.PUB.LPSTOOLS
  ```

To view a sample job stream, refer to **knockout.job.lpstools**.

Command Summary

The following list provides a summary description of KNOCKOUT commands that you can use to quickly locate the command that suits the task at hand. Detailed information on each command is provided in the next section.

NOTE Portions of the command codes are printed in uppercase to denote the part of the command that KNOCKOUT requires in order to distinguish one command from another. However, the commands themselves are not case-sensitive.

Table 10.1 *KNOCKOUT commands*

Command Code	Description
END	Quits reading commands
EXCLUDE	Excludes logical devices
Exit	Terminates KNOCKOUT
HELP	Invokes KNOCKOUT help
IDLE	Sets the time before killing session
LOOP	Controls delay at the top of loop
REPORT	Displays the current settings
SET/REset	Enables or disables KNOCKOUT options

Command Definitions

This section contains a detailed description of each KNOCKOUT command. Syntax is provided for some of the commands.

END

If an END command is found, no further commands will be read. An end-of-file is interpreted as an END command.

EXCLUDE

This command has the following syntax:

```
EXCLUDE LDEVS [ldev [,ldev...]] [console]
```

EXCLUDE specifies the ldevs that should not be aborted. If the operator's console is moved with the MPE ":CONSOLE" command, specifying "EXCLUDE CONSOLE" causes KNOCKOUT to find the system console at the top of each loop.

NOTE The EXCLUDE ldev is checked before *any* IDLE rule is applied. This means that if you say "EXCLUDE CONSOLE", you can never have a rule that usefully checks the console.

Exit

Exit terminates KNOCKOUT.

HELP

HELP invokes KNOCKOUT Help.

Additional help is available on the following topics:

OVERVIEW	Overview of KNOCKOUT
EXAMPLES	Examples of KNOCKOUT commands
WARN, NOWARN	KNOCKOUT concept of warning idle users
GLOBAL	an option of the SET command
LOGWARN, LOGABORT, NOLOG	logging options of SET command

For information about a specific topic or command, enter "HELP" followed by the name of the topic or command. For example:

```
HELP IDLE
```

or

```
HELP LOGWARN
```

IDLE

This command has the following syntax:

```
IDLE jobname, user.account = seconds [options...]
```

where `options` are:

```
WARN | NOWARN | LDEV = ldev [/ldev] | LDEV = CONSOLE
```

Up to 40 IDLE commands may be issued in the KNOCKOUT job.

Note that **jobname**, **user**, and **account** specifications may be exact or may use the usual MPE wildcard (i.e.: **@,** or **MARY??#**).

If NOWARN is used, then sessions matching this pattern will not be warned before being aborted. If WARN is used, then the session will be warned before being aborted.

NOTE WARN results in a session getting twice as much idle time as a NOWARN session.

If neither WARN nor NOWARN are specified, the most recent SET WARN or SET NOWARN value is used for this IDLE pattern. The KNOCKOUT default is SET WARN.

The LDEV option allows this IDLE rule to be restricted to a specified ldev or range of ldevs. The "LDEV = CONSOLE" means that the console *at the time of checking*, which differs from saying "LDEV = 20".

For example:

```
IDLE @,FIELD.SUPPORT=10 NOWARN

IDLE MARY,@.ACCTING=NEVER

    ! above affects sessions logged on with jobname MARY,

    ! any user name, and account name ACCTNG

IDLE @, @.@ = 60 LDEV = CONSOLE

    ! above logs off the system console after one minute of idleness
```

LOOP

This command controls the number of seconds KNOCKOUT will wait at the top of its loop. Setting this number to a very small value will waste system resources. The LOOP value defaults to 60 seconds if KNOCKOUT is run from batch, and 10 seconds if run interactively.

On an HP3000/925 with 14 sessions, KNOCKOUT takes about one tenth of a second of CPU time each time it LOOPs. SHOT can be used to watch the CPU usage of KNOCKOUT.

NOTE Running KNOCKOUT interactively is recommended for testing only.

REPORT

This command displays all of the current settings for the current KNOCKOUT process. **Report** is a handy device for debugging a new KNOCKOUT job. See the examples for an illustration of the layout on this report.

SET | REset

The SET/RESET commands are used to specify the following options:

GLOBAL #	The GLOBAL option specifies the number of seconds users who do not match any of the IDLE patterns are allowed to be idle before being (optionally) warned and aborted. If SET NOWARN is in effect, then such users will not be warned before being aborted. The initial value of the GLOBAL timeout is zero (0), which disables it.
LOGAbort (Set by default)	When KNOCKOUT aborts a user and if LOGABORT is true, then it will send a message to the system operator reporting the knockout. This option can be canceled by using NOLOG.
LOGWarn (Set by default)	When KNOCKOUT warns a user and if LOGWARN is true, then it will send a message to the system operator reporting the warning. This option can be canceled by using NOLOG.
NOLOG	NOLOG turns off LOGABORT and LOGWARN.
NOWARN	Tells KNOCKOUT to not warn a user when they are about to be aborted for idleness. **NOTE:** A "warned" user gets twice the idle limit, but a "nowarned" user gets only the specified idle limit.
WARN (Set by default)	WARN causes KNOCKOUT to warn users before they are aborted for idleness.

TOOLBOX STANDARDS

The ToolBox collections from Lund Performance Solutions have a uniform user interface. As a result, in addition to the commands specific to each Toolbox tool, most tools allow the commands described in "TOOLBOX STANDARDS" on page 311.

KNOCKOUT Examples

In the three examples that follow, we cover a set of KNOCKOUT commands and briefly discuss their results. The remaining examples illustrate other common applications.

Figure 10.1 demonstrates that since no SET WARN or SET NOWARN commands were used, and no IDLE commands used the NOWARN keyword, all of the sessions will be warned before being aborted.

The session JOHN,MANAGER.SYS has an idleness limit of 10 (i.e., he is warned after 10 seconds, then aborted after 10 more) even though the **jobname**, **user.account** also matches the second IDLE command!

```
IDLE JOHN,@.@ = 10
IDLE @,MANAGER.SYS = 20 WARN
IDLE @,@.@ = 99
```

Figure 10.1 *Script Example*

Figure 10.2 works just like the prior example, but users are not warned. Instead, they are simply aborted. When JOHN,MANAGERS.SYS is idle for 10 seconds, he is aborted. The SET NOWARN is useful at sites where users have learned to respond to the idleness warning by hitting return a few times.

```
SET NOWARN
IDLE JOHN,@.@ = 10
IDLE @,MANAGER.SYS = 20
IDLE @,@.@ = 99
```

Figure 10.2 *Script Example*

In Figure 10.3, when JOHN,MANAGER.SYS is idle for 10 seconds, he will be aborted without warning. When FAY, MANAGER.SYS is idle for 20 seconds, she will be warned, and then aborted after 20 more idle seconds. MARY,MANAGER.SYS will be aborted without warning after 5 seconds. What about users who match none of the idle patterns? They will be warned after 34 seconds (the GLOBAL value) and aborted after 34 more seconds of idleness.

```
SET NOWARN
IDLE JOHN,@.@ = 10
IDLE @,MANAGER.SYS = 20 WARN
IDLE MARY,@.S@ = 5
SET WARN
SET GLOBAL 34
```

Figure 10.3 *Script Example*

Figure 10.4 illustrates a KNOCKOUT warning message and log off procedure.

```
:FROM/J31 MANAGER.SYS/ 14:55:45 WARNING: IDLE limit exceeded...session
FROM/J31 MANAGER.SYS/ will be aborted soon unless usage resumes.
FROM/J31 MANAGER.SYS/ 14:55:57 IDLE limit exceeded...aborting session.

SOFTWARE ABORT   (FSERR 32)
SESSION aborted by system management. (CIERR 6027)
CPU=45. Connect=43. WED, SEP 5, 2001,  2:55 PM.


<Your 'VT-MGR' connection has terminated>
```

Figure 10.4 *KNOCKOUT Warning Message*

KNOCKOUT Error Messages

In the error messages that follow, "xxxx" refers to a number that is filled in at runtime by KNOCKOUT.

Table 10.2 *KNOCKOUT Error Messages*

Message	Cause	Action
Bad acctname pattern.	User entered a pattern for a acctname which KNOCKOUT does not understand.	Review acctname pattern, it should follow conventions defined by Hewlett-Packard's JOB command.
Bad jobname pattern.	User entered a pattern for a jobname which KNOCKOUT does not understand.	Review jobname pattern, it should follow conventions defined by Hewlett-Packard's JOB command.
Bad username pattern.	User entered a pattern for a username which KNOCKOUT does not understand.	Review username pattern, it should follow conventions defined by Hewlett-Packard's JOB command.

Message	Cause	Action
The maximum number of IDLE patterns "xxxx" has already been defined.	User has defined more IDLE patterns than KNOCKOUT supports.	Rethink, consolidate IDLE patterns into fewer IDLE commands.

11

THE MAGNET TOOL

MAGNET scans a set of files for the presence of one or more text strings. Many options allow for flexible pattern description. Moreover, MAGNET supports a very powerful fileset specification syntax so you can qualify your file searches for maximum efficiency.

Capabilities

Program capabilities required include IA, BA, DS and PH. No special user capabilities are required to run MAGNET.

Usage

MAGNET can be run from either the supplied UDC or from a fully-qualified RUN statement.

- UDC

```
:MAGNET "-f@.c -c 'main' 'define'"

:MAGNET "<fileset> [options] <text string>" [parm=#]
```

- RUN

```
:RUN MAGNET.PUB.LPSTOOLS;INFO="<fileset> [options]
      <text string>";[parm=#]
```

Run MAGNET with **parm=1** to suppress paging.

Options are specified in a list separated by one or more spaces. Most options start with a hypen (-) followed by an option character, followed by an optional string. The option character is not case-sensitive. Filesets may be specified using a LISTF style format or by using MAGNET's extended fileset specification syntax. MAGNET scans normal ASCII flat files and QEDIT (filecode=111) work files by default. However, MAGNET can scan any other file type (using the **-d** option) except for privileged files.

The output from MAGNET varies depending on user-selected options. The default output consists of a filename followed by an asterisk if the text string was found in the file.

Output example:

```
Wolf:/LPSTOOLS/PUB:  magnet "-f@.c -c 'main' 'define'"
MAGNET [2.26] - LPS Toolbox [A.09f]         (c) 1995 Lund Performance Solutions

This product is licensed to: ImageStats Demo

CAPTURE.C.LPSTOOLS          *
CHKWILD.C.LPSTOOLS          * *
TESTCHRO.C.LPSTOOLS         *
TESTCW.C.LPSTOOLS           * *
TESTFS.C.LPSTOOLS           * *
TESTGFS.C.LPSTOOLS          *

Scanned 6 files in 0.813 seconds
Wolf:/LPSTOOLS/PUB: _
```

Figure 11.1 *MAGNET Output*

If a "**-l**" option is specified, then the entire matching line is displayed as well as the name of the line in which it was found.

NOTE MAGNET modifies the access date of files that it has scanned. The text string is always the last item specified.

```
<text string>
        = <ASCII character, no blanks> | <ASCII characters, blanks ok> [<TEXT STRING>]
```

Figure 11.2 *Text String Definition*

Examples

Multiple Word Searches

You would enter the following to find all occurrences of the words "one", "two", and "three".

NOTE Placing single quotes around each word is required for multiple word searches.

```
:magnet "-f@ -l 'one' 'two' 'three'"
```

Single Word Searches

You would enter the following to find all occurrences of the word "five".

NOTE Single word searches do not require single quotes.

```
:magnet "-f@ -l five"
:magnet "-f@ -L 'five'"
```

Single Word Searches For Combined Words

You would enter the following to find all occurrences of the combined words "Lund Performance Solutions".

NOTE Single quotes are required when blanks are used to separate words within the desired string.

```
:magnet "-f@ -l 'Lund Performance Solutions'"
```

The file-set expression

```
<file set expression> ::= <file set descriptor>    (or:  ^indirectfile)
                          [ [ <set operator> <file set descriptor>] ...]

<set operator> ::= "+" | "-"

<file set descriptor> ::= <generic name>
                          [ [ "," <filter> ] ...]

<generic name> ::= { a file name, including wildcards, as defined in
                     the MPE "LISTF" command }

<filter> ::=    "CREDATE" <relop> <date>
              |  "MODDATE" <relop> <date>
              |  "ACCDATE" <relop> <date>
              |  "CODE"    <relop> <numeric value>
              |  "CODE"    <relop> <mnemonic>
              |  "LABELS"  <relop> <numeric value>
              |  "LIMIT"   <relop> <numeric value>
              |  "EOF"     <relop> <numeric value>
              |  "SECTORS" <relop> <numeric value>
              |  "BF"      <relop> <numeric value>
              |  "TEMP"
              |  "ASCII"
              |  "BINARY"
              |  "FIXED"
              |  "VARIABLE"
              |  "UNDEFINED"
              |  "CCTL"    <onoroff>
              |  "RIO"     <onoroff>
              |  "MSG"     <onoroff>
              |  "CIR"     <onoroff>

<onoroff> ::= "=" { "ON" | "OFF" }

<relop> ::= "=" | "<>" | "<" | "<=" | ">=" | ">"

<date> ::=    { a date in the format yy/mm/dd or yymmdd}
            |  "TODAY"
```

Figure 11.3 *MAGNET Extended Fileset Syntax*

NOTE All literals are case insensitive.

Some examples:

Example 1 shows all the files in the group pub.

```
@.pub
```

Example 2 shows all the files in this account, but not in the group pub.

```
@.@ - @.pub
```

Example 3 shows all files in this account, created after January 15 1991.

```
@.@,credate>910115
```

Example 4 shows all native mode program files in this account, larger than 1000 sectors.

```
@.@,code=nmprg,eof>1000
```

For further information, you may wish to refer to the appendices containing filecode lists (Appendix B) and the LISTF wildcard syntax (Appendix C).

Options Summary

MAGNET is a single-command based tool that uses multiple options to achieve the desired result. Options are briefly described below. Most options may be preceded with "no" to deselect the action (i.e., **-noascii**). Complete descriptions are provided in the next section.

Table 11.1 *MAGNET Options*

Option Name	Description
-a	All words must be present
-align	Aligns file-group-account names into columns
-ascii7	Defines printable as ascii 32..126 (see -printable)
-b	Saves output fileset as a permanent file. (see -o)
-binary	All files (default is ASCII and Qedit only)
-c	Ignores Case
-cctl	Uses carriage control when the -p option is used
-context	Displays two lines before and two lines after every match
-contexta #	Displays # lines After every match
-contextb #	Displays # lines Before every match
-count	Displays # matches found in each file
-cystop	Causes ^Y (CTRL-Y) to immediately terminate MAGNET
-d	Synonym for -binary

Option Name	Description
-dash	Prints line of dashes after 'hit' files.
-e	Shows line numbers in Editor format (12345.678)
-f [fileset]	Specifies set of files to search.
-fcode	Ignores filecodes (normally excludes most filecodes)
-g	Uses Gold Nugget KLONDIKE to prefetch
-h	Prints a page Header on each page
-help	Displays the help file
-invert	Inverts final 'interesting' decision
-l	Lists each occurrence of search string found
-lockword ccc	Uses the specified lockword (-lockword fakelock)
-lr	Scans for matches 'left-to-right' only
-m	Outputs set should have filenames in MPE format (-o)
-maxlines #	Limits number of matching lines shown per file to #
-maxread #	Limits number of records read per file to #
-maxrecords #	Limits number of records read per file to #
-maxtotal #	Stops scanning after that many total hits
-msg	Allow MSG files to be read
-msgcopy	Uses COPY mode when opening MSG file (on by default)
-msgwait	Allows waiting on empty MSG files (off by default)
-n	Shows line numbers for each line
-never...	Tells MAGNET to ignore subsequent occurrences of five options
-numbers	Shows online line number of matching lines, not the text
-o outputset	Name of file to save the list of matching files in
-olddates	Tells MAGNET to restore the old access date of files
-origin	Tells MAGNET to report origin of all option lines

Option Name	Description
-p device	Name of device for printed output (i.e.: -p LP)
-paging	Turns on paging for interactive output
-pascal	Adds underscore (_) to definition of a word (-w)
-prefetch	Use Gold Nugget KLONDIKE to prefetch (= -g)
-printable	Causes non-printables to display as dot (.) (default)
-q	Quiet. Suppresses progress messages (unless -h or -l)
-qedit	Search QEDIT files as well as ASCII files (default)
-qeditonly	Search only QEDIT files
-quote c	Use the specified character as a quoted string delimiter
-s specials	Defines extra characters that may be in 'word' (-w)
-spl	Adds apostrophe (') to definition of a word (-w)
-splash	Adds apostrophe (') and underscore (_) to word (-w)
-t text	Specifies text string to look for. Multiple strings may be specified.
-t "text"	
text	
"text"	
'text with spaces'	
-tellop	Sends message for each matching file to console
-timestamp	Prints a timestamp at end
-tree	Recursively search down any HFS directories we see
-u	Prevents screen-enhancements for search strings with -l
-w	Whole Word search only. -t fo -w won't match 'for'
-summary	Summary of results
-y	
-z #	Debugging mask set to integer # value
-[	Strings must start in column 1 to match

Option Name	Description
-(###	Start search in column ### (ignore prior columns)
-) ###	Right-most column search string may start in
-72	Limits searches to the first 72-bytes in a record

NOTE Single letter options are not case-sensitive. The options flagged with "(-w)" above mean that specifying them implies -w as well.

Options Definitions

Following is a detailed description of each of the MAGNET options. Some options include syntax.

-a

Specifies that a file will only be flagged if it contains all the search strings. The default is that the file will be flagged if it contains one or more of the search strings.

DEFAULT: -NOa

-align

Tells MAGNET that filenames should be displayed in a manner where the group parts line up (align), and the account parts line up. (i.e., "FOO .A .SYS ")

-NOalign tells MAGNET to display filenames in a compact manner (i.e., "FOO.A.SYS")

DEFAULT: -NOalign

-ascii7

Specifies that any ASCII characters outside of the range 32..126 should be displayed as dots (.). -NOascii will allow all characters in the range 32..255 to be displayed without change.

DEFAULT: -NOascii7

-b

Specifies that the output set must be saved as a permanent file. The default is to save it as a temporary file. (See also -o).

DEFAULT: -NOb

-binary

Specifies that all files must be searched. MAGNET normally defaults to searching only ASCII and QEDIT files (-qedit).

DEFAULT: -NOd (ASCII and QEDIT only)

-c

Tells MAGNET to ignore the Case of alphabetic characters while searching.

DEFAULT: -NOc (searching is case sensitive).

-cctl

Tells MAGNET to use Carriage Control when writing to -p device.

DEFAULT: -cctl (if using -p, else -NOcctl)

-context

Tells MAGNET to display 2 lines before every matched line, and 2 lines after every matched line. Equivalent to saying: -contextb 2 -contexta 2

DEFAULT: -NOcontext

-contexta

Tells MAGNET to display # lines After every matched line.

DEFAULT: -NOcontexta

-contextb

Tells MAGNET to display # lines Before every matched line.

DEFAULT: -NOcontextb

-count

Tells MAGNET to report number of matches found in each file (if the number is greater than 0).

NOTE not intended to be used with "-and"

-cystop

Tells MAGNET to use stop immediately and terminate when control-Y is hit.

DEFAULT: -NOcystop (control-Y will ask if you want to stop).

-d

Synonym for -binary (obsolete).

-dashes

Tells MAGNET to separate files with "hits" by a long dashed line. -dash implies -l. -NOdash does not imply -NOl.

-e

Tells MAGNET to convert line numbers in printed record to an Editor/3000 compatible format.

DEFAULT: -NOe (record numbers are not converted)

-f [fileset]

Specifies set of files to search. The fileset syntax is the same as for the :LISTF command. Multiple -f options are allowed. If fileset is omitted, "@" is assumed.

DEFAULT: -f@

-fcode

Normally, unless -binary was specified, MAGNET only looks at ASCII files (and, usually QEDIT files), but it will usually exclude ASCII files whose filecode isn't 0, 8, EDTCT, or OUTSP. -fcode tells MAGNET to skip the filecode check.

DEFAULT: -NOfcode (which means: exclude some files by filecode)

-h

Tells MAGNET to print a page header on each page.

DEFAULT: -NOh (no page header is printed)

-help

Tells MAGNET to print this display this entire help file.

-invert

Tells MAGNET to invert the final "interesting" choice.

-l

Tells MAGNET to list all occurrences of the search string. If -NOl is in effect, lines are not listed (instead, files that contain one or more occurrences of the search string(s) are flagged with an asterisk).

DEFAULT: -NOl (Lines are not listed)

NOTE If more than one string is to be searched and the -L option is in effect, searching is considerably slower.

-lockword ccc (or -NOlockword)

-lockword tells MAGNET to use the specified lockword when opening a file to search. If the file doesn't have a lockword, then MPE ignores the supplied lockword and opens the file anyway. If the file has a lockword which differs from the supplied lockword, the open request fails.

If -neverlockword is seen, then MAGNET ignores any subsequent -[no]lockword.

If -NOlockword is specified, and a file has a lockword, then MPE will prompt the user for a lockword at open time.

DEFAULT: -lockword fakelock (MPE/iX 5.0 and later)

DEFAULT: -NOlockword (MPE/iX 4.5 and earlier)

-m

Specifies that the output set should contain file names in MPE format. (See also -o).

DEFAULT: -NOm

-maxlines # (or -NOmaxlines)

Tells MAGNET to stop scanning a file after "#" matches are found. This is only meaningful in conjunction with "-l".

A value of 0 (or -NOmaxlines) means: don't limit the number of lines found.

DEFAULT: -NOmaxlines

-maxreads # (or -NOmaxreads)

-maxrecords # (or -NOmaxrecords)

Tells MAGNET to stop scanning a file after "#" records are examined. (MAXRECORDS is a synonym for MAXREAD).

A value of 0 (or -NOmaxrecords) means: don't limit the number of records examined.

DEFAULT: -NOmaxrecords

-maxtotal # (or -NOmaxtotal)

Tells MAGNET to stop scanning and terminate after a total of # hits have been found. It doesn't matter how many files were scanned, or how many had hits, or how many hits per file were found ... as long as the specified *total* number of hits were found, MAGNET will stop.

A value of 0 (or -NOmaxtotal) means: don't limit the number of hits.

DEFAULT: -NOmaxtotal

-msg

Tells MAGNET to read MSG files (normally skipped over). If -msgcopy is true (which it is, by default), "COPY" mode is used in reading the file, so the records don't "disappear".

NOTE Prior to MPE/iX 5.0, the records in a MSG file read in COPY mode are not neccessarily in chronological order.

DEFAULT: -NOmsg

-msgcopy

Tells MAGNET to use COPY mode when opening a MSG file. If -msg is set, and -NOmsgcopy is set, then records read from a message file will "disappear" from the file.

DEFAULT: -msgcopy

-msgwait

Tells MAGNET to wait when reading from an empty MSG file. Only has effect when -msg and -NOmsgcopy.

DEFAULT: -NOmsgwait

-n

Tells MAGNET to print line numbers for each listed line.

DEFAULT: -NOn (no line numbers are shown)

-never...

The five "-never..." options tell MAGNET to ignore subsequent occurrences of five options. The "never" option would typically be used in a global MAGNET configuration file. The options are:

-NEVERLOCKWORD	ignore all subsequent -[no]LOCKWORD
-NEVEROLDDATE	ignore all subsequent -[no]OLDDATE
-NEVERPREFETCH	ignore all subsequent -[no]PREFETCH
-NEVERTELLOP	ignore all subsequent -[no]TELLOP
-NEVERZ	ignore all subsequent -Z #

For example, if you never want to restore old access dates to files (and don't want any users overriding you!), add the following to MAGNET.CFG.LPSTOOLS:

```
-noolddate -neverolddate
```

-numbers

Tells MAGNET to only list line numbers (or sequence numbers if -e is used too) instead of the complete text of matching lines.

DEFAULT: -NOnumbers

-o outputset

Specifies that the file names of the files that "match" the search string(s) be saved in a file, named outputset.

DEFAULT: -NOo

-olddates

Tells MAGNET to try to restore the old access date for files that it touches. -NOolddates tells MAGNET to not try to reset the access dates of any files. -neverolddate tells MAGNET to ignore any subsequent -[no]olddates.

DEFAULT: -olddates

-origin

(Internal debugging option)

Tells MAGNET to display the origin of the current (and subsequent) option string(s) being parsed, as well as a copy of the string. If the string came from the MAGNET.CFG file in the logon account, it will be prefaced with "[cfg]". If the string came from the MAGNET.CFG.LPSTOOLS file, it will be prefaced with "[cfg]". If the string came from the INFO string, it will be prefaced with "[INFO]". If MAGNET is run with PARM=2, an implied -origin is done at the start.

DEFAULT: -NOorigin

-p device

Tells MAGNET the name of the device (e.g.: LP or 113) on which output is to be printed. When the -p option is given without a device, then LP is used. -NOp means: use $STDLIST.

DEFAULT: -NOp (output is printed on $STDLIST)

LPSLP is the formal file name opened when -p is specified. The -p directs most output to LPSLP, but some progress information will still come to $STDLIST for interactive users.

NOTE If a file equation for LPSLP exists which specifies the device name (i.e.: FILE LPSLP;DEV=LP;CCTL), then the device name specified by the -p is ignored by MPE. A file equate for LPSLP which will allow a -p to specify a device AND which sets the output priority to 13 could be done: FILE LPSLP; DEV= ,13

-paging

Tells MAGNET to paginate interactive output. See also PARM=1

DEFAULT: -paging (for interactive, -NOpaging for batch)

parm=1

This option is used to suppress paging.

NOTE No dash is used for this option.

parm=2

Does an implied **-origin**.

-pascal

Add underscore (_) to list of characters that make up words. This is a shorthand of saying: -w -s_

DEFAULT: -NOpascal

-prefetch (or: -g)

Tells MAGNET to use the KLONDIKE tool to prefetch files that will be scanned. This may yield a substantial performance improvement for very large files. If -neverprefetch is seen, then MAGNET ignores any subsequent -[no]g or -[no]prefetch.

DEFAULT: -NOprefetch

-printable

Tells MAGNET to display non-printable characters as dots (.). (A non-printable character is one whose ASCII code is 0..31, or 127..255.)

DEFAULT: -printable

-q

Tells MAGNET to be Quiet. This suppresses progress messages and (if -H and -L are not used) produces only a list (one per line) of file names of files that have one (or more) occurrences of the specified string(s). This output is appropriate for directing to a file and manipulating with an editor.

DEFAULT: -NOq (not quiet)

-qedit

Tells MAGNET that it is allowed to treat (and read) QEDIT files as though they were ASCII files. NOQEDIT tells MAGNET to skip reading QEDIT files.

DEFAULT: -qedit

-qeditonly

Tells MAGNET to only select QEDIT files.

DEFAULT: -noqeditonly

-quote c

(where "c" is any printable character)

Tells MAGNET the character you will be using as a "quote" to enclose tokens. For example, to search for the text "'" (composed of the 3 characters quote, apostrophe, quote), and also search for the three letter word FOO, you could say: -quote \ \"'"\ \FOO\

DEFAULT: -quote '

-s specials

Specifies a string of characters to be considered as non-terminators when a WHOLE-WORD search is done (see -W option). Multiple -s strings are concatenated, along with any -pascal, -spl, and -splash.

NOTE -s implies -w.

Example: -s_ (includes "_" in words)

DEFAULT: -NOs (no special characters for words defined)

-spl

Add apostrophe (') to list of characters that make up words. This is shorthand for saying: -w -s'

DEFAULT: -NOspl

-splash

Add apostrophe (') and underscore (_) to list of characters that make up words. This is shorthand for saying: -w -s_'

DEFAULT: -NOsplash

-t text or -t "text" or text or "text" or 'text with spaces'

- alternate ways of specifying search text

Specifies the text to search for. The search text need not be enclosed in quotes. Quotes are necessary if any blanks or special characters are part of the text.

DEFAULT: (no text is specified, MAGNET will report an error)

More than one text can be given by repeating the text option. There is no default value, therefore at least one text option must be given.

Examples:

-f@.pub.sys Fred	Looks for: ...Fred... in @.pub.sys
"Fred Fulton"	Looks for: ...Fred Fulton... in logon group
c@a	Looks for: ...c@a... in logon group
	(note: "@" is not a wildcard here)

-tellop

Tells MAGNET to send a message to the operator console announcing each matching file found. If -nevertellop is seen, then MAGNET ignores any subsequent -[no]tellop.

DEFAULT: -NOtellop

-timestamp

Tells MAGNET to print a timestamp (date & time) at the end of its output.

-tree

Tells MAGNET to recursively search down any hierarchical directories that the initial fileset might include.

-NOtree tells MAGNET to ignore HFS directories.

DEFAULT: -tree

-u

Tells MAGNET not to enhance the found data in lines printed to the screen.

DEFAULT: -NOu (i.e.: enhance)

-w

Tells MAGNET to search for Whole words only. If this option is in effect, a string will only match if it is preceded and followed by a separator. A separator is any non-alphanumeric character, that is not given in the -s option. Using any of the options -s, -pascal, -spl, or -splash implies -w.

DEFAULT: -NOw (any string matches)

-summary

-y

Tells MAGNET to produce a short summary of the search results. ("-summary" is a synonym for "-y")

DEFAULT: -NOy

Example summary:

```
Scanned 31 files in 2.914 seconds (found 1 matching files)
Search string: magnet.
```

NOTE If -p is used with -y, the summary is sent to both the terminal and the -p device.

If -p is used with -y, the summary is sent to both the terminal and the -p device.

-z

Sets some internal debugging flags to the specified value. If -neverz is seen, then MAGNET ignores any subsequent -[no]z

DEFAULT: -NOz

-[

Tells MAGNET to only check for a match against strings in column #1.

DEFAULT: Search all columns.

-(

Tells MAGNET to start the search at column ###. The leftmost column of a file is 1. Example: -(9

-)

Tells MAGNET the right-most column that text is allowed to start in. Example: -(4 -)5 -t begin will search for "begin" starting in columns 4 or 5.

-72

Shorthand method of saying -)72

TOOLBOX STANDARDS

The ToolBox collections from Lund Performance Solutions have a uniform user interface. As a result, in addition to the commands specific to each Toolbox tool, most tools allow the commands described in “TOOLBOX STANDARDS” on page 311.

MAGNET Examples

Example 1:

```
:run magnet.pub.lpstools;info="-f@.source foo -s_ -W -C"
```

The above command line do the following: ignores the case of characters, searches for "whole words" only (so a line with "food" will not match), adds underscore to the list of characters that can be in a word (so "foo_foo" will not match, but foo'fum will). It also searches all ASCII/QEDIT files in the SOURCE group in the logon account. The -C options tells MAGNET to internally upshifts all text (and the string "foo") before checking for matches. Note that "-w" is redundant, as "-s" implies "-w".

Because -L was not specified, MAGNET will simply produce a list of files that include one (or more) occurrences of "foo".

Example 2:

```
:run magnet.pub.lpstools;info="-f@.source-@q.source-bnf.source -
tFOO -L"
```

Search all ASCII/QEDIT files in the SOURCE group except the file BNF.SOURCE and except any file ending in "Q". List each line that contains "FOO".

Example 3:

```
:run magnet.pub.lpstools;info="-f@.source -(4 -)4 -tbegin -L -
binary"
```

Search all files in the SOURCE group for "begin" starting only in column 4. Each line found will be listed.

Example 4:

```
:run magnet.pub.lpstools;info="-f@.source 'foo' -noqedit";parm=1
```

Will search all ASCII files (no binary, no QEDIT) in the SOURCE group. However, after displaying 22 lines of matches (assuming there are that many matches) MAGNET will not pause. This is particularly useful when scanning large filesets.

The following is an example of typical MAGNET output on the text string, "standard". MAGNET highlights each occurrence of the specified text string in context of the entire line in which it is found. After you enter the command, the resulting output is shown in Figure 11.4.

```
Wolf:/LPSTOOLS/PUB:  magnet "-f@.help.lpstools-standard.help.lpstools -c -l stan
dard"
MAGNET [2.26] - LPS Toolbox [A.09f]         (c) 1995 Lund Performance Solutions

This product is licensed to: ImageStats Demo

CAPTURE.HELP.LPSTOOLS
disc file.  The terminal must obey standard Hewlett-Packard terminal
   to standard for CAPTURE to work, but that would otherwise be ignored
ETC.HELP.LPSTOOLS
ETC allows standard MPE-LISTF style pattern matching, where
FASTLIB.HELP.LPSTOOLS
FASTLIB is a library of fast replacements for the standard intrinsics:
The five intrinsics are "plug-compatible" with the standard intrinsics.
The FASTLIB intrinsics differ from the standard intrinsics in only two
ways:  (1) they are much faster; and (2) if a standard intrinsic wants
circumstances as the standard intrinsics, but without the same abort
KLONDIKE.HELP.LPSTOOLS
SET sets the specified options to "true".  RESET sets the standard
SET sets the specified options to "true".  RESET sets the standard
SET sets the specified options to "true".  RESET sets the standard
MODIFY.HELP.LPSTOOLS
          "standard" HP26xx escape sequences, and some of them
REDWOOD.HELP.LPSTOOLS
the Toolbox standard command: SET COPYLP may be used instead of
SHOT.HELP.LPSTOOLS
   standard MPE wildcards.
      a process' standard signal, message, and interrupt ports.

Scanned 25 files (15840 lines, 15 hits) in 1.506 seconds
Wolf:/LPSTOOLS/PUB:
```

Figure 11.4 *MAGNET Output on a Text String Search*

MAGNET Error Messages

Each MAGNET error message is described in the following table.

Table 11.2 *MAGNET Error Messages*

Message	Cause	Action
Invalid file set	Improper use of fileset specification	Review "specifying filesets" in the MAGNET chapter (see “The file-set expression” on page 116).

Message	Cause	Action
Unknown option	An unknown option character was detected by MAGNET.	Valid option character are: a, b, c, d, e, f, g, h, l, m, n, o, p, q, s, t, w, [, (,).

12

THE MODA TOOL

MODA is designed to simplify the tasks of creating or modifying HP3000 accounts, groups and users. MODA also has two commands that are used for cloning accounts and sync-ing account structures.

Operation

If you have ever had to modify the capabilities on an account via the ALTACCT command you will instantly appreciate MODA's power. Normally, when you need to modify an attribute of an account, you need to type your changes plus all of the attributes that you don't want to change. With MODA, all you have to key in are your changes.

A typical example of how MODA simplifies routine tasks is illustrated by the steps involved in adding SM to an account. Without MODA, you would have to re-key all of the capabilities that the account currently has and then add SM to the list. With MODA, you simply add SM to the list. This concept applies to any modifiable attribute of an account, group, or user. Further, the same holds true for creating accounts, groups, and users.

The main mechanism used to make modifying and creating accounts so easy is MODA's line-editor, which is called MODIFY. Whenever you want to modify an object, MODA displays a fully-specified MPE ALT command on the screen which you edit to suit your needs. Using MODIFY's editing commands (see "MPE File Codes" on page 289) you simply change the line to look the way you want it to, and then press **Enter** to execute the command. A brief summary of MODIFY's commands is provided later in this chapter.

In addition to these time-saving commands, MODA also has commands for cloning accounts and sync-ing accounts. MODA's CLONEACCT command can be used for duplicating all aspects of an account's structure into a new account structure. This functionality is ideal for facilitating version control. MODA's COPYACCT command is used to bring two accounts in sync with regard to account structure. For example, if you wanted account A to have all of the groups and users that account B has, you would simply use MODA's COPYACCT command to do this.

MODA uses the MPE COMMAND intrinsic for maximum compatibility with future releases of MPE/iX.

Capabilities

The Account, NewAccount, and CLONExxx commands require the user to have SM capability. All other commands require AM or SM capability.

Usage

MODA can be run from either the supplied UDC or from a fully-qualified RUN statement.

- UDC

  ```
  :MODA [<command> [,<command>] [...]]
  ```

- RUN

  ```
  :RUN MODA.PUB.LPSTOOLS;INFO="[<command> <command>...]"
  ```

MODA is typically run without parameters.

Command Summary

The following list provides a summary description of MODA commands, which can be used to quickly locate the command that suits the task at hand. Detailed information on each command is provided in the next section.

NOTE Most MODA commands can be abbreviated to one or two letters. Command names are shown with mixed upper case and lower case letters, where the minimum abbreviation is indicated by the upper case letters.

Table 12.1 *MODA Commands*

Command Code	Description
Account	Displays account attributes
CLONEaccount	Clones account structure to new account
COPYaccount	Copies attributes to existing account
Exit	Terminates MODA
Group	Displays group attributes
HELP	Invokes MODA help
NewAccount	Edits attributes for a new account

Command Code	Description
NewGroup	Edits attributes for a new group
NewUser	Edits attributes for a new user
User	Displays attributes for a user

Command Definitions

Following is a detailed description of each MODA command.

NOTE Brackets ([]) are used to indicate optional parameters. Angle brackets (<>) are used to denote a parameter to be filled in.

Account

This command has the following syntax:

```
Account <name> [ <subset> ]
```

The ACCOUNT command displays the attributes for the specified account. These attributes may be edited as required. If the account does not exist, an error is reported. Using the ACCOUNT command requires SM capability.

subset is one (or more) of the following, optionally separated by blanks:

```
ACCess CAPability CONnect CPU FILES HOME LOCattr MAXPRI PASSword
```

When a **subset** is supplied, only those fields will be displayed for editing. If no **subset** is found, then all of the attributes (fields) for the **account**, **group** and/or **user** will be displayed. The keywords for **subset** may be abbreviated to just the uppercase portion shown above.

For example, to change just the capabilities for account "FOO", you could enter:

```
:MODA A FOO CAP
```

CLONEaccount

This command has the following syntax:

```
CLONEaccount <oldacct> <newacct> [ ACAP = ALL ] [Quiet] [NOPASS]
                    [ GCAP = ALL ] [ UCAP = ALL ]
```

The CLONEACCOUNT command replicates (clones) the accounting structure of an existing account into a new account. The **newacct** will be created with the same attributes as the **oldacct**.

For every user and group within **oldacct**, a duplicate will be created within **newacct**. The groups and users created within **newacct** will have the same attributes as **oldacct**.

Note that UDC settings and Private Volume information is not replicated from **oldacct** to **newacct**.

The COPYACCOUNT and CLONEACCOUNT commands are very similar. The COPYACCOUNT command requires that **newacct** must already exist, while the CLONEACCOUNT command requires that **newacct** must NOT currently exist.

If the ACAP = ALL option is used, then the **newacct** will have all capabilities, rather than the set from **oldacct**.

If the GCAP = ALL option is used, then every group within **newacct** will have all available capabilities, rather than the set from the original groups from **oldacct**.

If the UCAP = ALL option is used, then every user within **newacct** will have all available capabilities, rather than the set from the original users from **oldacct**.

The NOPASS keyword will cause the **newacct** and all of its groups and users to be created without passwords.

The QUIET keyword will cause most of the information generated by the CLONEACCOUNT command to be suppressed.

COPYaccount

This command has the following syntax:

```
COPYaccount <oldacct> <newacct> [ ACAP = ALL ] [QUIET] [NOPASS]
            [CREATE] [ GCAP = ALL ] [ UCAP = ALL ]
```

The COPYACCOUNT command copies the attributes from **oldacct** to **newacct** for the account, and for every user and group within **oldacct**. Only those groups and users within **newacct** that also appear within **oldacct** will be affected unless the CREATE keyword is used.

The COPYACCOUNT and CLONEACCOUNT commands are very similar. The COPYACCOUNT command requires that **newacct** must already exist, while the CLONEACCOUNT command requires that **newacct** must NOT currently exist.

CREATE	Instructs the COPYACCOUNT command to create within **newacct** any group or user found within **oldacct** that does not currently exist.
ACAP = ALL	If this option is used, then the **newacct** will have all capabilities, rather than the set from **oldacct**.
GCAP = ALL	If this option is used, then every group within **newacct** will have all available capabilities, rather than the set from the original groups from **oldacct**.

UCAP = ALL	If this option is used, then every user within **newacct** will have all available capabilities, rather than the set from the original users from **oldacct**
NOPASS	This keyword will cause the passwords for **newacct** and all of its groups and users to be remain unchanged.
QUIET	This keyword will cause most of the information generated by the COPYACCOUNT command to be suppressed.

Exit or //

The Exit command terminates MODA. MODA also accepts // as a synonym for EXIT.

Group

This command has the following syntax:

```
Group <name> [ <subset> ]
```

The GROUP command will display the attributes for the specified group for editing. If the group does not exist, an error is reported.

You can edit attributes of groups in accounts other than your logon account by specifying the group as **group.account**. For example, to edit just the password for the group PUB in the account HPOFFICE, you could enter:

```
G PUB.HPOFFICE, PASS
```

HELP

The HELP command invokes the MODA help facility.

NewAccount

This command has the following syntax:

```
NewAccount [ <templateaccount> ] [ , CAP = ALL ]
```

The NEWACCOUNT command (minimum abbreviation NA) is used to edit the attributes for a new account. If NEWACCOUNT is entered without a **templateaccount**, then a default set of attributes is chosen for editing. If a **templateaccount** is entered, then MODA will fetch the attributes for that account and display them for editing. The **templateaccount** provides a simple way to create a new account with the same attributes as an existing account.

CAP=ALL tells MODA that the attributes for editing should have all possible capabilities.

Example 1: Creating a new account without a template, with "ordinary" capabilities:

```
NEWACCOUNT FOO
```

This will present text to be edited that looks like:

```
:NEWACCT FOO,MGR;CAP=AM,AL,GL,DI,UV,LG,PS,CS,ND,SF,BA,IA,MR,DS,PH;
ACCESS=(A,W,X,R,L:AC);MAXPRI=CS;LOCATTR=0;FILES=;CPU=;
CONNECT=;PASS=
```

Example 2: Creating a new account without a template, with all capabilities:

```
NA FOO,CAP=ALL
```

This will present text to be edited that looks like:

```
:NEWACCT FOO,MGR;CAP=SM,AM,AL,GL,DI,OP,CV,UV,LG,PS,NA,NM,CS,ND,
SF,BA,IA,PM,MR,DS,PH;ACCESS=(A,W,L,X:AC);
MAXPRI=CS;LOCATTR=0;FILES=;CPU=;CONNECT=;PASS=
```

Example 3: Creating a new account, FOO, that looks like the LPSTOOLS account:

```
NEWACCOUNT LPSTOOLS
```

This will result in a note:

```
LPSTOOLS exists, used as a template
```

and will present text to be edited that looks like:

```
:NEWACCT ?       ,MGR;CAP=SM,AM,AL,GL,DI,OP,CV,UV,LG,CS,ND,
SF,BA,IA,PM,MR,DS,PH;ACCESS=(A:AC;W:AC;L:ANY;X:ANY);
MAXPRI=CS;LOCATTR=0;FILES=;CPU=;CONNECT=;PASS=
```

The "?" should be edited to be "FOO".

NewGroup

This command has the following syntax:

```
NewGroup [ <templateaccount> ] [ , CAP = ALL ]
```

The NEWGROUP command (minimum abbreviation NG) is used to edit the attributes for a new group. If NEWGROUP is entered without a **templategroup**, then a default set of attributes is chosen for editing. If a **templategroup** is entered, then MODA will fetch the attributes for that group and display them for editing. The **templategroup** provides a simple way to create a new group with the same attributes as an existing group.

CAP=ALL tells MODA that the attributes for editing should have all possible capabilities.

MODA will report if the **templategroup** exists.

See "NewAccount" on page 137 for examples similar to NEWGROUP.

NewUser

This command has the following syntax:

```
NewUser [ <templateuser> ] [ , CAP = ALL ]
```

The NEWUSER command (minimum abbreviation NU) is used to edit the attributes for a new user. If NEWUSER is entered without a **templateuser**, then a default set of attributes is chosen for editing. If a **templateuser** is entered, then MODA will fetch the attributes for that user and display them for editing. The **templateuser** provides a simple way to create a new user with the same attributes as an existing user.

CAP=ALL tells MODA that the attributes for editing should have all possible capabilities.

MODA will report if the **templateuser** exists.

See “NewAccount” on page 137 for examples similar to NEWUSER.

User

This command has the following syntax:

```
User <name> [ <subset> ]
```

The USER command will display the attributes for the specified user for editing. If the user does not exist, an error is reported.

You can edit attributes of user in accounts other than your logon account by specifying the user as **user.account**. For example, to edit just the local attributes for the user MARY in the account ACCTNG, you could enter:

```
USER MARY.ACCTNG, LOC
```

Selected Summary for the MODIFY Editor Commands

These commands are provided here as a convenient reference to the more common commands used in editing the ALTACCT statement.

NOTE The following command codes are invoked by pressing the letter while holding down the "Ctrl" key.

Table 12.2 *MODIFY Editor Commands*

Command Code	Description
^A	Goto end of line (append)
^B	Activate Insert mode

Command Code	Description
^D	Delete the character at the cursor position
^E	Erase contents from the cursor position to end-of-line
^G	Oops, undo changes
^H	Backspace (non-destructive) cursor left
^O	Activate Overwrite mode
^Y	Abort changes
<Spacebar>	Space bar operations move the cursor to the right

NOTE Pressing ^A (Ctrl+A) or ^B (Ctrl+B) on the system console keyboard invokes special system-management related modes rather than the actions noted above. For instance, ^A invokes CONSOLE mode. if you accidentally go into console mode, press **Return** to quit. If you press ^B at the system console, type "CO" followed by **Return**.

See "The MODIFY Editor" on page 303 for a complete description of the MODIFY editor.

TOOLBOX STANDARDS

The ToolBox collections from Lund Performance Solutions have a uniform user interface. As a result, in addition to the commands specific to each Toolbox tool, most tools allow the commands described in "TOOLBOX STANDARDS" on page 311.

MODA Examples

Following is an example of the MODA tool CLONEACCOUNT command.

```
Wolf:/LPSTOOLS/PUB: run moda

MODA [2.2] - LPS Toolbox [A.09f]            (c) 1995 Lund Performance Solutions

For Help at the MODA prompt enter   ?
This product is licensed to: ImageStats Demo

MODA: cloneaccount lpstools,lpstback
Cloning from account LPSTOOLS to account LPSTBACK
Created account LPSTBACK
   cloned new group C.LPSTBACK
   cloned new group CFG.LPSTBACK
   cloned new group CM.LPSTBACK
   cloned new group COBOL.LPSTBACK
   cloned new group DATA.LPSTBACK
   cloned new group DECL.LPSTBACK
   cloned new group EXTERNAL.LPSTBACK
   cloned new group H.LPSTBACK
   cloned new group HELP.LPSTBACK
   cloned new group HLP.LPSTBACK
   cloned new group INTRIN.LPSTBACK
   cloned new group JOB.LPSTBACK
   cloned new group O.LPSTBACK
   cloned new group PASCAL.LPSTBACK
   cloned new group PICK.LPSTBACK
   cloned new group PUB.LPSTBACK
   cloned new group PUBSYS.LPSTBACK
   cloned new group RL.LPSTBACK
   cloned new group SPL.LPSTBACK
   cloned new group USL.LPSTBACK
   cloned new group UTIL.LPSTBACK
   cloned new group XL.LPSTBACK
   cloned user  MGR.LPSTBACK
Cloned ok

MODA: exit

END OF PROGRAM
Wolf:/LPSTOOLS/PUB:
```

Figure 12.1 *Clone Account*

MODA Error Messages

Each BETIMES error message is described in the following table.

Table 12.3 *MODA Error Messages*

Message	Cause	Action
Expected valid account name	User entered an account-name that does not conform with the convention established by Hewlett-Packard for account names.	Review NEWACCT section in the MPE/iX Command Reference Manual for a complete description of the format for account names.
Expected valid group name	User entered a group-name that does not conform with the convention established by Hewlett-Packard for group names.	Review NEWGROUP section in the MPE/iX Command Reference Manual for a complete description of the format for group names.
Expected valid user name	User entered a user-name that does not conform with the convention established by Hewlett-Packard for user names.	Review NEWACCT section in the MPE/iX Command Reference Manual for a complete description of the format for user names.
You must have AM or SM capability to run this program.	User who tried to run MODA did not have adequate capabilities to run MODA.	Log into an account that has proper capabilities or run GRANT from the **System Manager's Toolbox** to temporarily give yourself the necessary capabilities.

13

THE PAGES TOOL

The PAGES tool allows users to see how memory on their MPE/iX computer is really being used. To achieve this, PAGES offers many commands that allow the user substantial flexibility in requesting memory usage information.

Operation

PAGES looks at every page of physical memory (i.e.: RAM memory) and keeps track of how each page is used. After PAGES scans completely through every page of physical memory, it reports a summary of page usage.

The user has many choices for requesting memory information from PAGES. Memory statistics can be gathered on a number of criteria such as object class, dirty, Recoverable Overlay Candidate (ROC), In Motion In (IMI), and frozen memory.

Other options allow for complete characterization of all available memory, or brief summaries of usage based on user and system demands. PAGES also has an option that shows the advantages realized by adding memory to your HP3000.

To obtain its information, PAGES analyzes each entry in the Physical Page Directory (PDIR). Because of the rather specific nature of this tool, several special sections have been included that cover technical concepts and terminology that may make using PAGES more meaningful. These sections will follow the *Usage* section.

NOTE Based on system use, memory demands and usage can change radically. Drawing conclusions from a single session with PAGES would undermine the objective of analyzing system performance. We recommend that you conduct several PAGES sessions over a period of time to ensure that the information is truly representative of the resource usage at your site.

Capabilities

Program capabilities required include IA, BA, PM, DS and PH. No special user capabilities are required to run PAGES.

Usage

PAGES can be run from the supplied UDC or from a fully-qualified RUN statement.

- UDC

  ```
  PAGES
  ```

- RUN

  ```
  RUN PAGES.PUB.LPSTOOLS
  ```

PAGES does not use any INFO parameters. When executed, PAGES displays a report on:

1. which series HP3000 is running,
2. the operating system version, and
3. how much memory there is.

Memory

This section will provide background information that may enhance your understanding of how PAGES operates and how to use the information that it generates.

Physical and Logical Memory

Physical memory on the MPE/iX machine is organized in pages, where each page contains exactly 4,096 bytes. (On older PA-RISC hardware where the physical page size is 2,048 bytes, MPE always allocates a pair of physical pages at a time, so the logical page size is still 4,096 bytes). Memory is used for many purposes and is always allocated in whole pages, although it is never allocated until it is needed. Another phrase often used to describe memory is "logical page". A logical page is two adjacent physical pages. Each logical page begins with an even numbered physical page.

Most tools that deal with pages in memory use 4,096 bytes as a page size. This is properly called the "logical page" size.

Virtual Memory

Virtual memory on the MPE/iX machine can be thought of as a set of virtual address spaces, with each space measuring 2^{32} bytes (4 gigabytes) in length. Virtual spaces are identified by a space identifier. Virtual addresses are formed by linking the space identifier and the offset within a virtual address space.

Virtual Addresses

Each virtual address space (2^{32} bytes) is divided into 2,048 byte pages or a single physical page size. Unique addressing of any page in a virtual address space requires 21 bits (2^{32} bytes divided by 2,048 bytes). By the same token, uniquely addressing any byte within a page would require a total of 11 bits ($2048=2^{11}$).

Translation Lookaside Buffer

Transforming the virtual address into a physical page of memory is handled by the "Translation Lookaside Buffer" (TLB). The TLB hardware accepts a virtual address as input. The virtual page number portion of the virtual address is used as an index into the TLB's table. If an entry exists at that location, then mapping occurs and a 21-bit physical page number is issued. This physical page number is then linked with the 11-bit page offset portion of the original virtual address to give the full 32-bit physical page address.

The TLB hardware is not large enough to contain all translations. If an entry is not found, a memory structure known as the "Physical Page Directory" (PDIR) is accessed to get the translation information. The PDIR is large enough to contain all translations. The process of accessing the PDIR to get the required translation is known as "TLB handling", or "TLB miss handling". This condition is generally known as a page fault.

PID

A Protection Identifier is a 15-bit number that is assigned to a page for security purposes. When page access is attempted, the PID is matching against a PID list in a control register. If no match is made, access is denied. PID=Zero means no checking is done. The file system verifies that the PID is loaded before access is possible.

Memory Objects

Memory usage on the MPE/iX can be divided into two general categories: system and user. Within these two categories memory can be used for many different tasks. These tasks are grouped by logical functionality into "Object Classes", or by kind into "Object Types".

Object Classes

Every page of virtual memory has an associated "object class", a value in the range 0..800 (also known as "magic number"). Each object class has a particular meaning. For example, object class 8 means "Native Mode System Library" (i.e., NL.PUB.SYS). Thus, the object class is a method of classifying objects (like sex, religion, or skin color is used to classify humans).

Object class is often abbreviated "objclass".

Object Types

PAGES organizes object classes into 8 different types based on use:

SYSTEM_CODE

SYSTEM_DATA

TURBO_DATA

USER_CODE

USER_DATA

USER_STACK

USER_FILE

UNUSED

These Type declarations are generally used to classify memory use.

Glossary of Terms

This will provide information on the terms used in this program that will enhance your understanding of how to use the information that PAGES generates.

DIRTY

Refers to those physical pages that have been written to but not yet posted to disk.

REFERENCED

Refers to those physical pages that have been referenced (read or write) "recently". The memory manager periodically resets the referenced bits.

UNUSED

Refers to those physical pages that are currently in an unused state.

FROZEN

Refers to a specific type of page state. When a page is frozen, it will not be swapped out using the normal demand paging algorithm.

INUSE

Refers to the normal state for physical pages that have been allocated to a process.

IMI (In Motion In)

Refers to a page of physical pages that is assigned to a virtual page, where the virtual page is in the process of being transferred from memory to disk.

PRESENT

Refers to a page of physical memory that is assigned to a virtual page, where the virtual page is "present" and available for access.

ROC (Recoverable Overlay Candidate)

Refers to a page of physical memory that is assigned to a virtual page, where the virtual page is marked as "not present". If a ROC page is accessed, a hardware trap occurs. The operating system marks the page as present, and re-starts the instructions. Physical pages marked as ROC are picked up by the memory manager when it is searching for a physical page to use in handling a page fault for some virtual page.

ABSENT

Refers to a page of physical memory not currently assigned to hold a virtual page.

Object Class By Type

DATA_CLASS

The DATA_CLASS contains object classes associated with user's data. This includes:

CM_DATA

CM_USER_DATA

NOTE Object classes associated with stacks and heaps are classified as USER_CLASS.

FILE_CLASS

The FILE_CLASS contains object classes associated with mapped files (excluding TurboIMAGE files).

SYSTEM_CLASS

The SYSTEM_CLASS contains every object class not contained in any of the other classes. These object classes are typically data/code used by the operating system, not directly by the user.

TURBO_CLASS

The TURBO_CLASS consists of (roughly) the object classes:

TURBO_GLOBAL_CB

TURBO_BUFFER_CB

TURBO_USER_CB

TURBO_REMOTE_CB

TURBO_DSCB_EXT

TURBO_SYS_CB

TURBO_ILR_CB

TURBO_MAIN_CB

TURBO_ABORT_CB

TURBO_ROOT

TURBO_DATA_SET

TURBO_DATA_BASE_ACCESS

TURBO_DBABORT_I

TURBO_ILR_LOG

TURBO_DBRECOV_RESTART

TURBO_DBCHANGE_FILE

The exact classification is controlled by the appropriate PAGES##.DATA file.

UNUSED_CLASS

The UNUSED_CLASS has no object classes associated with it. Instead, when PAGES finds an unused page in memory, it considers it to be part of the unused class.

USER_CLASS

The USER_CLASS consists of the object classes that seem to be associated with "user" oriented data or code...

NM_STACK

CM_STACK

NM_CODE

CM_CODE

NM_HEAP

CM_USER_CODE

NM_PROGRAM

CM_PROGRAM

NOTE Pages belonging to the stack or heap of system processes are counted as part of the USER_CLASS, because PAGES cannot determine whose stack/heap pages are in memory.

Command Summary

The following list provides a summary description of PAGES commands, which can be used to quickly locate the command that suits the task at hand. Detailed information on each command is provided in the next section.

NOTE Portions of the command codes are printed in uppercase to denote the part of the command that PAGES requires in order to distinguish one command from another. However, the commands themselves are not case-sensitive.

Table 13.1 *PAGES Commands*

Command Code	Description
Exit	Terminates PAGES
FIND <what>	Search for specified virtual address or pages
FRozen #	Search for frozen count
HELP	Displays more help information
HPDIR	Walks the page table
IPDIR	Displays summary info for memory manager

Command Code	Description
MMG	Doesn't work on 6.5
PREFetch	(experimental)
Objclass #	Search for specified Object Class
PROCesses	Report memory usage by process (PIN)
RAMUSage	Report affect of adding memory
SCan	Scans through memory (long report)
SET/RESET	Sets/Resets various options
SR6SR7	Reports SR6/SR7 space usage
Status	Reports memory status (short report)
/	Terminates PAGES
?	Synonym for HELP

NOTE Commands may be abbreviated to the uppercase portion.

Command Definitions

This section discusses each of the PAGES commands in detail. At the end of this section is information on each of the option that can be used with the [RE]SET command.

Exit

The Exit command (synonym: /) terminates PAGES.

FIND

The FIND command has the following syntax:

```
FIND [ <address> | ALL | SID spaceid# | SPACEid spaceid# ]
     [options]
```

where:

```
<address> ::= offset | space.offset

options ::=
    [ABSent]
    [DIRTy]
    [DISK]
    [FROZen  [#1 [/#2]]]
    [IMI]
    [INUSEonly]
    [LAUNCH   #1 [/#2]]     (MPE/iX 5.0 and later)
    [LPAGEs   #1 [/#2]]
    [NOTINUSEONLY]
    [OBJclass #1 [/#2]]
    [PIDs     #1 [/#2]]
    [PINs     #1 [/#2]]
    [PRESent]
    [REFerenced]
    [RESident]
    [ROC]
    [UNUSED]
```

The FIND command looks at every page of physical memory, searching for pages that match your specifications. The meaning of each option is explained below.

Specifying more than one option usually results in a page having to meet all of the options. The exception is the options IMI, PRESent, and ROC. If page meets any of those four specified options AND all other options, then it is displayed. For example: most options are "and"ed together; the IMI, PREsent, and ROC options are "or"ed together.

<address>

If a virtual address was specified, only those pages that match that address will be reported.

NOTE Sometimes, several logical pages appear to be associated with the same virtual address. PAGES cannot distinguish between these to determine which (if any) is currently "active".

```
Wolf:/LPSTOOLS/PUB: run pages

PAGES [50.13367] - LPS Toolbox [A.09f]      (c) 1995 Lund Performance Solutions

For Help at the PAGES prompt enter   ?
This product is licensed to: ImageStats Demo

SERIES 918LX
MPE/iX 6.5
#CPUS: 1
Memory size: 160 MB (167,772,160 bytes; 40,960 4KB pages)

PAGES: find $c0000000
Looking for virtual address: $a.$c0000000

Found physical page $132d, virtual page = $0000000a.$c0000000 @ $0.$010f0b40
   IPDIR bits:  REF =     1, DIRTY =     1
                InUse     =  TRUE, MemRes   =  TRUE, NoSwap   = FALSE
                KickedOut = FALSE, Avail    = FALSE, Frozen# =     1
                Referenced= FALSE
   VP_State = PRES_STATE, ObjectClass = 0 =
   PIN = 0, usage = D
Found 1 pages
```

Figure 13.1 *FIND Command*

The FIND command displays the following header line when it starts to search for logical pages matching specified characteristics:

Page$ VirtSpace.VrtOffset RIVDB $PID MN Fr Sta PIN Ob# Object Class Name

Each of the fields in the heading is explained below:

Page$	The logical page number. (Remember, a logical page is 4,096 bytes.)
VirtSpace	The virtual space identifier. (Upper 32 bits of a virtual address.)
VrtOffset	The virtual offset. (Lower 32 bits of a virtual address.)
R	Referenced Bit. This bit is turned on when a page is referenced.
I	(unknown)
V	(unknown)
D	Dirty Bit. This bit is set on when a page is stored into.
B	Data Breakpoint bit. This bit is on when a page has a data breakpoint on it. Data breakpoints are setup with DEBUG's DATAB command.

$PID	Protection ID. This 15-bit value represents the "lockword" for a page.
M	Memory-Resident. When this bit is turned on, the page is "locked" permanently into memory.
N	(unknown)
Fr	Frozen counter. When this value is non-zero, the page is temporarily frozen into memory.
Sta	Page State. The values for this field are PRESENT, IMI, ROC, and ABSENT. Each is explained in the glossary.
PIN	Process Identification Number of process that cause the page to be brought into memory. **NOTE 1** The process may have terminated, and the PIN could have been re-used. PAGES cannot determine if that has happened. **NOTE 2** This field can be suppressed by doing RESET PIN, or enabled by doing SET PIN.
Ob#	Object Class Number. (See also Object Class Name). If a SET OBJNUM has been done, then this field appears.
Object Class Name	Every "object" (a range of virtual addresses) created by MPE/iX has a magic number associated with it called the "object class". This number can be used to help determine how many pages of disc storage should be fetched when a page fault occurs. The Object Class Name field is the "English" name for the Object Class Number.

ALL (default)

FIND ALL tells PAGES to look at all logical pages, not just those associated with a particular virtual address or virtual space.

```
Wolf:/LPSTOOLS/PUB: run pages

PAGES [50.13367] - LPS Toolbox [A.09f]      (c) 1995 Lund Performance Solutions

For Help at the PAGES prompt enter   ?
This product is licensed to: ImageStats Demo

SERIES 918LX
MPE/iX 6.5
#CPUS: 1
Memory size: 160 MB (167,772,160 bytes; 40,960 4KB pages)

PAGES: find all

Page$ VirtSpace.VrtOffset RD MN Froz Status   PIN Object_Class_Name
----- ---------.--------- -- -- ---- ------   --- -----------------
    1 $00000000.$00001000          1 PRESENT    0
    2 $00000000.$00002000          1 PRESENT    0
    3 $00000000.$00003000          1 PRESENT    0
    4 $00000000.$00004000          1 PRESENT    0
    5 $00000000.$00005000          1 PRESENT    0
    6 $00000000.$00006000          1 PRESENT    0
    7 $00000000.$00007000          1 PRESENT    0
    8 $00000000.$00008000          1 PRESENT    0
    9 $00000000.$00009000          1 PRESENT    0
    a $00000000.$0000a000          1 PRESENT    0
    b $00000000.$0000b000          1 PRESENT    0
    c $00000000.$0000c000          1 PRESENT    0
    d $00000000.$0000d000          1 PRESENT    0
    e $00000000.$0000e000          1 PRESENT    0
    f $00000000.$0000f000          1 PRESENT    0
   10 $00000000.$00010000          1 PRESENT    0
   11 $00000000.$00011000          1 PRESENT    0
   12 $00000000.$00012000          1 PRESENT    0
   13 $00000000.$00013000          1 PRESENT    0
   14 $00000000.$00014000          1 PRESENT    0
Hit <return> to continue, / to stop:
```

Figure 13.2 *FIND ALL option*

SPACE # (or SID #)

The SPACE option tells PAGES to look only for logical pages that are used to hold virtual pages belonging to the specified space ID.

Example: `FIND SPACE $3a6`

Most of the rest of the FIND options restrict the set of pages that will be reported:

ABSent

Restricts the FIND command to just those logical pages that are in the "Absent" state. Normally, few pages are in this state.

NOTE It is unclear if the various fields describing a page are in a valid state for absent pages!

Example: `FIND ABSENT` (will find pages that are absent)

DIRTy

Restricts the FIND command to just those logical pages that have been written to and not yet posted to disc.

Example: `FIND DIRTY`

FROZen [#1 [/#2]]

Restricts the FIND command to just those logical pages whose frozen count is in the range specified. (If /#2 is omitted, #2 is set to #1, which will seach for just that frozen count.)

If #1 is also not given, #1/#2 defaults to 1/255.

Example 1: `FIND FROZEN 3` (similar to FROZEN 3 command)

Example 2: `FIND FROZEN 3/9`

IMI

Restricts the FIND command to just those logical pages that are in the "In Motion In" state.

IMI, PRESent, and ROC are "OR"ed together.

Example: `FIND IMI`

INUSEonly

Restricts the FIND command to just those logical pages that appear to be "in use" by MPE/iX. MPE reserves some pages in low memory for special purposes (i.e., for use by the MEMSAVE utility used by the DUMP process), and PAGES normally ignores these pages.

INUSEonly is on by default, and can be turned off by specifying NOTINUSEONLY or UNUSED.

LAUNCH #1 [/#2]] (MPE/iX 5.0 and later)

Restricts the FIND command to just those logical pages in the whose "launch counter" is in the specified range.

LPAGE #1 [/#2]

Restricts the FIND command to just those logical pages in the range #1 to #2. (If /#2 is omitted, #2 is set to #1, which will "find" at most one logical page.)

Example 1: `FIND LPAGE 90`

Example 2: `FIND LPAGE 0/20`

NOTINUSEONLY

Tells FIND that all pages should be examined, not just those marked "in use" (see INUSEonly above).

OBJclass #1 [/#2]

Restricts the FIND command to just those logical pages whose object class is in the range specified. (If /#2 is omitted, #2 is set to #1, which will search for just that object class.)

Example 1: `FIND OBJCLASS 200` (similar to OBJCLASS 200 command)

Example 2: `FIND OBJCLASS 200/209`

PID #1 [/#2]

Restricts the FIND command to just those logical pages whose protection ID (PID) is in the range specified. (If /#2 is omitted, #2 is set to #1, which will search for just that PID.)

Example 1: `FIND PID 34`

Example 2: `FIND PID 34/90`

PIN #1 [/#2]

Restricts the FIND command to just those logical pages which were brought into memory on behalf of processes in the specified PIN range. (If /#2 is omitted, #2 is set to #1, which will search for just that PIN.)

Example 1: `FIND PIN 10`

Example 2: `FIND PIN 10/12`

PRESent

Restricts the FIND command to just those logical pages that are in the "PRESent" state.

IMI, PRESent, and ROC are "OR"ed together.

Example: `FIND PRESENT`

REFerenced

Restricts the FIND command to just those logical pages that have been referenced (read or write) "recently". The memory manager periodically resets the referenced bits.

Example 1: `FIND REF`

Example 2: `FIND DIRTY REF`

RESident

ROC

Restricts the FIND command to just those logical pages that are in the "Recoverable Overlay Candidate State".

IMI, PRESent, and ROC are "OR"ed together.

UNUSED

Restricts the FIND command to just those logical pages that are NOT marked as "in use" by MPE (see INUSEonly above).

NOTE If the ABSent option is not seen, then no pages in the "absent" state will be shown.

The above options restrict the set of pages found. Once a page is found, however, information about it is reported. The following options can affect the output displayed:

DISK

Tells the FIND command to report the disk ldev and disk page number of the start of the object on disk.

FRozen

The FROZEN command has the following syntax:

```
FRozen #
```

This command looks at every page of physical memory and reports those pages that have been "frozen" the specified number of times. This is similar to the FIND FROZEN option, but provides less information.

HELP

The HELP command has the following syntax:

```
HELP [xxxx]
```

HELP <cr> produces a display of the entire help file.

HELP <xxxx> displays help for all commands that start with xxxx.

Example: `HELP F` displays help for all commands that begin F.

HPDIR

This command has the following syntax:

```
HPDIR [ <DUMP>          |
        <FLAGS ccc>     |
        <INVALIDS>      |
        <LPAGE  # [/#]> |
        <OFFSET # [/#]> |
        <PID    # [/#]> |
        <SID    # [/#]> |
        <SUMMARY>       |
        <USAGE  # [/#]> ]
```

This is an experimental command to list all or part of the HPDIR table.

NOTE HPDIR exists only on MPE/iX 5.0 or later.

SPACEID may be used as a synonym for **SID**.

The **DUMP** option causes details of every selected HPDIR entry to be listed. **DUMP** causes all entries to be selected unless the **FLAGS**, **LPAGE**, **OFFSET**, **PID**, **SID**, or **USAGE** option is used.

The **FLAGS**, **LPAGE**, **OFFSET**, **PID**, and **SID** options can be used to dump a subset of the HPDIR entries. Each of these options restricts the set of entries that are displayed in a "logical and" manner. Thus, if both **FLAGS** and **OFFSET** are specified, an entry will be displayed only if it matched both the **OFFSET** specification, and one (or more) of the **FLAGS** characters.

FLAGS ccc restricts the dumped entries to those whose flags match one or more of the "ccc" characters. Note that "ccc" is case sensitive. To dump only those entries that are dirty, a "FLAGS d" would be used.

The **LPAGE** option allows a logical page (or range of logical pages) to be specified.

Example: `LPAGE 100/200` will list only entries for logical pages whose 100..200;

The **OFFSET** option allows a virtual address offset (or offset range) to be specified.

Example: `OFFSET 100/200` will list only entries for pages whose virtual address is >= X.100 and <= X.200, for all X.

The **PID** option allows a protection ID (or range of PIDs) to be specified.

Example: `PID 100/200` will list only entries for pages whose Protection IDs are in 100..200.

The **SID** (or SPACEID) option allows a Space ID (or range of SIDs) to be specified.

Example: `SID 100/200` will list only entries for virtual addresses in the range 100.0 through 200.$7fffffff.

The **USAGE** option allows a usage # (0..7) (or range of usage #s) to be specified.

Example: `USAGE 2/3` will list only entries for pages whose usage# is 2 or 3 (code, or code/rwx).

The **INVALIDS** option causes the address of currently invalid HPDIR entries to be listed. (Usually, none are found.)

The **SUMMARY** option, default, causes HPDIR to display just a summary of the usage of the HPDIR table.

Objclass

The OBJCLASS command has the following syntax:

```
Objclass #
```

This command searches through memory looking for pages that belong to the specified object class. Every page found is reported. This is similar to the FIND OBJCLASS option, but less information is displayed.

PROCesses

This command has the following syntax:

```
PROCesses [min#pages] [<SORT | NOSORT>]
```

The PROCESSES command scans through memory, trying to determine what process caused each page to be brought into memory. It then summarizes the number of pages, by process.

The **SORT** option reports the processes in order of descending number of pages brought into memory.

The **min#pages** option excludes processes that have brought fewer pages into memory. min#pages defaults to 20.

NOTE The "number brought in" is not a historical counter ... it is the number of pages *currently in memory* that were brought in on behalf of a particular PIN (Process Identification Number). This usually means that the process either page faulted on the page, or did a "prefetch" on the page. In the case of pages shared by multiple processes, only the most recent page-fault (or prefetch) is known to us.

Note that, since PINs are reused, there is no way of knowing if a page brought in for PIN 123 was brought in for the *current* process with PIN 123, or a prior process that used PIN 123.

RAMUSage

The RAMUSAGE command reports how much "user" memory would be available if you added more physical memory to the system. It assumes that all of the newly added memory would be used for user memory, not system code/data.

SCan

This command has the following syntax:

```
SCan [%ofmemory]
```

The SCAN command looks at every page of physical memory and then reports statistics about how physical memory is being used. If the optional parameter, **%ofmemory**, is specified, then only those object classes that occupy more than the specified percent of memory are shown.

Example: `SCAN 2`

```
Wolf:/LPSTOOLS/PUB: run pages

PAGES [50.13367] - LPS Toolbox [A.09f]      (c) 1995 Lund Performance Solutions

For Help at the PAGES prompt enter   ?
This product is licensed to: ImageStats Demo

SERIES 918LX
MPE/iX 6.5
#CPUS: 1
Memory size: 160 MB (167,772,160 bytes; 40,960 4KB pages)

PAGES: scan
Memory size: 40,960 pages (160 MB)
     23,229 Dirty,   26,117 Referenced,    9,403 Frozen.      191 Unused.
  (  17,169 Dirty,   18,526 Referenced of the unfrozen pages)

   State of in-use pages:  37,890 Present, 0 IMI, 2,808 ROC, 70 Absent.

Frequency of "freeze" counts:
   freeze count   # pages
         0          31,365
         1           8,892
         2             124
         3               1
         4             385
        31               1

Frequency of page type (from IPDIR):

    PageType     Count        PageType     Count
    --------   --------       --------   --------
      DATA_R      1,224         GATE_0          3
     DATA_RW     33,157         GATE_1          2
        CODE      4,552         GATE_2          1
    CODE_RWX      1,829         GATE_3          0

Frequency of pages of different object classes:

   (showing only those object classes occupying > 1% of memory)
```

```
OC#  Object Class               #pages | OC#  Object Class               #pages
---- ------------------------- ------- | ---- ------------------------- -------
   0 TRANSIENT_DATA              6,924 |    2 NM_STACK                    2,644
   3 CM_STACK                      883 |    8 NM_SYS_LIB                  1,921
   9 CM_SYS_LIB                    774 |   26 SSM_MAP                     1,227
  70 LDR_PROCESS_XRT             1,555 |   73 LDR_PFL table               2,170
  77 USER_NM_DATA               10,247 |   81 FS_PLFD table                 803
  83 NM_HEAP                     2,870 |   87 PIBX                        1,222
 663 NM_PROGRAM                  1,610 |  665 ORD_FIX                       835
 711 XM_LOG                        425 |

Memory usage by "type" of Object Class:

       Class     #LogicalPages  #MB  % total
   ------------- -------------  ---  -------
   SYSTEM_CODE           3,043   11     7.4%
   SYSTEM_DATA          17,917   69    43.7%
   UNUSED                  191    0     0.5%
   USER_CODE             1,776    6     4.3%
   USER_DATA            10,336   40    25.2%
   USER_STACK            6,397   24    15.6%
   USER_FILE             1,299    5     3.2%

   Totals:              40,959   159   100.0%

   "User" pages are  48.8% of memory (79 MB out of 160 MB)
```

```
  $SID   #Pages | $SID   #Pages | $SID   #Pages | $SID   #Pages |
    $0   3,352 |    $a   7,977 |    $b   1,911 |   $10     159 |
   $12   1,403 |   $13     178 |   $15     186 |   $1a     167 |
   $26     134 |   $2b     774 |   $64      95 |   $66      50 |
   $6a     239 |   $c4     275 |   $da     348 |   $fa   1,893 |
  $116      49 |  $186      96 |  $1b1     181 |  $1b4     142 |
  $1b6     171 |  $1b8     106 |  $1d0      40 |  $205      49 |
  $242      41 |  $247      53 |  $250      43 |  $258      50 |
  $267     155 |  $269      63 |  $27a     279 |  $27b      45 |
  $27e      92 |  $285      41 |  $288     128 |  $289     169 |
  $292      65 |  $5d5      86 |  $5d6     146 |  $5d7     146 |
  $5d8     146 |  $5d9     146 |  $5da     146 |  $5db     146 |
  $5dc     146 |  $5dd     146 |  $5de     146 |  $5df     146 |
  $5e0     145 |  $5e1     146 |  $5e2     146 |  $5e3     146 |
  $5e4     146 |  $5e5     146 |  $5e6     146 |  $5e7     146 |
  $5e8     146 |  $5e9     146 |  $5ea     146 |  $5eb     146 |
  $5ec     146 |  $5ed     146 |  $5ee     146 |  $5ef     146 |
  $5f0     146 |  $5f1     146 |  $5f2     146 |  $5f3     146 |
  $5f4     146 |  $5f5     146 |  $5f6     146 |  $5f7     146 |
  $5f8     146 |  $5f9     146 |  $5fa     146 |  $5fb     146 |
  $5fc     145 |  $5fd     146 |  $5fe     146 |  $5ff     146 |
  $600     146 |  $601     146 |  $602     146 |  $603     146 |
  $604     146 |  $605     146 |  $606     146 |  $607     146 |
  $608     146 |  $609     145 |  $60a     146 |  $60b     146 |
  $60c     146 |  $60d     146 |  $60e     145 |  $60f     177 |
  $610     146 |  $611     157 |  $612     177 |  $613     177 |
  $614     177 |  $615     177 |  $616     177 |  $617     177 |
  $618     177 |  $619     177 |  $61a     177 |  $61b     177 |
  $61c     177 |  $61d     177 |  $61e     177 |  $61f     177 |
  $620     177 |  $621     177 |  $622     177 |  $623     177 |
  $624     177 |  $625     177 |  $626     177 |  $627     177 |
  $628     177 |  $629     177 |  $62a     177 |  $62b     177 |
  $62c     177 |  $62f     131 |  $639      67 |  $63a      51 |
  $641      57 |  $649      57 |  $64c     111 |  $64e     126 |
  $652     135 |  $658     139 |

Largest SID ever seen: $65a

PAGES:
```

Figure 13.3 *SCAN Command*

SR6SR7

This command has the following syntax:

```
SR6SR7 option
```

The SR6SR7 command (and SR6 and SR7 commands) scans the SR6 and/or SR7 quadrant of virtual memory. These two one-gigabyte chunks of virtual memory are where many system data structures are allocated, as well as short mapped files.

The "SR6 ..." command is equivalent to "SR6SR7 SR6 ...".

The "SR7 ..." command is equivalent to "SR6SR7 SR7 ...".

The options are described below.

ALL	Turns on all options, for both SR6 and SR7.

BOTH	Requests free space distribution report for both SR6 and SR7.
List	Lists every Virtual Space Object Descriptor (VSOD) associated with an object in SR6 and/or SR7
SUMmary	Reports only the high-level summary of SR6/SR7 space usage.

SET | RESET

These commands have the following syntax:

```
SET option
RESET option
```

The SET/RESET commands are used to set/reset various options. Options available are described below.

SHOWADD	Enabling this option causes PAGES to display the advantage (in terms of user memory) of adding more memory to the machine. This information is displayed at the end of both the STATUS and SCAN reports.
TIMES	When selected, this option causes PAGES to display CPU usage after each command is executed.
OBJNUM	Selecting this option will cause PAGES to display object class numbers rather than object class names. The primary effect of this option is on the output of the FIND command.
PIN	This option causes the FIND command to display the PIN which brought the page into memory. **NOTE** If the process has been terminated before you have issued this option, the PIN indicated may be in use by another process.

Another three options available are: EXPERT, OBJCLSUMMARY and SIDCOUNTS.

Status

The STATUS command provides a quick summary of how memory is being used.

NOTE The section starting with 'Memory usage by "type" of Object Class' can be suppressed by doing RESET OBJCLSUMMARY or enabled by doing SET OBJCLSUMMARY.

```
PAGES: status
Memory size: 40,960 pages (160 MB)
     23,160 Dirty,   26,013 Referenced,    9,403 Frozen.      220 Unused.
  (  17,112 Dirty,   18,434 Referenced of the unfrozen pages)

   State of in-use pages:  37,822 Present, 0 IMI, 2,847 ROC, 70 Absent.

Memory usage by "type" of Object Class:

       Class      #LogicalPages  #MB  % total
   -------------- -------------  ---  -------
   SYSTEM_CODE           3,043   11     7.4%
   SYSTEM_DATA          17,911   69    43.7%
   UNUSED                  220    0     0.5%
   USER_CODE             1,778    6     4.3%
   USER_DATA            10,294   40    25.1%
   USER_STACK            6,382   24    15.6%
   USER_FILE             1,331    5     3.2%

   Totals:              40,959   159   100.0%

   "User" pages are  48.8% of memory (79 MB out of 160 MB)

PAGES:
```

Figure 13.4 *STATUS Command*

Memory usage is dynamic. Each time you execute the STATUS command, different output is reported.

NOTE In general, regarding optimal memory quantity, the percentage of "user" memory pages should be at least 60-70% for most sites. If user memory page counts are less than this, you may need more memory.

TOOLBOX STANDARDS

The ToolBox collections from Lund Performance Solutions have a uniform user interface. As a result, in addition to the commands specific to each Toolbox tool, most tools allow the commands described in "TOOLBOX STANDARDS" on page 311.

PAGES Examples

The following illustrates a PAGES operation showing that amount of memory used by the object CM_STACK (objclass number 3).

```
Wolf:/LPSTOOLS/PUB: run pages

PAGES [50.13367] - LPS Toolbox [A.09f]      (c) 1995 Lund Performance Solutions

For Help at the PAGES prompt enter   ?
This product is licensed to: ImageStats Demo

SERIES 918LX
MPE/iX 6.5
#CPUS: 1
Memory size: 160 MB (167,772,160 bytes; 40,960 4KB pages)

PAGES: objclass 3
Logical page:   $7d, Virtual: $00000152.$41626000
Logical page: $1665, Virtual: $0000002d.$41622000
Logical page: $1680, Virtual: $000001fb.$41621000
Logical page: $16c6, Virtual: $000001fc.$41621000
Logical page: $220b, Virtual: $00000171.$41625000
Logical page: $22f8, Virtual: $00000032.$41622000
Logical page: $23e9, Virtual: $00000216.$41623000
Logical page: $2430, Virtual: $0000018a.$41624000
Logical page: $25d0, Virtual: $0000027a.$41621000
Logical page: $25f2, Virtual: $00000147.$41623000
Logical page: $2633, Virtual: $00000111.$41623000
Logical page: $26ba, Virtual: $00000150.$41624000
Logical page: $273f, Virtual: $0000019b.$41621000
Logical page: $274f, Virtual: $0000002d.$41623000
Logical page: $27a4, Virtual: $00000148.$41626000
Logical page: $2800, Virtual: $0000019b.$41622000
Logical page: $280d, Virtual: $00000111.$41621000
Logical page: $29a6, Virtual: $00000171.$41626000

...

Logical page: $9e8a, Virtual: $0000027a.$41627000
Logical page: $9ef3, Virtual: $00000216.$41625000
Logical page: $9f40, Virtual: $0000018a.$41622000
Logical page: $9f4a, Virtual: $00000152.$41621000
Logical page: $9f9d, Virtual: $00000130.$41621000

PAGES:
```

Figure 13.5 *Object memory*

The following three figures show various applications of the FIND command.

```
PAGES: find roc

Page$ VirtSpace.VrtOffset RD MN Froz Status   PIN Object_Class_Name
----- ---------.--------- -- -- ---- ------   --- -----------------
   4f $00000015.$00374000          0 ROC       66 XM_LOG
   52 $000005db.$4164e000          0 ROC       80 USER_NM_
   83 $000005e4.$46805000          0 ROC      500 USER_NM_
   88 $000005e4.$4020c000          0 ROC       80 LDR_PFL
   94 $000001a9.$00003000 R        0 ROC       66 ORD_FIX
  c08 $000005e4.$4020d000          0 ROC       80 LDR_PFL
  c0a $000005e4.$4020e000          0 ROC       80 LDR_PFL
  c0e $00000617.$467f3000          0 ROC      551 USER_NM_
  c3c $00000015.$0038a000          0 ROC       66 XM_LOG
  c3f $00000015.$00385000          0 ROC       66 XM_LOG
  c94 $0000006a.$00154000          0 ROC       66 XM_LOG
  cb6 $00000015.$002e9000          0 ROC      576 XM_LOG
  cb9 $00000015.$002ea000          0 ROC      576 XM_LOG
  cde $0000006a.$001ee000          0 ROC       66 XM_LOG
  ce6 $0000006a.$00160000          0 ROC      576 XM_LOG
 136a $0000006a.$00193000          0 ROC       66 XM_LOG
 136c $0000006a.$00194000          0 ROC       66 XM_LOG
 136d $000005d7.$41642000          0 ROC       80 USER_NM_
 136e $000005d7.$41643000          0 ROC       80 USER_NM_
 136f $0000006a.$00195000          0 ROC       66 XM_LOG

Page$ VirtSpace.VrtOffset RD MN Froz Status   PIN Object_Class_Name
----- ---------.--------- -- -- ---- ------   --- -----------------
Found 20 pages


PAGES:
```

Figure 13.6 *FIND ROC Command*

```
PAGES: find dirty

Page$ VirtSpace.VrtOffset RD MN Froz Status   PIN Object_Class_Name
----- ---------.--------- -- -- ---- ------   --- -----------------
   42 $0000056b.$41846000 RD       0 PRESENT  379 NM_STACK
   43 $00000606.$4165a000  D       0 PRESENT   80 USER_NM_DA
   45 $00000601.$4184c000  D       0 PRESENT  529 NM_STACK
   46 $00000600.$467e4000  D       0 PRESENT  528 USER_NM_DA
   47 $00000601.$4184e000  D       0 PRESENT  529 NM_STACK
   48 $0000027a.$41656000 RD       0 PRESENT   71 USER_NM_DA
   4c $00000601.$41659000  D       0 PRESENT   80 USER_NM_DA
   4d $00000626.$4680e000 RD       0 PRESENT  566 USER_NM_DA
   4e $00000026.$00033000 RD M     1 PRESENT    - FIO_STATE
   53 $0000000b.$83b3f000 RD       0 PRESENT   68 XM_CB_POOL
   55 $000001b8.$41848000 RD       0 PRESENT  575 NM_STACK
   56 $000001b8.$4164d000 RD       0 PRESENT  578 USER_NM_DA
   58 $0000000a.$d5622000 RD       0 PRESENT   80 PIBX
   59 $000005e9.$41672000  D       0 PRESENT   80 USER_NM_DA
   5a $0000061f.$41656000 RD       0 PRESENT   80 USER_NM_DA
   60 $0000047e.$467dc000 RD       0 PRESENT  152 USER_NM_DA
   62 $00000600.$467dd000  D       0 PRESENT  528 USER_NM_DA
   64 $00000601.$4165e000  D       0 PRESENT   80 USER_NM_DA
   65 $00000600.$467da000  D       0 PRESENT  528 USER_NM_DA
   66 $00000600.$467db000 RD       0 PRESENT  528 USER_NM_DA

Page$ VirtSpace.VrtOffset RD MN Froz Status   PIN Object_Class_Name
----- ---------.--------- -- -- ---- ------   --- -----------------
Found 20 pages


PAGES:
```

Figure 13.7 *FIND DIRTY Command*

```
PAGES: find unused

Page$ VirtSpace.VrtOffset RD MN Froz Status   PIN Object_Class_Name
----- ---------.--------- -- -- ---- ------   --- -----------------
    0 $0000638b.$00001466       65535 PRESENT    0 TERM_SP_DM
   54 $000002d0.$41843000 RD        1 ABSENT    66 NM_STACK
   55 $000001b8.$41848000 RD        1 ABSENT   575 NM_STACK
   56 $000001b8.$4164d000 RD        1 ABSENT   578 USER_NM_DAT
  c36 $00000262.$00006000 R         1 ABSENT   575 ORD_FIX
  c38 $000001b8.$41621000 RD        1 ABSENT   578 CM_STACK
  c39 $000002d0.$4020c000 R         1 ABSENT    68 LDR_PFL tab
  c8d $000002d0.$40101000 R         1 ABSENT    68 LDR_PROCESS
  d25 $000002cf.$00008000 RD        1 ABSENT   578 LDR_UNSAT t
  d66 $000001b8.$41625000 R         1 ABSENT   578 CM_STACK
 13bf $000002d0.$41846000 RD        1 ABSENT   576 NM_STACK
 1f12 $000001b8.$41681000 R         1 ABSENT   575 USER_NM_DAT
 2332 $0000000a.$d638d000 RD        1 ABSENT   578 FS_PLFD tab
 2454  reserved                     0 ABSENT     0
 26d4 $000002eb.$00001000 RD        1 ABSENT    68 LDR_UNSAT t
 271c $000001b8.$41648000 RD        1 ABSENT   578 USER_NM_DAT
 2849 $000001b8.$40102000 RD        1 ABSENT   578 LDR_PROCESS
 2866 $000001b8.$41671000 RD        1 ABSENT   578 USER_NM_DAT
 2904 $000001b8.$40202000 RD        1 ABSENT   578 LDR_PFL tab
 2907 $000001b8.$41684000 RD        1 ABSENT   578 USER_NM_DAT

Page$ VirtSpace.VrtOffset RD MN Froz Status   PIN Object_Class_Name
----- ---------.--------- -- -- ---- ------   --- -----------------
Found 20 pages


PAGES:
```

Figure 13.8 *FIND UNUSED Command*

PAGES Error Messages

Each PAGES error message is described in the following table.

Table 13.2 *PAGES Error Messages*

Message	Cause	Action
Bad frozen count value	Either no number or an illegal number was entered.	Enter a number in the range 0 to 127 (decimal).
Bad Object Class number	Either no number or an illegal number was entered.	Enter a number in the range 0 to 500 (decimal).

Message	Cause	Action
Expected an address	Invalid **virtual address** given to FIND command.	Enter valid **virtual address**, see FIND command.
Expected offset within space	Bad **virtual address** offset specification.	FIND **spaceid** expects "**.<offset>**" or "**.@**"
Expected space ID	Bad **virtual address** space id specification.	FIND SPACE expects **spaceid**.
Failed to open PAGES.DATA.LPSTOOLS data file	PAGES could not open companion file.	Check to ensure that PAGES.DATA.LPSTOOLS is available.
Frozen count must be in range 0..127	A number not in range 0 to 127 was entered as a FROZEN count.	Choose a number in range 0 to 127 (decimal).
Unexpected I/O error reading PAGES.DATA, unable to load Class data	FILE SYSTEM I/O error during the loading of PAGES.DATA.LPSTOOLS.	Ensure the integrity of the file PAGES.DATA.LPSTOOLS.
Unknown class type	During the loading of PAGES.DATA.LPSTOOLS, an unknown class type was discovered.	Correct the offending entry to the file PAGES.DATA.LPSTOOLS.

14

THE REDWOOD TOOL

REDWOOD is a logfile analyzer that specializes in looking at file-close records from MPE XL, MPE V, and MPE IV logfiles.

Operation

Optimizing disk I/O performance can be a costly and time consuming job. REDWOOD can make this process easier by identifying the most frequently accessed files on the system. Thus, when you do choose to optimize your system, you can be sure that your time is being spent productively. System optimization can yield a significant decrease in execution time. Determining which files to optimize to achieve these kinds of results is a matter of analyzing the frequency of logical access, physical access, and the number of times a file is opened. This is the type of information that REDWOOD provides.

REDWOOD makes a compressed copy of the data in the system log file(s) and places it into a user-defined summary file. REDWOOD uses this summary file to create the reports you design.

As you use REDWOOD, you will notice that REDWOOD frequently displays a "(**CR = <value>**)" at user prompts. This is REDWOOD's way of showing default choices. Press "CR" (**Enter**) to select the default.

REDWOOD analyzes both MPE V log files and MPE/iX log files. Before using REDWOOD, you will need to make sure that FCLOSE logging is enabled so that REDWOOD has something to which to report. Use the SYSGEN utility to determine and modify (if necessary) your system's configuration to include **fclose logging**.

Refer to the *System Startup and Shutdown* manual for details on modifying system log files. Or, use the instruction sequence that follows.

Getting Started

The following instruction explain how to enable FCLOSE logging on an MPE/iX machine. Classic system log files are defined as TYPE 160. Native Mode log files are defined as TYPE 105.

1. Logon as MANAGER.SYS
2. Type **:sysgen** to invoke the SYSGEN program. You should see something similar to the following display:

```
Wolf:/SYS/MANAGER: sysgen
SYSGEN version E.03.00 : catalog version E.03.00    MON, SEP 10, 2001,  4:09 AM
Copyright 1987 Hewlett-Packard Co. All Rights Reserved.

        **note** Retrieving NMMGR configuration data...


        ** First level command **

        io                log (lo)       misc (mi)        spu (sp)
        sysfile (sy)

        basegroup (ba)    keep(ke)       permyes (pe)     show (sh)
        tape (ta)

        clear (cl)(c)     exit (ex)(e)   help (he)(h)     oclose (oc)
        redo
 sysgen>
```

Figure 14.1 *SYSGEN Program Screen*

3 Next, type **log** at the **sysgen>** prompt to display the following configuration commands.

```
        ** LOG configurator commands **

        show (sh)         slog (sl)       ulog (ul)

        clear (cl)(c)     exit (ex)(e)    help (he)(h)    hold (ho)
        oclose (oc)       redo
    log>
```

Figure 14.2 *LOG Configuration Commands*

4 Type **sl on=160,105** at the **log>** prompt.

5 Next, type **hold** at the **log>** prompt.

6 Then, type **exit** at the **log>** prompt. This returns you to the **:SYSGEN** prompt.

7 Type **keep** at the **:SYSGEN** prompt.

8 Answer **yes** to the **Purge old configuration (yes/no):** prompt.

9 Finally, type **exit** to terminate the program.

Overview

Disc I/O on the PA-RISC HP3000 is affected by few factors under the direct control of the programmer. These factors include: the amount of memory on the machine, the number of users competing for memory, and the transaction manager. Factors under the control of the programmer include: file locality (random or sequential access), FCONTROLs to post dirty pages, and the sequential write queue.

Disc I/O on the Classic HP3000 is affected by many things: blocking factors, the number of buffers available, the type of file (variable or fixed), the spread of the disc files across the available devices, the type of calls made against the file as well as the sequence of those calls and their frequency.

For an individual file, all of these considerations must be taken into account when we attempt to optimize our system. As well, we must consider the use of each file as it pertains to the application and if it is a structured file, we must analyze our selection of keys and file relationships.

The benefits involved in handling these tradeoffs correctly are many, as are the costs involved with incorrect file choices.

Consider an example on the Classic HP3000, a file which will be accessed sequentially. The blocking factor determines how many physical I/O's are necessary to process a given number of logical records. If a file contains 10,000 logical records which must be sequentially read by a program then the formula for the number of physical reads necessary is as follows:

```
physical'reads = [ logical'records / blocking'factor ]
```

where the expression [ ] is rounded up to the nearest integer.

Thus, in our ten thousand record file, a blocking factor of two results in 10000/2, or 5,000 physical reads against the file. A blocking factor of twenty results in 10000/20, or 500 reads and an improvement factor of ten in the number of physical transfers necessary. Logical record accesses cost in terms of cpu as well as the amount of memory required during the transfer. Physical accesses cost in terms of several additional resources - the disc movement required, the controller time, channel time, etc. Assuming our application does require the logical accesses, we can not reduce them (although NOBUF may reduce their impact).

However, by varying the blocking factor, we can reduce the number of physical accesses required (at a cost of increasing the amount of memory our buffers are using). The end result of this change can be significant. One test showed that altering a blocking factor from two to twenty, resulted in a thirty percent decrease in the execution time of a sequential-read application. Similar optimizations are possible by varying some of the other file characteristics or by rewriting portions of the file handling applications.

The above is an excellent example of the difference between MPE XL on the PA-RISC HP3000 and MPE V on the Classic HP3000. Changing the block factor on MPE XL usually results in no performance differences at all. (For a few files, it can still improve the performance, but this is for relatively rare kinds of files (i.e.: RIO).)

Such optimizations can provide significant improvements in the performance of a system. However, each such optimization has an implicit cost associated with it. It may be simply the time the programmer must take to redo the file equations or :BUILD commands of a batch job. On the other hand, the optimization may involve extensive recoding effort as well as testing. Most optimizations involve exploiting relatively harmless tradeoffs. In the case of the blocking factor change mentioned above, the decrease in the number of physical I/O's more than compensated for the additional burden the larger buffers placed on memory. This is not always the situation though, and each modification must be tested to determine if it does indeed improve or degrade performance.

Since these file optimizations have costs associated with them, we would like to pay the cost only when we are sure of a chance of a reasonable return on our investment. There is a theory of programming which says that eighty percent of the time a program is executing it is in twenty percent of the code. This 80/20 rule can also be applied to files, or "eighty percent of all file activity is against twenty percent of the files". To receive a maximum return on our optimization investment, we should obviously focus our attention on this "top twenty" percent, and the topic which REDWOOD addresses is the identification of that twenty percent.

Method

This top 20 might also be referred to as the "busy" or "heavily used" files on the system. Before we can identify the top 20, we must establish our criteria for deciding which files are busier than others. Three criteria come immediately to mind when we speak of heavy file usage:

1 the number of logical accesses to the file,

2 the number of physical accesses, and

3 the number of times a file is opened.

This information (with a few exceptions noted later) is available to us through the type 5 record recorded by MPE V in its log files and in the type 105 records recorded by MPE XL.

The format of the type 5 (FCLOSE) record is given on pages 6-123 and 6-124 of the old SYSTEM MANAGER/SYSTEM SUPERVISOR Reference Manual. The information which will be of interest to us includes:

- file name
- logical device # of file label & first extent
- number of records processed
- number of blocks processed

where:

file name

The fully qualified formal designator associated with the file, **fname.group.account**; some program temporary files may have a blank name.

logical device

The logical device number of the file label. This ldev may not contain the entire file since only a single extent need reside entirely on one device.

number of records

The number of logical records which have been read or written since the file was opened, this value gives us a measure of the application's activity.

number of blocks

The number of blocks which have been transferred to/from the file. This value is a measure of the physical I/O against the file in all instances except the following two cases:

a. a rewind (FCONTROL (5)) against a variable record file resets this value to zero;

b. for files which are accessed with MULTI-REC (bit 11 of AOPTIONS or MR in file eq.) and where the block size is equal to an integer multiple of 128 words. In this case the value is the number of blocks processed (which will probably be greater than the number of physical I/O's).

MPE XL's type 105 record has similar information within it. The layout, however, does not appear to be documented in any publicly accessible manuals. In the process of writing REDWOOD, we located and were able to determine its contents.

NOTE REDWOOD can read logfiles from MPE XL, MPE V, and MPE IV, automatically.

The information contained in these log records is synthesized into one of several reports which help users determine what their busy files are. The specific algorithm used will be covered later, but briefly our method is to gather all of the type 5 (& 105) records for each unique file on the system, and total the number of records processed, the number of blocks processed as well as the count of type 5 (& 105) records encountered.

This gives us a measure of:

- the application activity - number of records processed,
- the physical I/O - number of blocks processed (refer to note above)
- the number of FOPEN's - type 5 and type 105 record count.
- the average and maximum size of each file (XL only)

Each of these measures gives us a slightly different measure of the relative use of the files, so our method allows us to choose the top twenty percent of the files judged to be busiest on any of these values. For that matter, we may choose any percentage from one to one hundred. Once these busy files have been identified, we can begin to optimize these files knowing that any improvements we make will have the maximum effect on the overall system performance.

Advantages / Disadvantages

There are several other methods which could have been used in an attempt to determine which files are our busiest. These include direct monitoring of the I/O on the system, embedded measuring tools and alternative reporting schemes using log records. The method used here has the following advantages:

- The MPE logging facility is universal across the 3000 product line and does not require a specific machine or MPE level (true, the logfile format changed from MPE IV to MPE V to MPE XL, but REDWOOD understands all three formats)

- No special programming or capabilities are required (REDWOOD does not need PM capability)
- Since the method analyzes log files which have been closed, it may be run during "off hours" or on a separate machine, thus there is no effect on the system caused by the tool itself (other than the negligible overhead of enabling logging itself). REDWOOD can be used on an MPE XL machine to analyze logfiles :RESTOREd from a Classic machine.
- We have several distinct measures of usage: logical & "physical" I/O counts, file size (XL only), and file-open counts
- This method is not restricted to a particular file type or structure, although its usefulness may not be as great with some
- The only necessary modification to the system configuration is that logging be enabled for FCLOSE's (type 5 or 105)

The disadvantages to this approach are as follows:

- On MPE XL, we cannot determine the number of physical I/Os that occurred. MPE XL will "post" (write to disc) pages of files at its leisure. Thus, a dozen FWRITEs to the same record may result in anywhere from 0 to 12 disc writes.
- On MPE XL, any "mapped access" to a file is not reflected in the file close record. Thus, access by TurboImage is not seen (except for the file close record itself).
- On MPE V, although it does give you physical I/O, these figures are not related to any figure such as specific IMAGE calls, etc.
- This tool works well for summary reporting of disc activity, however, in some cases the real concern may center on "burst" I/O activity; i.e., the total number of I/O's spread across the day is small, but within 2 seconds after ENTER is hit, the activity is concentrated.
- There is no mechanism for determining the I/O rate for a period of time.
- These records can not be used to summarize disc activity per device since the LDEV is only for the first extent; not all extents of a file are required to reside on a single device.
- The number of blocks processed does not equal the number of physical I/O's if one of two cases is true:
  a. a rewind on a variable record file sets the number to zero
  b. files opened with multi-rec which also have blocks which end on a sector boundary may access multiple blocks in a single I/O, the number is higher than the actual number of physical I/O's
- On MPE XL, type 105 records do not contain a "blocks transferred" count, so REDWOOD synthesizes a value by assuming a blocking factor of 1.
- On MPE XL, type 105 records contain the number of logical reads and the number of logical writes. On MPE V, type 5 records contain only the sum of these two values. REDWOOD does not yet have the ability to separate the read count and write count. Instead, it always adds them together.

Capabilities

Program capabilities required include IA, BA, DS and PH. No special user capabilities are required.

Usage

Invoke REDWOOD using the supplied UDC or with the RUN command detailed below.

- UDC

  ```
  :REDWOOD [parm=#]
  ```

- RUN

  ```
  :RUN REDWOOD.PUB.LPSTOOLS;PARM=#
  ```

parm=# is used to change the max default number of records that REDWOOD can process in a single summary log file. The number entered for **parm** is multiplied by 1000 to obtain the new max default value (DEFAULT: 40000). For example:

```
REDWOOD 90
```

Or:

```
RUN REDWOOD.PUB.LPSTOOLS;PARM=90
```

(Sets default to 90,000 records)

Command Summary

REDWOOD has several commands which when executed in a particular sequence will produce a summary file which can then be re-sorted and listed for several different reports. The commands are listed in the next table.

NOTE Portions of the command codes are printed in uppercase to denote the part of the command that ACAP requires in order to distinguish one command from another. However, the commands themselves are not case-sensitive.

Table 14.1 *REDWOOD Commands*

Command Code	Description
CReate	Creates a "summary log" file
EXClude	Sets/resets exclusion options

Command Code	Description
Exit	Terminates REDWOOD
HELP	Gets full help information
List	Sorts and reports a "summary log" file
LP	Directs LIST output to a printer
RESET	Resets various options
SET	Sets various options
TERMinal	Directs LIST output to terminal
//	Terminates REDWOOD (synonym of Exit)
?	Gets help information (subset of HELP)

Command Definitions

Depending on which commands you use, REDWOOD produces a summary file which can be re-sorted and listed for several different reports. REDWOOD commands include primary functions, options, exit procedures and Help. These are discussed in detail next.

CReate

CREATE produces a summary file of the FCLOSE records from one or more log files. The log file (s) used are read sequentially and all type 5 and 105 (FCLOSE) records for disc files (subtype 0) are extracted. These records are then sorted by the file formal designator (**file.group.account**) to group all records for the same file. EDITOR work files of the form **Knnnnnnn**, where **nnnnnnn** is a valid seven-digit number, are transformed to have file names of the form, **K#######**, to group all k-files for each group/account into one record. (This is controlled by the [RE]SET EDITOR command.) Similarly, FSEDIT work files (whose names are of the form **F#######**) are gathered together under the control of the [RE]SET FSEDIT command.

This temporary sort file is then read sequentially and a summary file is built containing one record for each unique formal designator. This record contains information including: the device number (or pseudo-LDEV for MPE XL), total number of records processed, total number of blocks processed, FCLOSE count and an indicator for whether the device number was the same for all FCLOSE's. If this indicator is set to TRUE, then at there was at least one record which contained a logical device different from the other records for that file. This indicates that the file has moved, possibly due to a purge and re-create.

Since the log files may have been stored from the test system and restored to an account or group other than PUB.SYS, the CREATE command allows the user to override the default group and account (PUB.SYS) for the log file(s) to be analyzed. Once the group and account has been established, the four digit number of the first log file is entered and then the four digit number of

the ending log file is entered if different from the first. Once these numbers are in, REDWOOD requests the name of a summary file which it will attempt to create to hold the summary records for each FCLOSE'd file.

After the log file range has been specified, REDWOOD reads through each log file whose number is in the desired range and extracts all the file close records, storing them in the summary file.

NOTE In an effort to let you know that REDWOOD is alive and well, it will print a dot (.) every 1000 records read from a log file. Additionally, it will print an asterisk (*) for every 1000 file close records found.

EXCLude

This command has the following syntax:

```
EXCLUDE [ZERO] [NONE] [PERM] [DEFault] [LDEV #] [SMALL #blocks]
        [NONZERO] [BIGsectors #]
```

REDWOOD has the ability to "exclude" records from being considered based on a variety of criteria. The exclusion is checked as each record is read from a log file.

BIGsectors #	Instructs REDWOOD to exclude any file-close records for files larger than the specified number of sectors.
DEFault	This is equivalent to entering the following commands: REDWOOD: SET EDITOR FSEDIT NOGMULTI MERGEDOMAINS ARITRAP REDWOOD: RESET SORTSCR REDWOOD: EXCLUDE ZERO SMALL=0 BIGSEC=0 (resets EXCLUDE PERM NONZERO)
LDEV #	Instructs REDWOOD to exclude any file-close records that were for files whose file label was on the specified ldev. If you want to exclude more than one LDEV, you can do so with multiple EXCLUDE LDEV commands.
NONE	Instructs REDWOOD to not exclude ANY records from the summary file.
NONZERO	Instructs REDWOOD to exclude all file-close records that had more than zero (0) blocks transferred. This provides a way of seeing only those files that were opened and not used.
PERM	Instructs REDWOOD to exclude all file-close records that refer to permanent disc files.

SMALL #blocks	Instructs REDWOOD to exclude any file-close records for files with less blocks transferred than the specified number.
ZERO	Instructs REDWOOD to exclude all file-close records that said zero (0) blocks were transferred.

Exit

The EXIT command closes all open files and ends the program execution.

A :EOD entered at any prompt will also terminate REDWOOD.

List

The LIST command will sort and report on the records found in a summary file, whose name can be directly entered or a return can be used to indicate that the same summary file will be used again. The file is sorted in one of eight or nine different manners. When this sorted file is then listed and totalled, the files are in an order such that the "busiest" files are listed first. The user can choose to list just the busiest ten percent of the system's files. If the sort key chosen stays the same between two LISTings of the same summary file, then the sort is not executed to save time.

The possible sort options available are:

1 RECORDS processed
2 BLOCKS processed (only for MPE V & IV logfiles)
3 FCLOSE count
4 REC/BLK ratio
5 REC/BLK ratio (exclude probable NOBUF)
6 File Name (A.G.F)
7 File Name (F.G.A)
8 Average Size
9 Maximum Size

The LIST command produces a report with a header like the following:

```
                                                    %   J        Maximum Average
File    .Group .Account Ldn  # Records  #Fcloses  Cum  S   R/B  sectors sectors
---------------------- ---  ---------  --------  ---  -   ---  ------- -------
```

Figure 14.3 *LIST Command Report Header*

Each of these fields is described below.

Table 14.2 *Header Fields*

Field	Description
File.Group.Account	This field is the name of the file, as found in the file close log record. If the "File" field is "nameless", then this indicates that a nameless file was seen. Such files are used as scratch files by programs. A file name of **K#######** means that one (or more) EDITOR/3000 workfiles were seen for the group.account shown. Files beginning with a **K** and having 7 trailing digits in their name are lumped together if a SET EDITOR has been seen (default). A file name of **F#######** means that one (or more) FSEDIT workfiles were seen for the group.account shown. Files beginning with a **F** and having 7 trailing digits in their name are lumped together if a SET FSEDIT has been seen (default).
Ldn	The LDN column reports the approximate ldev that for each file. Note that the actual ldev is not given. Instead, the volume-set index is shown. (This is because MPE XL puts the index, and not the ldev, into the file close record.) For discs that belong to the system volume set (i.e.: MPEXL_SYSTEM_VOLUME_SET), the indices usually correspond exactly with the ldevs. The CI command :DSTAT ALL will report a list of ldevs and what volumes are mounted on them.
# Records	The # Records column reports how many records were read/written via the file system for each file. Any access to a file via mapped access will not be reflected in this number.
# Fcloses	The #Fcloses column reports how many file-close records were seen for each file.
% Cum	The %Cum column reports the cumulative percent that each line represents. If a summary file was sorted by RECORDS processed, and the first two lines had %Cum columns values of 15 and 20, then this means that the 15% of all records processed (for ALL files) were processed for the first file. And, of all the records processed, 5% were for the second file. Since 15 + 5 = 20, the %Cum column for the second file shows 20. If a cutoff percentage is entered (other than 100), then the listing stops after it gets to a line where the %Cum value matches (or exceeds) the cutoff value.

Field	Description
JS, J/S J S	The J/S column reports whether the file was used from jobs (J), sessions (S), both (" "), or only by system processes (?).
R/B	The R/B column reports the block factor for a file.
Maximum Sectors	The Maximum Sectors column reports how large was the largest version of each file.
Average Sectors	The Average Sectors column reports the average size of each file version for each file. This column holds a non-blank entry only when the average sectors is different from the maximum sectors.

Some columns in the LIST output have special characters to indicate various things. This section documents each of these characters.

*	An asterisk (*) after the Ldn column indicates that more than one ldev was seen for a file. This means that the file "moved" (or was purged and rebuilt) at least once during the period covered by the logfiles.
J	A **J** in the J/S column indicates that a file was used only by batch jobs.
S	A **S** in the J/S column indicates that a file was used only by interactive sessions.
Blank space	A blank space in the J/S column indicates that a file was used by jobs and sessions.
?	A question mark (?) in the J/S column indicates that a file was used by system processes, and not by jobs or sessions.

LP

The LP command opens a file with the formal name LPSLP, which defaults device=LP. All reports are sent to this file until the TERM or EXIT commands are used. A file equation may be used to redirect this file.

If a hard-copy is desired at the same time as an on-line report, the Toolbox standard command: SET COPYLP may be used instead of the LP command.

SCAN

The SCAN command acts like the CREATE command, except that no summary file is created. It is useful for scanning over log files for I/O errors.

SET | REset

These commands have the following syntax:

```
SET/RESET <option> [ , <option> ...]
<option>::=
    ARItrap
    CPU
    mergeDOMains
    EDITor
    FIRSTHour = #
    FSEDIT
    GMulti
    LASTHour = #
    LOGERRors
    NONZERO
    TEMPONLY
    ZERO
```

ARItrap	SET ARITRAP tells REDWOOD to pay attention to any internal arithmetic faults that might occur. RESET ARITRAP tells REDWOOD to ignore faults.
CPU	SET CPU tells REDWOOD to report its CPU usage at the end of handling a command.
mergeDOMains	SET mergeDOMains tells REDWOOD to treat old and new files with the same name as the same file. RESET MERGEDOMAINS tells REDWOOD to group all old files separately from all new files.
EDITor	SET EDITor tells REDWOOD to merge all EDIT/3000 temporary files (K#######) into one file.

FIRSTHour = #	SET FIRSTHour = # tells REDWOOD to ignore any file-close records that occurred before the specified hour of the day. The hour value must be in the range 0..23.
FSEDIT	SET FSEDIT tells REDWOOD to merge all FSEDIT temporary files into one file.
GMulti	SET GMulti tells REDWOOD to attempt to compensate for the MPE bug in reporting the number of blocks transferred for files that were opened with GMULTI (i.e.: message files). When compensation occurs, it is flagged as such in the summary report. RESET GMULTI disables the compensation.
LASTHour = #	SET LASTHour = # tells REDWOOD to ignore any file-close records that occurred after the specified hour of the day. The hour value must be in the range 0..23.
LOGERRors	SET LOGERRors tells REDWOOD that you want to see I/O errors that were in the logfile. RESET LOGERRORS tells REDWOOD to not report I/O errors.
NONZERO	SET NONZERO is a synonym for EXCLUDE NONZERO. RESET NONZERO disables excluding non-zero file-close records.
TEMPONLY	SET TEMPONLY tells REDWOOD that you want to exclude file-close records for permanent disc files. SET TEMPONLY is a synonym for EXCLUDE PERM. RESET TEMPONLY disables the exclusion.
ZERO	SET ZERO is a synonym for EXCLUDE ZERO. RESET ZERO disables excluding zero-block file-close records.

SORTSCR

The SORTSCR command causes REDWOOD to scan selected logfiles looking for file-close records for SORTSCR files. SORTSCR is the name of the scratch file used by the SORT subsystem. Unlike the CREATE command, the SORTSCR command does not build a summary file.

TERMinal

The TERM command closes the previous LPSLP file and directs the output from subsequent LIST commands to $STDLIST. This is the default case when the program is first run.

// and /

At most prompts, entering a single slash (/) will "bail out" of that question and return you to the outer command prompt. At any prompt, a "//" will act like the EXIT command.

TOOLBOX STANDARDS

The ToolBox collections from Lund Performance Solutions have a uniform user interface. As a result, in addition to the commands specific to each Toolbox tool, most tools allow the commands described in “TOOLBOX STANDARDS” on page 311.

REDWOOD Examples

The following example steps you through a typical REDWOOD session. The Max sort records at the time REDWOOD was invoked was 40,000.

```
Wolf:/LPSTOOLS/PUB: redwood
REDWOOD [2.3] - LPS Toolbox [A.09f]         (c) 1995 Lund Performance Solutions

For Help at the REDWOOD prompt enter   ?
This product is licensed to: ImageStats Demo

Max sort records (based on ;PARM=) are 40000
```

Figure 14.4 *Invoking a REDWOOD Session*

The first step almost always involves creating the Summary Log. This is demonstrated in the next figure.

```
Enter command (CR = CREATE): cr

Exclusions set:
   ZERO

Are the log files in PUB.SYS? (Default=Y) -->y<--

Want a LISTF of the log files? (Default=Y) -->y<--

FILENAME

LOG0032     LOG0033     LOG0034     LOG0035


Enter first log file number (CR=exit) : 32

Enter last log file number (CR =   32): 32

Enter name of new summary file (CR = SUMLOG): TESTLOG
Initializing sort...Reading log file...
LOG0032 opened ok (eof = 916)
       148 desirable fclose records found
```

Figure 14.5 *Creating a Summary Log*

Once the Summary Log is created you may specify reporting options. For this Summary Log (named TESTLOG by the user), there are 148 files closing actions that can be analyzed.

```
Found total of 148 desirable file-close records (for disk files).
Excluded 140 other file-close records.

Number of types of log records:
        #Records        Type
        --------        ------------------------------------
            288            5   :File Close
              1          101   :NM Log: System Up
             13          104   :NM Log: process termination
             11          111   :NM Log: llio
            577          115   :NM Log: console
              1          120   :NM Log: ncs spooling
             25          144   :NM Log: file open

Building summary file now...
89 summary records in file.
```

Figure 14.6 *Summary Log Report*

Using the LIST command to display record information, you can get basic count information. Additionally, REDWOOD displays a list of sort options you can use to create a report.

```
Enter command (CR = LIST): LIST

Enter name of summary file (CR = TESTLOG): TESTLOG

Summary created 09/10/01 from MPEXL log file 32
Log files dated: 09/05/01..09/10/01
Files of different domain (new/temp/old) are reported together.
Exclusion options:
        ZERO ... files opened & closed with no I/O are exclused.
        EDITOR = EDIT/3000 work files are merged as K########
        FSEDIT = FSEDIT work files are merged as F########
Counts:
        # files in summary =     89,    # records processed  =   6217
        # fcloses          =    148,    # blocks processed   =   6217
CPU = 3000/9xx

Sorted on: (unknown)

Sort on:
        1) RECORDS processed
        3) FCLOSE count
        4) REC/BLK ratio
        5) REC/BLK ratio (exclused probably NOBUF)
        6) File Name (A.G.F.)
        7) File Name (F.G.A.)
        8) Average Size
        9) Maximum Size
Enter sort type [1..9] (CR=1): 1/3,9
sorting summary file...
  (sort done)

Enter cutoff percentage, or # followed by number of files to list
(CR = 100%): 55%
```

Figure 14.7 *LIST Command*

The following output reports on 55% of the pool of records, from the largest on down. Sort options specified were the range 1 through 3 and option 9.

```
Will generate report with 80 characters per line.
REDWOOD 2.3 from Lund Performance Solutions  MON, SEP 10, 2001, 3:05 PM

Summary created 09/10/01 from MPEXL log file 32
Log files dated: 09/05/01..09/10/01
Files of different domain (new/temp/old) are reported together.
Exclusion options:
        ZERO ... files opened & closed with no I/O are excluded.
        EDITOR = EDIT/3000 work files are merged as K########
        FSEDIT = FSEDIT work files are merged as F########
Counts:
        # files in summary =     89,    # records processed  =   6217
        # fcloses          =    148,    # blocks processed   =   6217
CPU = 3000/9xx

Sorted on: RECORDS processed

Page # 1                                                 %   J        Maximum Average
File    .Group .Account Ldn  # Records  #Fcloses  Cum  S   R/B  sectors sectors
----------------------- ---  ---------  --------  ---  -   ---  ------- -------
SY910213.UDCS   .UTIL    1        990         4    15 S           160
CCA24861.TEST   .SPLASH  2        586         2    25 S           464      240
UDC1    .MISC   .MINER   3        518         2    33 S           144
TWF     .ASM    .SPLASH  1        482         1    41 S           387
SPLFROMS.ASM    .SPLASH  1        452         1    48 S           384
COMMAND .PUB    .SYS     3        408         2    55 S           208
                                --------    ------
Totals for 6 files:             3436          12
The files reported represent:     55%          8%
Totals for 89 files:            6217         148

Notes:
        The ldev column reports volume set indices, not the ldev for each file.
        (Although the ldevs and indices for disk in the
        MPEXL_SYSTEM_VOLUME_SET usually match up.)

                Symbol    Column#    Meaning
                ------    -------    -------
                   *         32      Logical device number changed between FCLOSE's.
                   S         56      File used only by sessions.
```

Figure 14.8 *Output Report*

REDWOOD Error Messages

Each REDWOOD error message is described in the following table.

Table 14.3 *REDWOOD Error Messages*

Message	Cause	Action
Illegal file name	An illegal file descriptor was entered.	Review HP documentation concerning valid filenames.

Message	Cause	Action
Illegal file title:	An invalid filename was given.	Review HP's documentation about file name descriptions.
Oops ... log files MUST be called LOG####[.group[.acct]]	Log files can be moved into groups other than PUB.SYS.	Log files names should not be altered; rename to original name given by MPE XL.
Sorry, that is a damage summary file ... you should probably purge it.	The summary file is damaged and cannot be used.	Use REDWOOD to reconstruct the summary file from individual log files.
That file has an improper file code.	REDWOOD summary files use a file code of 1001.	Use LISTF to locate summary files.
Unknown SET/RESET option:	An unknown REDWOOD set/reset option was given.	Valid choices for REDWOOD's set/reset options are: ARITRAP, CPU, MERGEDOMAINS, EDITOR, FIRSTHOUR, FSEDIT, GMULTI, LASTHOUR, LOGERRORS, NONZERO, TEMPONLY, ZERO.

15

THE REP TOOL

This tool provides a fast, easy, and consistent way for copying files on MPE/iX. REP copies TurboIMAGE databases and KSAM files (for both native and compatibility modes), as well as standard MPE files.

Operation

File copying (next to file listings via LISTF) is perhaps one of the most common tasks anyone has to perform on the HP3000. REP can be used for virtually all of your file copying needs.

Standard MPE Files

REP has several options that can be used to tailor its operation to your exact needs. Option include: the initial number of extends, max extends, filecode, crunch, and xlcrunch.

Database Files

REP will copy entire TurboIMAGE databases (root file and datasets) just by specifying the root file. The option ROOTONLY will make REP copy only the root file and not the datasets. The DBSTORE option can be used to update the database timestamp and store flag.

Batch

When REP is run in batch, then it assumes that you always want to purge the old copy of the destination file, should one exist.

Capabilities

Program capabilities required include IA, BA, PM, DS and PH. No special user capabilities are required to use REP.

Usage

REP can be run from the supplied UDC or with a fully-qualified RUN statement.

- UDC

```
REP [<inputfile> [[,] <outputfile>] [-] [<option>[,]...] [;parm=#]
```

- RUN

```
RUN REP.PUB.LPSTOOLS;info="[<inputfile> [[,] <outputfile>] [-]
                           [<option>[,]...]" [;parm=#]
```

The "-" is optional when both an inputfile and an outputfile are present, and is required to signify the start of the option list if an output file is omitted.

In both modes, if the input file name has a group (or group & account) specified, then the default output file name is the input name without the group.account. For example, if the input file name is **FOO.FIE.FUM**, then the default output file name is **FOO.logongroup.logonacct** and will be used as the output file name if you hit "return" when prompted.

If you are using NS/3000, the inputfile and outputfile file names are allowed to have nodenames (i.e.: **rep fun:fozzie, localfun**).

File equations can be used for either the input or output files (or both).

Options Summary

The following table provides a summary description od REP options, which can be used to quickly locate the command that suits the task at hand. Detailed information on each option is provided in the next section.

Table 15.1 *REP Options*

Option	Description
CODE	Specifies filecode for output file.
[NO]CRUNCH	Closes output file with crunch disposition.
DBSTORE	Updates timestamp and store flag database copies.
DELAY	Pause during copy operation.
DEVICE	Specifies outfile device.
DISABLESTAR	Tells REP not to print "*" in quiet mode.
DOTS	Displays progress "dots" on screen during copy.
EXTENTS	Initial extents for output file.
FILECODE	Specifies filecode for output file.
KEYFILE	New keyfile for KSAM file copies.

Option	Description
LDEV	Specifies disk drive for output file.
LOCAL	Use logon group and account names for output file.
MAXETENTS	Specifies maximum number of extents for output file.
NOKSAM	Copies KSAM file as non-KSAM file.
NULL	Does not perform copy, creates output file.
[NO]PURGE	Overwrites (purges) output file if it exists.
[NO]QUIET	Suppresses some messages.
[NO]ROOTONLY	Only copies rootfile of database.
[NO]TEMP	Creates output file in temporary domain.
[NO]TIMES	Displays CPU usage.
[NO]XLCRUNCH	Closes output file with xlcrunch disposition.
YES	Overwrites (purges) output file if it exists.

The following PARM bits are recognized by REP as synonyms for some of the keyword options:

Table 15.2 *PARM bits for REP*

Bit#	PARM	Equivalent Option	Meaning
15	1	PURGE	You want to purge OLD destination files.
14	2	LOCAL	Target file is the same file-part name, but in logon group and account.
13	4	QUIET	You don't like informative messages.
12	8	ROOTONLY	You don't want to copy entire databases.
11	16	TIMES	You want CPU and elapsed times shown
10	reserved		
9	64	DOTS	You want to see progress dots ".".
8 .. 0	reserved, should be 0		

PARM options may be combined by adding their respective values.

Options Definitions

Following is a detailed description of each of the REP options.

CODE

This option has the following syntax:

```
CODE = <filecode>
```

By default, REP gives the output file the same filecode as the input file. However, the filecode can be specified as another type if you wish. See Appendix B for a list of the most commonly used filecodes. (This option and the FILECODE option are identical in operation.)

[NO]CRUNCH

This option causes the output file to be closed with "crunch" disposition, which sets the file limit to the file eof.

NOCRUNCH (default) tells REP not to crunch the output file.

DBSTORE

This option tells REP to update the timestamp & store flag that are stored in a database's rootfile. Normally, this operation is done by the DBSTORE program when a database is stored off to tape. However, users may want these flags when a database is duplicated.

The timestamp is used by DBRECOV to help identify the correspondence between logfiles and backup databases. See the *TurboIMAGE/iX Database Management System Reference Manual* - DBSTORE section for more details.

DELAY

This option has the following syntax:

```
DELAY [=#]
```

During large copy operations this statement will cause REP to pause between chunks (where a chunk is part of a file that was copied). A desirable side effect to using this option is that it keeps REP operations from starving other processes that are running at that time. Of course, using this option will cause the overall time required to copy the file to increase. However, other users on the system will be happy that you did! The delay is specified in milliseconds. If the DELAY option is specified without a delay number, 10 milliseconds will be used.

DEVICE

The DEVICE option has the following syntax:

```
DEVice = devicename
```

Specifies the devicename that the output file should be placed on.

DISABLESTAR

When REP is used in QUIET mode it displays an asterisk (*) to indicate that it has finished copying the file. However, there may be times that this is undesirable, so if you specify this option the asterisk (*) will not be displayed at the end of the copy. Situations when it is desirable include calling REP programmatically to perform a copy and you wish to keep display output to a minimum.

DOTS

Specify this option if you would like to display progress dots "." as it proceeds through a copy operation. The default option setting is to not print progress dots.

EXTENTS

This option has the following syntax:

```
EXTENTS = #
```

Specifies the initial number of extents to allocate for the output file.

FILECODE

This option has the following syntax:

```
FILECODE = <filecode>
```

REP normally gives the output file the same filecode as the input file. However, the filecode can be specified to another type if you wish. See Appendix B for a list of most commonly used filecodes. (This option and the CODE option are identical in operation.)

KEYFILE

This option has the following syntax:

```
KEYFILE = filepart
```

Specifies the name of the new key file for KSAM output files. The filepart may not have a group or account.

LDEV

This option has the following syntax:

```
LDEV = ldev#
```

Specifies the disk drive the output file should be placed on.

LOCAL

Output filename is the same file-part name, but in the logon group & account.

MAXEXTENTS

This option has the following syntax:

```
MAXEXTENTS = #
```

Used to specify the maximum number of extents for the output file.

NOKSAM

Tells REP to copy a KSAM file as a non-KSAM file.

NULL

This option causes the output file to be created and the space to be allocated, however no actual file transfer is performed.

[NO]PURGE

PURGE tells REP that it is ok to purge any existing file with the same name as the output file. NOPURGE tells REP that it must ask for permission to purge.

[NO]QUIET

QUIET Tells REP to suppress some output messages.

[NO]ROOTONLY

ROOTONLY tells REP to not copy an entire database when it is given a root file. NOROOTONLY (default) tells REP to automatically copy the entire database when given a root file.

[NO]TEMP

TEMP tells REP to leave the output file in the job/session temp domain. NOTEMP tells REP that the output file should be permanent. The default is TEMP if the input file is a temporary file, and NOTEMP otherwise.

[NO]TIMES

TIMES tells REP to report the CPU and Elapsed time that a file copy took. NOTIMES suppresses the time report.

[NO]XLCRUNCH

XLCRUNCH tells REP to close the output file with the new "xlcrunch" disposition. This tells the file system to discard any unused disc space past the EOF but does not set the file limit down to the file eof. Thus, an xlcrunched file is expandable later.

YES

The YES option is a synonym for PURGE. YES tells REP that it is ok to purge old files with the same name as the output file.

TOOLBOX STANDARDS

The ToolBox collections from Lund Performance Solutions have a uniform user interface. As a result, in addition to the commands specific to each Toolbox tool, most tools allow the commands described in “TOOLBOX STANDARDS” on page 311.

REP Examples

Following are examples of the REP tool.

```
Wolf:/LPSTOOLS/PUB: run rep;info="testdoc.fum newfile filecode+5555"

REP [2.8] - LPS Toolbox [A.09f]             (c) 1995 Lund Performance Solutions

This product is licensed to: ImageStats Demo
From testdoc.fum, to newfile
(filecode := 5555)
      # of records copied = 1

END OF PROGRAM
```

Figure 15.1 *Specify Different Filecode*

```
Wolf:/LPSTOOLS/PUB: run rep;info="testfs.spl -xlcrunch"

REP [2.8] - LPS Toolbox [A.09f]             (c) 1995 Lund Performance Solutions

This product is licensed to: ImageStats Demo
To := testfs.PUB.LPSTOOLS
From testfs.spl, to testfs.PUB.LPSTOOLS
      # of records copied = 253

END OF PROGRAM
```

Figure 15.2 *XLCRUNCH Copy*

The next example copies TESTDOC.FUM.LPSTOOLS to TESTDOC.PUB.LPSTOOLS.

```
Wolf:/LPSTOOLS/PUB: run rep;info="testdoc.fum testdoc"

REP [2.8] - LPS Toolbox [A.09f]             (c) 1995 Lund Performance Solutions

This product is licensed to: ImageStats Demo
From testdoc.fum, to testdoc
      # of records copied = 1

END OF PROGRAM
```

Figure 15.3 *REP Example 1*

The following example copies FOO.FUM.LPSTOOLS to FOO.PUB.LPSTOOLS.

If interactive, and FOO.logongroup.logonacct (in this case, FUM.LPSTOOLS) already exists, REP will ask for permission to purge it.

```
Wolf:/LPSTOOLS/PUB: run rep;info="foo.fum"

REP [2.8] - LPS Toolbox [A.09f]             (c) 1995 Lund Performance Solutions

This product is licensed to: ImageStats Demo
To := foo.PUB.LPSTOOLS
From foo.fum, to foo.PUB.LPSTOOLS
      # of records copied = 2

END OF PROGRAM
```

Figure 15.4 *REP Example 2*

The next example copies FOO.FUM.LPSTOOLS to FOO.PUB.LPSTOOLS, automatically purging FOO.FUM.LPSTOOLS, if it existed.

```
Wolf:/LPSTOOLS/PUB: run rep;info="foo.fum";parm=1

REP [2.8] - LPS Toolbox [A.09f]             (c) 1995 Lund Performance Solutions

This product is licensed to: ImageStats Demo
To := foo.PUB.LPSTOOLS
...purged old.
From foo.fum, to foo.PUB.LPSTOOLS
      # of records copied = 2

END OF PROGRAM
```

Figure 15.5 *REP Example 3*

The next example copies FOO.FUM.LPSTOOLS to FOO.PUB.LPSTOOLS, automatically purging FOO.logongroup.logonacct, if it existed.

```
Wolf:/LPSTOOLS/PUB: run rep;info="foo.fum - YES"

REP [2.8] - LPS Toolbox [A.09f]             (c) 1995 Lund Performance Solutions

This product is licensed to: ImageStats Demo
To := foo.PUB.LPSTOOLS
...purged old.
From foo.fum, to foo.PUB.LPSTOOLS
      # of records copied = 2

END OF PROGRAM
```

Figure 15.6 *REP Example 4*

The next example copies FOO.FUM.LPSTOOLS to FOO.PUB.LPSTOOLS, automatically purging FOO.FUM.LPSTOOLS, if it existed.

```
Wolf:/LPSTOOLS/PUB: run rep;info="foo.fum - YES"

REP [2.8] - LPS Toolbox [A.09f]             (c) 1995 Lund Performance Solutions

This product is licensed to: ImageStats Demo
To := foo.PUB.LPSTOOLS
...purged old.
From foo.fum, to foo.PUB.LPSTOOLS
      # of records copied = 2

END OF PROGRAM
```

Figure 15.7 *REP Example 5*

REP Error Messages

Each REP error message is described in the following table.

Table 15.3 *REP Error Messages*

Message	Cause	Action
Copying from $NULL is illegal!	Bad source (i.e., FROM) filename.	The FROM file must be a file that exists in the permanent or temporary domain.

Message	Cause	Action
Identical FROM file and TO file names are not allowed!	Bad choice of name for either the FROM or TO file.	Change either the FROM or the TO filenames to a different name.
IMAGE root files and datasets may not have lockwords.	Lockwords were discovered on either the Image rootfile or dataset.	REP does not support copying Image files with Lockwords.
TO file may not be a system file (except $NULL or $NEWPASS).	User specified a destination filename that was neither $NULL nor NEWPASS.	Enter a valid destination filename.

16

THE SHOT TOOL

SHOT provides information on what processes are active on the machine. It provides a snapSHOT of process activity, and allows the priority of processes to be adjusted. SHOT can also be used to alter the state of a process.

Operation

SHOT's primary purpose is to show you processes that are interesting. At startup, "interesting" is defined as "processes that have used CPU since the last time you hit **Return**". This is called the DELTA command, and is executed by default when you hit **Return**. Thus, if you run SHOT and then just hit **Return**, **Return**, ..., each time SHOT will show you the processes that have used some CPU since the prior **Return**. See SHOT's ADM command and Process-Display section for details.

Viewing System Activity

The display of just CPU-using processes is called a "delta" display. (The "delta", or difference between the current CPU usage and the prior CPU usage is non-0.)

The ALL command displays all processes, regardless of their CPU usage.

You can restrict the display of active processes shown by the delta command in several ways:

- only show processes using more than a specified percentage of the CPU (use: **%n** i.e.,: %5 will only show processes that have used at least 5% of a CPU in the recent interval)
- exclude specified PINs (via the EXCLUDEPINS command)
- exclude processes waiting on specified wait events (i.e., EXCLUDEWAITS BREAK)

The "TREE" command shows all processes, according to their tree structure (parent/child), and ignores exclusion rules.

In the DELTA and ALL commands, the format of the output can be controlled via the ADM (Automatic Display Mode) command. For example, you can add a column to show the number of transient disk pages used by a process by saying: `ADM + SECPAGES`

The command `ADM ?` will list all available ADM fields.

The HIGHLIGHT command can be used to tell SHOT to highlight certain processes (i.e., HIGHLIGHT USER @.SYS)

You can control a process using:

ABORTCONNS, ABORTPROCIO, BREAK, KILL, PRIORITY, RESUME, SUSPEND

You can get information about a single process using:

ANCESTORS, CPU, DAD, FAMILY, JS, PIN, TP, TRACE, TREE

In addition to showing processes, SHOT has several other kinds of displays it can do:

- display jobs and/or sessions
- display job/session tables (JOBINFO)
- display jobqueue information (LISTJOBQ)
- display a list of different programs currently running (and the number of processes using each program)
- report system uptime (UPTIME)
- display a summary of why processes are waiting (SUMWAIT)
- display information about locked SIRs
- display information about PIN and VSM tables (TABLES).

Altering System Activity

In addition to viewing process information, you can alter some aspects of process activity. SHOT's BREAK and RESUME commands are used to suspend and reactivate processes by PIN. The PRIORITY command is used to change the CPU queue in which a process is executing.

NOTE Placing a process in the A or B queue could "take over" the system and cause a system halt. The A and B queues are typically only used by system processes.

SHOT's KILL command is used for terminating a single process. The KILL command is especially useful for processes which seem to be "hung".

Capabilities

Program capabilities include IA, BA, PM, DS and PH. User PM capability is required to use the STACK TRACE or DEBUG commands. User SM is required to use the ABORTCONNS, ABORTPROCIO, BREAK, RESUME or KILL commands. OP, SM or PM is required to use the PRIORITY command.

Usage

SHOT can be run either from the supplied UDC or from a fully-qualified RUN statement.

- UDC

  ```
  SHOT <command>
  ```

- RUN

  ```
  RUN SHOT.PUB.LPSTOOLS;INFO="<command>"
  ```

When you run SHOT with a command parameter, it executes that command and then automatically terminates.

The SHOT Process Display

The SHOT display is something you'll want to know about in order to understand the information being presented. This section explains the display components. The first line is a Time & Date header, and it is followed with several user-selectable fields. The following section describes these fields in detail.

The header for the default display format for SHOT is:

```
? Pin  Cpu   Process Name           Pri Q Delta  %  Ic Wait State
- --- ------ ---------------------- --- - ----- --- -- ----------
```

Figure 16.1 *SHOT - Header for the default display format*

Display column: ?

ADM field: STATE

The Process State is denoted by a question mark (**?**) in the SHOT header line. This shows the state of a process when the state is something other than alive.

The Process State are described in the next table:

Table 16.1 *SHOT - Process States*

Character	State	Description
?	unknown	PCB for process has not been allocated
d	dying	Process is beginning to terminate
x	dead	Process terminated, PCB not deallocated
<blank>	alive	Process is running
i	initiating	Process being born
u	unborn	Process just started being created

Display column: Pin

ADM field: PIN

The **PIN** is the Process Identification Number for a process. This is a 16-bit number which is reused when a process terminates.

See also: PID

Display column: CPU

ADM field: CPU

The **CPU** column reports the total processor time (CPU) used by a process since it started. The values shown are usually in milliseconds, but will have a letter prefix if they have used more CPU than can be displayed without overflowing the column width (which defaults to 6 characters, or 999,999 milliseconds). The prefixes are: **s** for seconds, **m** for minutes, **h** for hours, **D** for days, and **W** for weeks.

Display column: Process Name

ADM field: NAME

The **Process Name** column shows the name of the process running. The majority of processes are programs (files with filecode of PROG or NMPRG) that were :RUN (or started with the CREATE or CREATEPROCESS intrinsic). A few processes are started by privileged code pointing to a procedure in either NL.PUB.SYS or SL.PUB.SYS and saying: Start!. These processes are referred to as being *procreated*. These processes do not have normal names (i.e.: EDITOR.PUB.SYS) because there is no associated program file. SHOT tries to determines the name of the original procedure that was started and, if successful, displays the first 32 characters (or so) of the procedure name. If unsuccessful, the procedure address is shown in hexadecimal as a space and offset (i.e.: $a.4b2d90).

NOTE SHOT is unable to determine the names of procedures started from SL.PUB.SYS. It indicates these as **(CM Procedure)**.

Display column: Pri

ADM field: PRI

The **Pri** column shows the current priority of a process. The priority is a value in the range 0..255, with 0 being the highest priority. The CI's :SHOWQ command reports the base and limit priority values for the C, D, and E scheduling queues.

Display column: Q

ADM field: QUEUE

The **Q** column reports what scheduling queue a process is in.

The possible queues are:

Table 16.2 *Queue Codes*

Code	Description
A	Process is in the **A** subqueue
B	Process is in the **B** subqueue
C	Process is in the **C** subqueue
D	Process is in the **D** subqueue
E	Process is in the **E** subqueue

NOTE As of MPE XL 2.1, it is possible to put a process into four apparently new queues called **BM**, **CM**, **DM**, and **EM**. In fact, MPE does not have four new queues. Instead, when a process is placed into one of these **queues** (with the ALTPROC command), it is simply placed at the base of the BS, CS, DS, or ES queue and is marked as a system process so that its priority will not degrade over time.

Display column: Delta

ADM field: DELTA

The **Delta** column shows the amount of CPU used by a process since the last time a process display was shown. The values in this field are typically in milliseconds, but will have a suffix of **s** for seconds, **m** for minutes, **h** for hours, **D** for days, and **W** for weeks.

NOTE The CPU delta value for JSMAIN processes (those that are JSMAIN.PUB.SYS) will sometimes be negative. Whenever a job or session starts (and runs), it is assigned a JSMAIN process, which will use a small amount of CPU time in getting the job or session started. When the job or session terminates, the JSMAIN process uses a little more CPU time, but then sometimes stays around in *idle* mode, waiting to be assigned to a new job or session. When JOB.PUB.SYS (the parent of all jobs) or SESSION.PUB.SYS (the parent of all sessions) chooses that JSMAIN for a new job or session, JOB or SESSION resets the JSMAIN's total CPU time counter back to about 1/2 second of CPU. If SHOT noted the CPU total for a JSMAIN before and after this *reset*, the process will look like it used a negative amount of CPU time!

Display column: %

ADM field: PERCENT

The **%** column shows the approximate percentage of available CPU that a process has used since the last time a process display was shown. Due to the way SHOT determines CPU usage, it is possible that the sum of all percentages shown might exceed 100%, particularly if the previous process display was done very recently.

Display column: Ic

ADM field: EXECMODE

The **Ic** column shows the Initial and Current modes for a program. The first character of this column is: C, N, O, or P. The second character of this column is: n or c.

Table 16.3 *Execution Mode Codes*

Code	Description
Cn	Process is a Compatibility Mode program, currently in Native Mode
Cc	Process is a Compatibility Mode program, currently in Compatibility Mode
Nn	Process is a Native Mode program, currently in Native Mode
Nc	Process is a Native Mode program, currently in Compatibility Mode
Oc	Process is a Compatibility Mode program (OCT'ed), currently in Compatibility Mode
On	Process is a Compatibility Mode program (OCT'ed), currently in Native Mode
Pn	Process is a POSIX Native Mode program, currently in Native Mode
Pc	Process is a POSIX Native Mode program, currently in Compatibility Mode

NOTE Processes that are *procreated* native mode procedures will have an **n** instead of **N** in the Initial column

Display column: Wait State

ADM field: WAIT

The **Wait State** column reports why a process is waiting. On a single CPU machine, every process (except SHOT itself) should either be waiting for something to happen (i.e.: a page to be read from disc) or should be **READY** to run. On a multiple CPU machine, several processes (in addition to SHOT) might be **EXECUTING** at the same time.

The following is in order of ADm field names (except the fields from the default display format, explained above).

Display column: BlkR

ADM field: BLOCKEDR

The **BlkR** column shows the numeric code indicating why a process is blocked (if it is), called the *blocked reason*. The same information is usually available in pseudo-english in the WAIT column.

Display column: Caps

ADM field: CAPABILITY

The **Caps** column shows the capability mask for a process. This is a 32-bit value, with bits meaning:

Table 16.4 *Capability Bits*

Cap	Bit#	Meaning (if bit on)
SM	0	System Manager
AM	1	Account Manager
AL	2	Account Librarian
GL	3	Group Librarian
DI	4	Diagnostician
OP	5	Operator
CV	6	Create Volumes
UV	7	Use Volumes
LG	8	User Logging
SP	9	System process handling
PS	10	Programatic Sessions
NA	11	Network Administrator

Cap	Bit#	Meaning (if bit on)
NM	12	Node Manager
CS	13	Communication Subsystem
ND	14	Use Non-Sharable devices
SF	15	Save Files
..	16..22	(reserved)
BA	23	BAtch access

Display column: StXDS

ADM field: CMSTACK

The **StXDS** column reports various values for the Compatibility Mode (CM) environment of a process. If width permits, the following is shown:

DBXDS	is the Data Segment number of the Compatibility Mode DB (usually the same as the CM Stack, unless in split-stack mode).
StXDS	is the Data Segment number of the CM Stack.
S	is the top-of-CM-stack register. This may not be accurate for processes currently executing on another CPU.
Z	is the limit-of-CM-stack register.

Display column: S Z

ADM field: CMSZ

The **S Z** column reports the value for the CM S and Z registers of a process. It is a subset of the information provided by the ADM-CMSTACK field.

S	is the top-of-CM-stack register. This may not be accurate for processes currently executing on another CPU.
Z	is the limit-of-CM-stack register.

Display column: Cflts

ADM field: CFAULTS

The **Cflts** column shows the total number of code page faults for each process. A **code page fault** was an attempt to access an instruction, but the virtual page was not in memory.

Display column: CMTIME

ADM field: CMTIME

The **CMTIME** column reports the total processor time (CPU) used by a process while in CM (Compatibility Mode).

See ADM-CPU for a description of the display format.

NOTE The CMTIME value is tracked only while the Measurement Interface (MI) is enabled. SHOT does not enable the MI. SOS/3000 and Glance/XL are examples of performance tools that enable the Measurement Interface.

Display column: C

ADM field: CRITICAL

The **C** column shows the depth of nesting in **critical** mode for each process. A non-critical process will have a blank in this column.

NOTE Processes that are **critical** (have a non-0 value here) will ignore :ABORT, :ABORTJOB, and other KILL requests until they become non-critical.

Display column: Dbg

ADM field: DEBUG

The **Dbg** column has a **Yes** for each process that has the **;DEBUG** flag on. Normally, if a program is run with **;DEBUG**, the bit is on. However, MPE/iX seems to set this bit for a few other processes for undetermined reasons.

Display column: DelCFl

ADM field: DELCFAULTS

The **DelCFl** column shows the number of code page faults for each process since the last display. **Del** is short for Delta.

Display column: DelDFl

ADM field: DELDFAULTS

The **DelDFl** column shows the number of data page faults for each process since the last display. **Del** is short for Delta.

Display column: DelFl

ADM field: DELDFAULTS

The **DelDFl** column shows the number of data page faults for each process since the last display. **Del** is short for Delta.

Display column: Dflts

ADM field: DFAULTS

The **Dflts** column shows the total number of data page faults for each process. A **data page fault** was an attempt to access an data on a virtual page, but the page was not in memory.

Display column: DirtyPgs

ADM field: DIRTYPAGES

The **DirtyPgs** column reports the number of memory pages currently in use by a process that are **dirty** (or, recently modified). (A page is 4,096 bytes.) It includes only the pages in the space used by the Process Private data for the process (i.e., all pages whose Space ID matches the process's StkID (c.f.)).

Display column: Dispatcher

ADM field: DISPATCHER

The **Dispatcher** column shows a bunch of dispatcher flags for each process. Flags are **on** if a letter appears in the column, and are off if the column is blank.

Flags (in column order, left-to-right):

s	Process dispatcher state: (leftmost column) E = Executing R = Ready S = Short wait L = Long wait
=	Locked onto specific CPU (MPE/iX 6.5 and later)
M	ON_MAX_CPU
q	QUEUE_MANAGER
<	DECAYABLE_PROCESS
@	AT_LIMIT
A	ADJUST_QUANTUM
B	BEING_PDISABLED
D	DEAD_PROCESS
L	LOCALITY_PENDING

P	ALREADY_PREEMPTED
Q	TO_BE_QUIETSCED
S	SWAP_IN_DESIRED
T	TRANS_COMPLETE
b	IO_BOOST_DELAYED
d	DEFERRED_PREEMPT
i	CHANGE_IO_MODE
m	ARTIFICIAL_MEMBER
p	BLOCK_PAGE_FAULT
l	TO_BE_LAUNCHED
u	USER_PROCESS
a	ALREADY_SCAN

If you want fewer columns, remember that you can do things like:

```
ADM + DISPATCHER:4
```

which would display only the left most 4 columns of flags (s=Mq).

Display column: DPlimit

ADM field: DPLIMIT

The **DPlimit** column shows the maximum value that DP can grow to for each process. This value is also used by some parameter address checking code.

Display column: Dump

ADM field: DUMPARMED

The **Dump** column shows **Yes** if a :STACKDUMP is in effect for a process. This means that if the process aborts, a mini stack trace will be generated.

Display column: Entry

ADM field: ENTRY

The **Entry** column shows the virtual address of the entry point of a program. For example, the address of PROGRAM/START/main.

Display column: EUID Caps

ADM field: EUIDCAPS

The **EUID Caps** column shows the capability mask for a process based on the current effective user-ID (a POSIX concept). If a process has never set the effective user-ID, the value is 0.

Display column: F

ADM field: FORKED

The **F** column shows an **f** for any process that was created by a POSIX fork() call.

NOTE In MPE/iX 4.5, a process created by fork() appears to have no name. SHOT will usually show its name as a virtual address in space $a.

Display column: $Fork SP

ADM field: FORKSP

The **$Fork SP** column shows the value of the Stack Pointer (SP) at the point that the process was **forked**.

Display column: GlbDP lim

ADM field: GDPLIMIT

The **GlbDP lim** column shows the value of the **Global DP Limit** for each process. This value is the limit in the DP area that will be copied if the process does a fork().

Display column: HeapBase

ADM field: HEAPBASE

The **HeapBase** column shows the virtual address of the base of the Native Mode heap for the process.

NOTE This is a process-local address, so you will need the process' SR5 (STKSID) value to form a usable 64-bit virtual address.

Display column: HeapLimit

ADM field: HEAPLIMIT

The **HeapLimit** column shows the virtual address of the end (limit) of the Native Mode heap for the process.

NOTE This is a process-local address, so you will need the process' SR5 (STKSID) value to form a usable 64-bit virtual address.

Display column: #HeapPgs

ADM field: HEAPPAGES

The **HeapPages** column shows the number of disk pages used by the Native Mode heap for the process. Each page is 4,096 bytes.

Display column: HeapSize

ADM field: HEAPSIZE

The **HeapSize** column shows the size of the Native Mode heap for the process.

Display column: INFO=

ADM field: INFO

The **INFO=** column shows the first part of the INFO string for a process.

Display column: INFO ptr

ADM field: INFOPTR

The **INFO ptr** column shows address of the INFO string for a process.

Display column: IntPri

ADM field: INTPRI

The **IntPri** column shows the current priority of a process. Unlike the **Pri** column (which shows value in the range 0..255, where 0 is the highest), this column shows the internal value used by the dispatcher. This value ranges from 0..32767, where 0 is the lowest priority and 32767 is the highest priority. Whenever MPE (or a performance tool) report a priority in the **0..255** style, the value was obtained by converting it from the internal format with the formula:

old'pri := (32767d - internal'pri) / 128d;

Old format priority values can be converted to new format via:

internal'pri := (255d - old'pri) * 128d + 1d;

Display column: Job#

ADM field: JSNUM

The **Job#** column reports the job/session number that a process belongs to. By default, this column is suppressed. It can be requested by saying: SET MOST or SET ALL.

Display column: Killer

ADM field: KILLER

The **Killer** column reports the PIN that has sent a KILL request to the process (if any).

Display column: LSTX

ADM field: LSTX

The **LSTX** column reports the Loader Segment Table indeX used by each process. This is of little practical value.

Display column: MemPages

ADM field: MEMPAGES

The **MemPages** column reports the number of main memory pages currently in use by a process. A page is 4,096 bytes. It includes only the pages in the space used by the Process Private data for the process (i.e., all pages whose Space ID matches the process's StkID (c.f.)).

Display column: Parent

ADM field: PARENTPIN

The **Parent** column shows the PIN (Process Identification Number) of the parent of a process.

Display column: Parm=

ADM field: PARM

The **Parm=** column shows the value of the PARM parameter at the start of each process.

Display column: ProgrmCounter

ADM field: PC

The **ProgrmCounter** field shows the current program counter for a process. This will typically be an address within the kernel (in NL.PUB.SYS). For processes that are waiting on some event or other, it is typically the address of the MPE internal routine enable_int+$2c.

NOTE This field is fetched from the Task Control Block, and thus is slightly stale for processes that are currently executing (for a single CPU system, this would be just the SHOT process).

Display column: Pending Events

ADM field: PENDING

The **Pending Events** column reports what events are waiting to happen to the process. For example, if a control-Y has been pressed, but the process is inside FOPEN, the control-Y will be left pending until FOPEN is exited. The value reported is the subqueues on the Process Interrupt

Port that have messages queued to them. A control-Y is often on subqueue 2, and would be reported as **Int: [2]**.

Display column: %SB

ADM field: PERCENTSB

The **%SB** column shows the approximate percentage of available CPU that a process has used since bootup.

Display column: Pft

ADM field: PFT

The **Pft** column shows the **Page Fault Type** for each process. This is the object class of the virtual page last faulted on.

Display column: Pin/Reuse

ADM field: PID

The **PID** is the Process ID, an extended form of the PIN (Process Identification Number) for a process. A PID is a non-reusable number identifying a process.

When SHOT is asked to show PIDs instead of PINs, it uses the form: #1/#2, where #1 is the PIN portion of the PID and #2 is the **reuse** portion of the PID. The command SET PIDS changes the **Pin** column into the **Pid/Reu** column.

Display column: Plabel Table

ADM field: PLABTAB

The **Plabel Table** column shows the virtual address of the **Plabel table** for the process. This table contains information about dynamically loaded procedures. If the process has not loaded any procedures, the address is 0.

NOTE This is a process-local address, so you will need the process' SR5 (STKSID) value to form a usable 64-bit virtual address.

Display column: P

ADM field: POSIX

The **P** column has a **P** if the process is flagged as a POSIX process. One example, as of MPE/iX 5.0, is SH.HPBIN.SYS (the POSIX shell).

NOTE The **Ic** column should have a **P** for POSIX programs.

Display column: PosixPid

ADM field: POSIXPID

The **PosixPid** column shows the POSIX PID (Process Identification Number) of a process. This is closely related to the PID of a process, but is encoded differently. Currently, the lower 16 bits is the PIN, and the upper 16 bits is the reuse count.

Display column: ProtID

ADM field: PROTID

The **ProtID** column shows the Protection ID associated with the process' stack.

Display column: Ptype

ADM field: PTYPE

The **Ptype** column shows the process-type for each process. Possible types are: system, main, son, ucop, detach, user, and task. The SET PTYPE command allows processes to be excluded according to process-type. Typical uses of each process type are described in the next table:

Table 16.5 *Process Types*

Ptype Name / Number		Examples
User	0	SHOT.PUB.LPSTOOLS (created by a Son/User)
Son	1	SHOT.PUB.LPSTOOLS (top level :RUN)
Main	2	CI.PUB.SYS (top level CIs)
Task	3	(unused?)
System	4	PROGEN.PUB.SYS, pm_cleanup
Detach	5	DIAGMON.PUB.SYS
UCOP	6	JSMAIN.PUB.SYS

NOTE **Son** is short for **User, Son of Main**.

Display column: PxProgname

ADM field: PXPROGNAME

The **PxProgname** column shows the virtual address of the POSIX-format name for a process (if available).

Display column: SecPages

ADM field: SECPAGES

The **SecPages** column reports the number of secondary storage (disk) pages used by a process. A page is 4,096 bytes. It currently includes only the pages for the NM stack, CM stack, and NM heap. (This is the majority of the transient disk space used by a process).

Display column: StartTime

ADM field: STARTTIME

The **StartTime** column shows the time that the process started.

Display column: StkSID

ADM field: STKSID

The **StkSID** field reports the Space ID of the NM stack for a process.

Display column: #susp

ADM field: SUSPEND

The **#susp** column shows the current value of the TCB field **suspend_cnt** for a process.

Display column: S

ADM field: SWDEPTH

The **S** column shows the current nesting count of **switch** calls (both switch-to-NM and switch-to-CM). This is NOT a cumulative value, but reflects instead the number of switch markers you would see if a stacktrace for the process was performed at that instant. This column is useful in detecting NM programs that are still using CM code (either directly or by calling MPE intrinsics that are still implemented in CM). If the **Ic** column shows a process as **Nn**, then the **S** column should be a multiple of 2. If the **Ic** column shows a process as **Nc**, then the **S** column should be an odd number.

Display column: S

ADM field: SYSCODE

The **S** column shows the depth of nesting in **system code** for each process.

NOTE A process is not in **system code** merely because it calls a procedure in XL.PUB.SYS or NL.PUB.SYS.

System code is a process-local counter that is incremented by calling an internal routine, enter_system_code, and decremented by calling leave_system_code. Every intrinsic theoretically calls these two routines at entry/exit time. Some types of interrupts are postponed while a process is marked as being in **system code**.

Display column: ThdPID

ADM field: THDPID

The **ThdPID** column shows the Thread Protection ID associated with the thread.

Display column: Ttyp

ADM field: THDTYPE

The **Ttyp** column shows the type of executable entity for each process. The values are:

proc	ordinary process
main	process that has initiated threads
thd	process that is a thread

NOTE **main** does not seem to be seen as often as expected.

Display column: #Thds

ADM field: THREADS

The **#Thds** column shows how many threads a process has (not counting the original thread). For most processes, this value is 0 and is suppressed from the display.

Display column: Tin

ADM field: TIN

The **Tin** column shows the tin (Thread Identification Number) for each process. This is usually 1. Note that this is not the number of threads a process has!

Display column: TotFl

ADM field: TOTFAULTS

The **TotFl** column shows the total number of code and data page faults for each process.

Display column: Tot%

ADM field: TOTPERCENT

The **Tot%** column shows the approximate percentage of available CPU that a process has used since SHOT started.

Display column: TrapISM

ADM field: TRAPISM

The **TrapISM** column shows the address of the most recent Interrupt Stack Marker used as the result of a code trap in the process. Note that this area of the stack may have been overwritten since the interrupt occurred.

Display column: TrapPC

ADM field: TRAPPC

The **TrapPC** column shows the value of the Program Counter (PC) at the time of the most recent code trap for the process.

Display column: TrapStat

ADM field: TRAPSTAT

The **TrapStat** column shows the value of the HPE_STATUS that explains the most recent code trap for the process.

Display column: User.Account

ADM field: USERACCT

The **User.Account** column shows the logon user ID and account ID of a process.

Display column: Wait#

ADM field: WAITNUM

The **Wait#** column shows the **blocked_reason** field of a process PIB. This is the fine-grained reason why the process is not currently executing (if the value is 100, then the process is not blocked). This is the raw numeric value that normally gets formatted as the **WAIT** column.

Display column: W.Port

ADM field: WAITPORT

The **W.Port** column reports the port number a process is waiting on (or, if it is waiting on multiple ports, the address of the array of port numbers). The field should be 0 if a process is not waiting on a port.

Display column: Wait Semaphore

ADM field: WAITSEM

The **Wait Semaphore** column reports the address of the semaphore a process is waiting on, if the process in a Control Block Wait (wait # 18).

Display column: sPIN

ADM field: WAITSEMPIN

The **sPIN** column reports the PIN of the process owning the semaphore a process is waiting on, if the process in a Control Block Wait (wait # 18) (and if SHOT can determine the owning PIN).

Display column: WakeD

ADM field: WAKEDAD

The **WakeD** column reports **Yes** if the process will awaken its parent process (**Dad**) when it terminates.

Display column: WorkGroup

ADM field: WORKGROUP

The **WorkGroup** column shows the Dispatcher WorkGroup that the process is in. (WorkGroup Manager is an optional product.)

Display column: XRThigh

ADM field: XRTHIGH

The **XRThigh** column shows the **high water mark** of the XRT table for the process.

After a **Delta**, **Jobs**, **Sessions**, **%**, or **All** display, SHOT reports the overall CPU utilization and the number of page faults. Both numbers are approximate, and represent the usage since the last display. An example is:

```
CPU status: busy 4%, idle 96% (147 CPU out of 3252)
# Page faults: 2 Code, 2 Data
```

Both numbers are approximate, because if a process terminates SHOT cannot see its final resource usage.

When CPU times are reported, they're in one of the following formats:

#####	number of milliseconds of CPU time
s #####	number of seconds of CPU time
m #####	number of minutes of CPU time
h #####	number of hours of CPU time
D #####	number of Days of CPU time
W #####	number of Weeks of CPU time

At the end of every major display, SHOT checks to see if the Measurement Interface (MI) is **on**. If it is, a warning message is displayed: **(Warning: MI on)**

This warning is generated because simply turning on the MI (which is done by most performance measurement tools (other than SHOT)) can degrade system performance by up to 70%,even when the tool is not actively collecting or displaying data.

This is an easy assertion to test:

1. Time how long a command takes.

   Example:

   ```
   :fcopy from=catalog.pub.sys;to=$null (throw away the first timing)
   :setvar elapsed hpconnsecs
   :setvar cpu hpcpumsecs
   :fcopy from=catalog.pub.sys;to=$null
   :setvar cpu hpcpumsecs - cpu
   :setvar elapsed hpconnsecs - elapsed
   :echo fcopy took !cpu CPU millisecs, !elapsed Elapsed seconds
   ```

2. Enable the MI (run GLANCE, wait for a screen display, hit **Break**)
3. Time how long the command takes now:

   ```
   :fcopy from=catalog.pub.sys;to=$null (throw away the first timing)
   :setvar elapsed hpconnsecs
   :setvar cpu hpcpumsecs
   :fcopy from=catalog.pub.sys;to=$null
   :setvar cpu hpcpumsecs - cpu
   :setvar elapsed hpconnsecs - elapsed
   :echo fcopy took !cpu CPU millisecs, !elapsed Elapsed seconds
   ```

4 Compare the timings from the second run of each command. The first runs are thrown away out of fairness: different amounts of the file and code may have been in memory prior to the start of the command.

Queues, Quantum & Performance

In this section, we will discuss the special topics of Queues, Quantum and Performance.

Queues

MPE/iX is a priority-based operating system. every process in the system is assigned a priority between 1 and 255 (1 is the highest priority). Processes are scheduled into and executed in one of five queues (AS, BS, CS, DS, ES), where each queue covers a range of priorities.

The AS and BS queues are fixed priority, linear queues. Typically these queues are used for system processes. The CS, DS, and ES are referred to as circular queues (or subqueues). Processes that execute in these queues begin with the highest priority process and decay towards the lowest priority process as CPU resources are consumed. When a process reaches the limit of a queue (i.e., the lowest priority), or when it completes a transaction such as a disk I/O, a terminal I/O, or is preempted, it will circulate back into the queue with a new priority status assignment along with the other processes.

Default priority ranges for the CS, DS, and ES queues are:

Table 16.6 Priority ranges for queues

Queue	Range
CS	152 - 200
DS	202 - 238
ES	240 - 253

Quantum

A quantum is the measure of time that determines how much CPU time a process can have at a given priority. For the CS queue, the quantum is calculated by the operating system (based on demand) and is referred to as the SAQ (System Average Quantum). The quantum for the DS and the ES queues is fixed, although the System Manager can alter this value with the TUNE command. For any of these queues the quantum is used to control the rate of process priority decay.

Additionally, timeslicing is used to limit CPU-bound processes. The hardware of the HP3000 generates an interrupt that is used by the dispatcher to determine if a process has exceeded its current quantum.

Performance Optimization

Generally speaking, optimal performance can only be achieved when a process is executing in native mode. The next best performance can be obtained by using the Object Code Translator. This program is used to translate your compatibility mode programs into native mode programs. The least desirable performance scenario occurs when running compatibility mode programs in emulation mode. When a program runs partly in native mode and partly in compatibility mode, it is called a **mixed-mode** program. Mixed-mode programs tend to operate at higher performance levels than those in strictly compatibility mode, but penalties are incurred for switching between the modes.

Command Summary

The following list provides a simple description of SHOT commands that you can use to quickly locate the command that suits the task at hand. Detailed information on each command is provided in the next section.

NOTE Portions of the command codes are printed in uppercase to denote the part of the command that SHOT requires in order to distinguish one command from another. However, the commands themselves are not case-sensitive.

Table 16.7 *SHOT Commands*

Command Code	Description
ABORTCONNS	Aborts all network connections for the specified process
ABORTPROCIO	Aborts all outstanding I/Os for the specified process
ADM	Formats process output display
All	Shows all processes
ANCestors	Displays the ancestors of the specified process
Break pin#	Suspends a process
CPU pin#	Prints the CPU usage for the specified pin
DAD	Similar to ANCESTORS
DEBUG	Invokes the HP DEBUG program
Delta	Shows processes using CPU between displays

Command Code	Description
DETERMINEPROCNAMES	Tells SHOT to regenerate SHOTPIDS.DATA
ERASEPROCNAMES	Tells SHOT to erase SHOTPIDS.DATA
EXCLUDEPins	Tells SHOT to exclude from the process display the specified process
EXCLUDEwaits	Tells SHOT to exclude from the process display any process that is waiting for the specified event(s)
Exit	Terminate the program
FAMily	Displays the ancestors of the specified process
HELP	Invokes SHOT help
HIGHLIGHT	Defines the display lines SHOT highlights
Job	Shows processes in a job-oriented display
JOBINFO	Displays information about the next job or session
JS pin#	Shows information about the specified pin
KILL pin#	Terminates the process
LISTJOBQ	Lists the defined job queues
PARENT	Similar to ANCESTORS
PAUSE #seconds	Tells SHOT to pause for specified number of seconds
PIN pin#	Displays information about the specified pin
PRIority pin#	Changes priority of a process
PROGrams	Lists all program files currently in use
Resume	Activates suspended process
[RE]SET	Includes options for selecting report items
Sessions	Lists all jobs and sessions
SHOW	Lists the values for certain process/job/port states
SIRs	Reports on locked SIRS
STARTIDLE	Starts a child process to consume otherwise idle CPU time

Command Code	Description
STOPIDLE	Terminates the SHOT idle process(es)
SUMREUSE	Asks SHOT to sum the PID reuse counter for all processes and to report the result
SUMWAIT	Displays the summary of wait reasons for the last SHOT display
Suspend	Tells SHOT to go to sleep until its parent or child wakens it
TABles	Displays information on tables
TP pin#	Displays accumulated CPU time for a process tree
TRace pin#	Prints stack trace for the specified pin
Tree [pin#]	Shows a nested Tree-oriented display of processes
USEq	Tells SHOT to read commands from the specified file
USErs	Shorthand for SET PTYPE=USER SON MAIN
?	Synonym for HELP
% #	Displays processes using this % of CPU per display
/	Terminate the program

Command Definitions

This section discusses each of the SHOT commands in detail. In several cases, we have provided syntax examples.

ABORTCONNS

The ABORTCONNS command has the following syntax:

```
ABORTCONNS pin
```

This command will attempt to abort all network connections for the specified process (usually a VTSERVER process).

NOTE This should be used with great care, and requires SM capability.

ABORTPROCIO

The ABORTPROCIO command has the following syntax:

```
ABORTPROCIO pin
```

This command will attempt to abort all outstanding I/Os for the specified process.

NOTE This should be used with great care, and requires SM capability.

ADM

The ADM command has one of the following syntaxes:

```
ADM

ADM ?

ADM EDIT

ADM [ DEFAULT | APPEND | + ] <fieldlist>
```

Where:

`<fieldlist>` is: `field [ , field ...] [ ) ] [ & field , field ... ]`

The ADM (Automatic Display Mode) command tells SHOT what information should be displayed for the Delta, All, TREE, and % displays. It also specifies the order of the information. Each item of information is called a **field**.

If no options follow the ADM command, the current field list is displayed:

```
SHOT [Delta]: ADM

Active ADM field list:
   ADM (STATE, PIN, CPU, NAME, PRI, QUEUE, DELTA, PERCENT, EXECMODE, WAIT)
```

Figure 16.2 *ADM Command - Example 1*

If ADM is followed by a simple list of fields, then the old list is cleared and the new list is used. The field name ALL means *all known ADM fields*, and produces extremely verbose and hard to read output. The field name ALLSINGLE means *all known ADM fields, with one field/value per line*. This is also extremely verbose.

If ADM is followed by the word DEFAULT, then the list is set to the default set of ADM fields, plus whatever fields follow on the rest of the command line. The default list of fields is usually:

```
ADM (STATE, PIN, CPU, NAME, PRI, QUEUE, DELTA, PERCENT, EXECMODE,
WAIT)
```

If ADM is followed by the word APPEND (or a plus sign), then the new list is appended near the end of the old list. If the old list had a WAIT field, then it is temporarily removed from the old list, the new fields are added, and then the WAIT field is placed at the end of the combined field list.

If ADM is followed by the word EDIT, then the current ADM list is displayed for editing. To exit the editing without using the new list, press control-Y or erase the entire list (control-X followed by control-E). If an error is found in the list, the edited list will be displayed for re-editing.

If a question mark (?) follows the ADM command, then all known fields will be listed, along with their length, and formatting characteristics.

```
SHOT [Delta]: ADM ?

Legal ADM fields:

                                                    Dup
   Field        Length  DefLen  Format              ok?  DisplayTitle
   -----------  ------  ------  ------------------  ---  ------------
   BLANK             1          Left                Yes
   BLANK4            4          Left                Yes
   BLANK8            8          Left                Yes
   BLOCKEDR          4          Right                    BlkR
   CAPABILITY        9          Right                    Caps
   CCODE             3          Right                    cc
   CFAULTS           5          Right  (blank 0s)        CFlts
   CLS              -1
   CMSTACK          11          Right                    StXDS DbXDS    S     Z
   CMSZ             13          Right                       S     Z
   CMTIME            6          Right                    CMtime
   COMMA             1          Left                Yes  ,
   CPU               6          Right                    CPU
   CR                1          <cr>                Yes  <cr>
   CRITICAL          1          Right  (blank 0s)        C
   DASH8             8          Left                Yes  --------
   DEBUG             3          Center                   Dbg
   DELCFAULTS        5          Right  (blank 0s)        DelCFl
   DELDFAULTS        5          Right  (blank 0s)        DelDFl
   DELFAULTS         5          Right  (blank 0s)        DelFl
   DELTA             5          Right  (blank 0s)        Delta CPU
   DFAULTS           5          Right  (blank 0s)        DFlts
   DIRTYPAGES        8          Right  (blank 0s)        DirtyPgs
   DISPATCHER       22          Left                     Dispatcher
   DPLIMIT           9          Right  (blank 0s)        DPlimit
   DUMPARMED         5          Left                     Dump
   ENTRY            14          Right                    Entry
   EUIDCAPS          9          Right                    EUID Caps
   EXECMODE          2          Center                   Ic
   FORKED            1          Left                     F
   FORKSP            9          Right  (blank 0s)        $Fork SP
   GDPLIMIT          9          Right                    GlbDP lim
   HEAPBASE          9          Right  (blank 0s)        HeapBase
   HEAPLIMIT         9          Right  (blank 0s)        HeapLimit
```

```
HEAPPAGES         8          Right  (blank 0s)       #HeapPgs
HEAPSIZE          8          Right  (blank 0s)       HeapSize
INFO             26          Left                    INFO=
INFOPTR          14          Right  (blank 0s)       INFO ptr
INTPRI            6          Left                    IntPri
JSNUM             6          Left               Yes  Job#
KILLER            6          Right  (blank 0s)       Killer
LSTX              5          Right  (blank 0s)       LSTX
MEMPAGES          8          Right  (blank 0s)       MemPages
NAME             26          Left                    Process Name
NEWPAGE          -1
PARENTPIN         6          Right  (blank 0s)       Parent
PARM              6          Right  (blank 0s)       Parm=
PC               13          Right                   ProgrmCounter
PENDING          20          Left                    Pending Events
PERCENT           3          Right  (blank 0s)       %
PERCENTSB         3          Right  (blank 0s)       %SB
PFT               3          Right  (blank 0s)       Pft
PID               7          Left               Yes  PIN/Reuse
PIN               3          Right              Yes  PIN
PLABTAB          13          Right  (blank 0s)       Plabel Table
POSIX             1          Center                  P
POSIXPID         10          Right  (blank 0s)       PosixPid
PRI               3          Right                   Priority
PROTID            6          Right  (blank 0s)       ProtID
PTYPE             4          Left                    Ptype
PXPROGNAME       32          Left                    PxProgname
QUEUE             1          Center                  Queue
SECPAGES          8          Right  (blank 0s)       SecPages
STARTTIME        17          Right  (blank 0s)       StartTime
STATE             1          Left                    ?
STKSID            6          Right  (blank 0s)       StkSID
SUSPEND           5          Right  (blank 0s)       #susp
SWDEPTH           1          Right  (blank 0s)       SW Depth
SYSCODE           1          Right  (blank 0s)       S
THDPID            6          Right  (blank 0s)       ThdPID
THDTYPE           4          Right  (blank 0s)       Ttyp
THREADS           5          Right  (blank 0s)       #Thds
TIN               3          Right  (blank 0s)       Tin
TOTFAULTS         5          Right  (blank 0s)       TotFl
```

```
TOTPERCENT        4          Right  (blank 0s)       Tot%
TRAPISM           9          Right  (blank 0s)       TrapISM
TRAPPC           13          Right  (blank 0s)       TrapPC
TRAPSTAT          9          Right  (blank 0s)       TrapStat
USERACCT         17          Left                    User.Account
WAIT             64          Left                    Wait State
WAITNUM           5          Right                   Wait#
WAITPORT          9          Right  (blank 0s)       Wait Port
WAITSEM          14          Left                    Wait Semaphore
WAITSEMPIN        5          Right  (blank 0s)       sPIN
WAKEDAD           5          Left                    WakeD
WORKGROUP        10          Left                    WorkGroup
XRTHIGH           7          Right  (blank 0s)       XRThigh
```

Figure 16.3 *ADM Command - Example 2*

Help on individual ADM fields can be obtained by typing `HELP ADM-<fieldname>` (i.e., HELP ADM-CR)

The title column shows the title string that will be displayed in the SHOT output (room permitting).

A format note of **(blank 0s)** means that zero values will be displayed as a blank field. A note of **Yes** in the **Dup ok?** column means that a field list may contain multiple instances of the field.

The DefLen column shows the default length of each field, but only when the default length does not match the current length. When adding a field to the ADM list, a different width (other than the default) can be specified by appending a colon (:) and the desired width.

For example: `CRITICAL:2`

NOTE Some of the SET commands do implicit ADM commands. For example, SET SWDEPTH adds the SWDEPTH field to the ADM field list.

Normally, the response to a Delta or ALL command is built by stringing the requested ADM fields together, one after another, with a blank separating each field.

Some of the ADM fields are for output formatting use, including:

BLANK	Displays a single blank
BLANK4	Displays four blanks
BLANK8	Displays eight blanks
COMMA	Displays a comma
CR	Displays a carriage return (i.e., splits the output into multiple lines at that point)
DASH8	Displays eight dashes ("--------")
CLS	Sends a home-up/clear-screen command (<esc>h <esc>J>). This would typically be used at the start of an ADM sequence. It is sent only with the ADM header, not once per process.
NEWPAGE	Sends a home-down, vertical tab, next-page command, which effectively places the cursor at the top of a fresh page of terminal memory, without losing data already in terminal memory (i.e., you can "page" back and see the prior displays).

The rest of the ADM fields are documented in “The SHOT Process Display” on page 203 section above.

All

This command has the following syntax:

```
All [USER | ALL]
```

The ALL command displays all processes in system. If **ALL USER** is specified, only **user** processes are displayed (until an ALL ALL is done). If **ALL ALL** is specified, all processes are displayed.

ANCestors

This command has the following syntax:

```
ANCestors pin# [[no]LOCK]
```

Displays the ancestors of the specified process, back to process 1 (1 is the final ancestor of every process).

LOCK guarantees that no process will terminate or start during the traversal of the process tree. NOLOCK tells SHOT not to do any locking. LOCK is the default.

Break pin#

Suspends the specified process.

NOTE Because breaking and resuming processes can cause system problems, only SM or PM users may issue this command.

CPU pin#

Reports the CPU usage for the specified process.

DAD pin

PARENT pin

These commands have the following syntax:

```
PARENT pin [ <All | [no]Verbose> ]
DAD    pin [ <All | [no]Verbose> ]
```

The PARENT (or DAD) command is similar to the ANCESTORS command, in that it can be used to display the parent, grandparent, etc., of a process. Unlike ANCESTORS, it has the option of displaying the program name (by specifying VERBOSE) as well. Additionally, the default is to just show the parent process, not the rest of the ancestors (ALL requests all of them).

NOTE Unlike the ANCESTORS command, DAD/PARENT never locks the process tree while traversing it.

DEBUG

Invokes DEBUG/iX, the free instruction-level debugger that comes with MPE/iX. To exit Debug and resume SHOT, enter: c

NOTE The first time DEBUG is invoked for a process, it takes a noticeable amount of time to startup. SHOT is aware of this and warns you the first time you use the DEBUG command.

Delta

Display only processes that have used CPU since the last display.

DETERMINEPROCNAMES

DETERMINEPROCNAMES tells SHOT to erase the SHOTPIDS.DATA file and determine anew the names for *procreated* processes (i.e.: pin ?). An implicit DETERMINEPROCNAMES is done by the first SHOT that runs after a system startup.

ERASEPROCNAMES

ERASEPROCNAMES tells SHOT to erase the SHOTPIDS.DATA file. A subsequent run of SHOT (or a subsequent use of the DETERMINEPROCNAMES command) will rebuild the data in the file.

NOTE This command is not generally useful to users.

EXCLUDEPins

This command has the following syntax:

```
EXCLUDEPins <LIST | ? | NONE | ALL | # [/#] | -#>  [...]
```

EXCLUDEPINS tells SHOT to exclude from the process display (for ALL, DELTA, JOBS, and SESSIONS) the specified processes.

EXCLUDEPINS	Will list the current filter.
EXCLUDEPINS ? (or EXCLUDEPINS LIST)	Same as EXCLUDEPINS
EXCLUDEPINS NONE	Turns off the exclude-pin filter entirely.
EXCLUDEPINS ALL	Says "exclude every process".
EXCLUDEPINS #	Says "exclude specified PIN"
EXCLUDEPINS #/#	Says "exclude specified PIN range"
EXCLUDEPINS -#	Says "do not exclude specified PIN"

See also: SET TRACE

EXCLUDEwaits

This command has the following syntax:

```
EXCLUDEwaits <LIST | ? | NONE | ALL | # | -# | field | -field> [...]
```

EXCLUDEWAITS tells SHOT to exclude from the process display (for ALL, DELTA, JOBS, and SESSIONS) any process that is waiting for the specified event(s).

EXCLUDEWAITS	will list the current filter.
EXCLUDEWAITS ? (or EXCLUDEWAITS LIST)	same as EXCLUDEWAITS
EXCLUDEWAITS NONE	turns off the exclude-wait filter entirely.
EXCLUDEWAITS ALL	says "exclude every process".
EXCLUDEWAITS field (or #)	says "exclude specified wait type".
EXCLUDEWAITS -field (or -#)	says "do not exclude specified wait type"

To see a list of the possible reasons for exclusion, enter:

```
EXCLUDEWAITS ??
```
(two question marks)

which will list all reasons, flagging the currently excluded ones with an asterisk (*).

See also: SET TRACE

Exit or /

Terminates the program.

FAMily pin#

This command has the following syntax:

```
FAMily pin# [[no]LOCK]
```

Displays the ancestors of the specified process, back to process 1 (1 is the final ancestor of every process).

Displays the descendants of the specified process, as far down the process tree as they exist. The children are displayed one per line, indented two spaces. The grandchildren are displayed one per line, indented four spaces. And so on!

LOCK guarantees that no process will terminate or start during the traversal of the process tree. NOLOCK tells SHOT not to do any locking. LOCK is the default.

HELP

This command has the following syntax:

```
HELP [commandname]
```

HELP commandname gets help information on all commands that match the specified portion of the command name. For example, HELP T would get information on the TP command, TRACE command, and TREE command.

HIGHLIGHT

This command has the following syntax:

```
HIGHLIGHT [CLEAR] [OFF] [ON] [PROGRAM file.group.acct]
                                [USER [job,]user.acct]
```

The HIGHLIGHT command defines what lines of a SHOT display should be highlighted. A line will be highlighted if it satisfies either the PROGRAM pattern or the USER pattern.

HIGHLIGHT OFF	temporarily disables highlighting, but without forgetting the PROGRAM and USER patterns.
HIGHTLIGHT ON	restores the highlighting check.
HIGHLIGHT CLEAR	clears all patterns and turns off highlighting.

The **file**, **group**, **acct**, **job**, and **user** patterns may have standard MPE wildcards.

NOTE *Procreated* processes are not subject to highlighting, as they do not have program files associated with them.

Example 1: to highlight users of QUERY.PUB.SYS, do:

```
HIGHLIGHT PROGRAM QUERY.PUB.SYS
```

Example 2: to highlight all users in the SALES account, do:

```
HIGHLIGHT USERS @.SALES
```

Job

This command has the following syntax:

```
Job [Only]
```

Displays a list of all jobs and sessions. For each job/session, a **process tree** is displayed, showing every process belonging to the job/session. The **Only** keyword restricts the display to only jobs (and not sessions).

JOBINFO

This command has the following syntax:

```
JOBINFO [NEXTJOB #] [NEXTSESSION #]
```

The JOBINFO command with no options tells SHOT to display some job/session global information.

NEXTJOB	Allows SM users to reset the counter that MPE uses to assign the next job's number.
NEXTSESSION	Allows SM users to reset the counter that MPE uses to assign the next session's number.

JS pin#

The JS command reports the Job/Session associated with a specified pin.

KILL pin#

Tells SHOT to try to kill the specified process. SHOT will ask for confirmation before it proceeds. Requires SM capability.

LISTJOBQ

This command has the following syntax:

```
LISTJOBQ [prefix]
```

Lists the defined job queues. If [prefix] is omitted, all job queue are listed. Otherwise, only those job queues that match the prefix are listed. For example, LISTJOBQ PROD will list PROD, PRODBIG, and PRODSLOW (assuming they exist, of course).

PAUSE #seconds

Tells SHOT to pause (by calling the PAUSE intrinsic) for the specified number of seconds. You should be able to terminate the pause early by pressing control-Y.

%

Tells SHOT to only display processes that have used at least the specified percentage of the CPU since the last display.

Example: `%5`

PIN pin#

pin#

Display information about the specified process.

Example 1: `45`

Example 2: `PIN 45`

PRIority pin#

This command has the following syntax:

```
PRIority pin# <CS | DS | ES | BS | AS>
```

Sets the specified pin to the desired priority.

NOTE Only a user with SM or PM can set a process to BS or AS priority, due to the possibility that a process running at that priority could take over the system.

Examples:

`PRI 0 CS` sets SHOT priority to CS

`PRI 45 ES` sets pin 45 to priority ES

PROGrams

Displays a list of programs that are currently running, sorted alphabetically. For each program, the number of processes running it is shown, along with the name of the program, the type of program (NM, CM, OCT), and (for OCT and CM) the capabilities of the program (i.e.: PM, ph). (A capability of **PrivSeg** means that at least one code segment in an CM or OCT program is privileged.)

Example:

```
#Procs   Program Name                Capabilities
------   --------------------------- ------------
     8 : CI.PUB.SYS                 NM:
     1 : DIAGMON.DIAG.SYS           NM:
     1 : LOAD.PUB.SYS               OCT: PrivSeg
     1 : NMFILE.PUB.SYS             CM:  PM ph PrivSeg
```

Resume pin#

Resumes a process suspended with the Break command.

NOTE Because breaking and resuming processes can cause system problems, only SM or PM users may issue this command.

Sessions

This command has the following syntax:

```
Sessions [Only]
```

Displays a list of all jobs and sessions. For each job/session, a **process tree** is displayed, showing every process belonging to the job/session. The **Only** keyword restricts the display to only sessions (and not jobs).

[RE]SET

These commands have the following syntax:

```
[RE]SET [<setoption> [ , <setoption> ... ] ]
```

The SET and RESET commands may be followed by one or more <setoption>s. Each option is explained below. An option following a **SET** keyword is turned on, an option following a RESET keyword is turned off. If no option follows SET or RESET, then all current option settings are displayed.

```
<setoption> ::=
```

[RE]SET ALL

Obsolete synonym for **[RE]SET MOST**.

SET DELAY #seconds

RESET DELAY

The DELAY #seconds option tells SHOT to automatically display the system activity every # seconds. This is accomplished by doing a timed-read of # seconds as a prompt. If input is received before the timeout, it will be acted on. Otherwise, the timeout causes the display to be updated with either ALL processes, Delta processes, or just those which have used more than the threshold percentage of CPU (see the % command). When a timed-read times out, the display that results is governed by the last Delta, ALL, or % command seen. DELAY 0 disables the timed out reads.

Default: 0

Example: `SET DELAY 10`

[RE]SET DELTA

Adds (SET) or deletes (RESET) the DELTA field from the ADM display.

[RE]SET EXECmode

When EXECMODE is on, SHOT reports the initial and current **mode** of execution for each process in a two character column. The first character reports the initial mode of a process (C for Compatibility Mode, N for Native Mode), and the second character reports the current mode (c for Compatibility Mode, n for Native Mode). The initial mode for CM programs (filecode PROG) is C (even if they were Object Code Translated). The initial mode for NM programs (filecode NMPRG) is N.

Default: ON

[RE]SET FAULTs

When FAULTS is on, SHOT reports the number of data faults and code faults a process has had since the last display (DELFAULTS). See also: CFAULTS, DFAULTS, DELFAULTS, and TOTFAULTS.

[RE]SET HEXPINs

When HEXPINS is on, SHOT will report pin numbers in hexadecimal instead of decimal. This is useful when using SHOT and DEBUG in parallel.

Default: OFF

[RE]SET JOBSTEP

When JOBSTEP is on, a JOB display will show the current job "step", which is (essentially) the last CI-command read.

NOTE Seeing the actual job step may require SM capability.

Default: ON

[RE]SET JSnum

When JSNUM is on, the process display will include job and session numbers.

Default: OFF

SET LINES #

SET COLUMNS #

LINES and COLUMNS values tell SHOT how big your display is. The default is 24 by 80, which is initially modified by the values of the CI variables **COLUMNS** and **LINES** (if they exist).

[RE]SET MOST

Setting MOST is equivalent to doing: SET hexpins, jobstep, jsnum, showptype, swdepth, unknown.

Resetting MOST is equivalent to doing:

```
RESET hexpins, jobstep, jsnum, showptype, swdepth, unknown
```

[RE]SET NUMPINs

SET NUMPINS (the default) tells SHOT that a terse summary should report the number of processes.

[RE]SET ONEchar

For interactive runs, the ONECHAR option (on by default) tells SHOT to do single-character reads at the prompt. The NOONECHAR option makes the prompt do a bigger read, so an entire command line can be entered at once.

[RE]SET PC

The PC option is shorthand for `ADM + (or -) PC`

[RE]SET PENDING

When the PENDING option is on, SHOT displays a column showing events that are **pending** for a process. This includes interrupts, activations, and other messages that would be sent to a process' standard signal, message, and interrupt ports.

[RE]SET PIDs

When the PIDS option is on, SHOT displays Process IDs instead of pins. A Process ID (pid) is displayed as a pin, a slash (/), and a **re-use** counter for the pin. A pid is unique during the lifetime of one bootup of MPE/iX.

[RE]SET POSIX

The POSIX option is shorthand for `ADM + (or -) POSIX POSIXPID`

[RE]SET PRIority

The PRIORITY option is shorthand for `ADM + (or -) PRI`

[RE]SET PTYPE

This command has the following syntax:

```
[RE]SET PTYPE [=] [ + | - ] <ALL | DETach | NONE | SYStem | TASK |
                                UCOp | USErs>
```

The PTYPE option tells SHOT what *kind* of processes should be shown in the process display. Every process has a **process type**, which is one of: detach, system, task, ucop, main, or user.

Example: `SET PTYPE = DETACH` causes SHOT to show only **detached** processes.

[RE]SET QUEUE

The QUEUE option is shorthand for `ADM + (or -) QUEUE`

RESET

All options found after the keyword RESET are turned OFF. A RESET keyword may be used in the same line as a SET keyword.

Example: `SET ONE RESET PIDS`

SET

All options found after the keyword SET are turned ON. A SET keyword may be used in the same line as a RESET keyword.

Example: `RESET PIDS SET ONE`

[RE]SET SHOWBIRTHs

[RE]SET SHOWDEATHS

Normally, when SHOT notices that a process has terminated, or that an new process has been born, it will not volunteer anything. SET SHOWBIRTHS tells SHOT to mention when it sees a new process. SET SHOWDEATHS tells SHOT to mention when it notices that a process has terminated.

[RE]SET SHOWPRIORwait

SET SHOWPRIORWAIT tells SHOT that in some cases it should report the last port number that a process waited on, even if we think it is currently not waiting.

NOTE This is experimental.

[RE]SET SHOWPTYPE

When SHOWPTYPE is on, the process display will show the **type** of each process. This option is not the same as the **PTYPE** option which can be used to filter out processes based on their type.

SET SUMMARY

This command has the following syntax:

```
SET SUMMARY = <NORMAL | TERSE | VERBOSE>
```

The SUMMARY option controls the amount of information displayed at the end of a DELTA or ALL command.

[RE]SET SWDEPTH

When SWDEPTH is on, the process display will show the Switch Depth of every process. This counter reflects the total number of switch-to-CM and switch-to-NM markers that are currently in the process's stack. The SWDEPTH column is labelled **S**. A blank denotes a switch depth of 0.

NOTE SWDEPTH does not show the cumulative number of switches done since the process started. Some system routines (i.e.: CM DBGET) switch modes in a manner that does not increment the SWDEPTH counters. This can result in SHOT showing a process as being **Cn** with an even value in the SWDEPTH column.

Default: OFF

[RE]SET THREADS

The THREADS option is shorthand for `ADM + (or -) THREADS THDTYPE TIN`

[RE]SET TIMES

SET TIMES tells SHOT to report the CPU and elapsed time that some SHOT commands use. Note that the DELTA/ALL command output is generated outside the timing statistics area.

[RE]SET TOTPERCENT

When TOTPERCENT is on, the process display will show the total cpu usage percentage since each process was first seen by SHOT.

[RE]SET TRACE

Experimental option. When SET, SHOW will produce a stack trace for any process selected in a DELTA or ALL report.

[RE]SET UNKNOWN

When UNKNOWN is on, SHOT will report processes that are in the **unknown** state if they use CPU time. Normally, processes should not be in this state AND be using CPU time. MPE/iX has a small timing window where this appears to sometimes happen.

Default: OFF

[RE]SET VJOBINFO

SET VJOBINFO tells SHOT to produce a verbose output for each job found in the **JOB** or **SESSIONS** command.

NOTE This is experimental.

[RE]SET WAITNUM

SET WAITNUM tells SHOT that the values in the ADM WAIT field should include the wait state number, as well as the wait state name.

[RE]SET WAITTRACE

SET WAITTRACE tells SHOT that if a DELTA or ALL command finds a process that is in a control block wait (wait # 18), *and* SHOT can determine the PIN of the process that owns the control block, then SHOT should do a stack trace of that process.

SET WIDTH #

SET 80

SET 132

SET WIDTH # sets the logical terminal width to the specified number of columns. No other action is taken, as SHOT assumes you have manually changed your terminal's width externally.

Note that SET WIDTH 132 is *not* identical to SET 132, because SET 132 does an implied SET MOST, and also emits the HP escape sequence to put your terminal into 132 column mode.

SET 132 tells SHOT to try to put an HP700/9x terminal into 132 column mode. SET 132 also implies SET MOST.

SET 80 tells SHOT to try to put an HP700/9x terminal into 80 column mode.

SHOW

This command has the following syntax:

```
SHOW <WAIT | JOBS | PORTS | PSM | PSP>
```

The SHOW command lists the values for certain process/job/port states. Generally, only SHOW WAIT is of much use.

SHOW WAIT shows all the wait states (which may be specified in `SET WAIT = xxxxx` command).

NOTE Wait states 0, 1, 2, and 3 are generally not seen in the SHOT **WAIT** column, because they are actually stored as more refined values (i.e., perhaps 18 instead of 2).

SHOW JOBS	Shows all the job states
SHOW PORTS	Shows the known port numbers
SHOW PSM	Shows some more reasons processes wait on a port
SHOW PSP	Shows some reasons processes wait on a port

SIRs

The SIRS command reports what SIRs (System Internal Resources) are locked (if any), by whom, and the list of waiting processes.

STARTIDLE

STOPIDLE

STARTIDLE starts a child process to consume otherwise idle CPU time. The new process runs at extremely low priority, and does *while true do;* (i.e., infinite loop).

Having a low priority process *eat* CPU time allows you to get a different view of CPU utilization. On a busy system, the birth/death of processes may distort SHOT's view of how busy the system is. After a STARTIDLE command, the amount of CPU used by the sum of the **SHOT idle** processes represents the amount of CPU that would normally have been available, unused. For example, on a single CPU system, if the SHOT idle process is using 75% of the CPU, then if it were not running, the CPU would be 75% idle (and 25% busy).

STARTIDLE will start one **SHOT idle** process per CPU.

STOPIDLE terminates the SHOT idle process(es).

NOTE These commands are rarely used, because SHOT has other methods of determining CPU utilization. They are here in case you want to use them.

All performance tools should report the CPU as 100% (or completely) busy after a STARTIDLE, and the *speedometer* on the front of the computer (or on the status line of ldev 20) should show FAFF (100% busy).

SUMREUSE

The SUMREUSE command asks SHOT to sum the PID reuse counter for all processes and to report the result. The sum should be the number of processes that have run on your system since bootup, counting the currently living processes.

Example:

```
SUMREUSE

Counting...

Sum: 7,057

Checked 122 PINs
```

SUMWAIT

This command has the following syntax:

```
SUMWAIT [CLEAR] [LIST] [ALL]
```

Displays the summary of wait reasons for the last SHOT display. The summary is cumulative. Every time SHOT does an ALL, DELTA, or "%" display, the summary is updated.

SUMWAIT LIST	Displays the current totals (that are not 0).
SUMWAIT ALL	Displays all totals, even the ones with 0.
SUMWAIT CLEAR	Sets all of the totals to 0.

Suspend

The SUSPEND command tells SHOT to go to sleep until its parent or child wakens it. It is implemented by calling: activate (0, 3);

TABles

The TABLES command reports on various Virtual Space Management tables. In particular, it reports the high-water Space ID (SID) that has been used, and the high-water Protection ID. It also reports how much short-mapped system space is available (SR6/SR7 space).

TP pin#

The TP command prints the accumulated CPU time for the entire process tree starting at the specified pin. (TP stands for Tree Process time).

TRace pin#

The TRACE command tells SHOT to print a stack trace for the specified pin.

Tree [pin#]

This command has the following syntax:

```
Tree pin# [depth#] [ALL]
```

Displays a **tree** of process information, with the specified process as the root. Using a pin# of 1 will result in the entire system's process tree. The depth# option, if specified, tells SHOT to limit the **depth** of the tree to the value specified.

The ALL keyword tells SHOT to temporarily ignore all process filters. (i.e., EXCLUDEPINS, EXCLUDEWAITS)

Example 1: `TREE 1` the tree of all processes

Example 2: `TREE 1,2` pin 1 and only its direct children (no grand-children)

Example 3: `TREE 0` process tree starting at SHOT

USEq

This command has the following syntax:

```
USEq filename
```

The USE and USEQ commands tell SHOT to read commands from the specified file. USE will echo the commands to $stdlist before executing them. USEQ does not echo the commands.

USE / USEQ files may not be nested.

USErs

The USER command is a shorthand version of the SET PTYPE command:

```
SET PTYPE = USER SON MAIN
```

USER will filter out all non **user** processes from most SHOT displays.

TOOLBOX STANDARDS

The ToolBox collections from Lund Performance Solutions have a uniform user interface. As a result, in addition to the commands specific to each Toolbox tool, most tools allow the commands described in “TOOLBOX STANDARDS” on page 311.

SHOT Examples

Here are some examples of the SHOT tool.

```
SHOT [Delta]: _
SHOT @ TUE, SEP 11, 2001, 10:09 AM

? PIN  CPU   Process Name               Pri Q Delta  %  Ic Wait State
- --- ------ -------------------------- --- - ----- --- -- --------------------
  107 s  606 SOSTASK.PGMS.SOSDEV        253 E    80  29 Nn TIMER
  323 s  614 SOSTASK.PGMS.SOSDEV        253 E    71  25 Nn TIMER
  396   1328 SHOT.PUB.LPSTOOLS          200 C    34  12 Nn (executing)
  510 s  645 SOSTASK.PGMS.SOSDEV        253 E    90  32 Nn TIMER

Summary:               Last 0.2 secs Since SHOT start  Since bootup
                       ------------- ----------------  ------------
CPU % busy                    100                  -             -
Code Page faults/sec            0.00             0.75             -
Data Page faults/sec            0.00             0.00             -
# Processes started         5,905                  0         9,791
Process Creates/min             0.00             0.00           1.0

(Warning: MI on)

Popular wait reasons          #PINs
---------------------------  -----
TIMER_WAIT                       3

SHOT [Delta]: _
```

Figure 16.4 *A typical SHOT DELTA Display*

The **DELTA** command shows which processes have recently used CPU resources and what percentage of CPU resource was consumed for each process.

The **ALL** command shows every process staring with pin#1 (PROGEN) in ascending order to the highest pin.

```
SHOT [Delta]: all
SHOT @ TUE, SEP 11, 2001, 10:14 AM

? PIN  CPU   Process Name               Pri Q Delta  %  Ic Wait State
- --- ------ -------------------------- --- - ----- --- -- --------------------
    1 s   21 PROGEN.PUB.SYS              13 A          Nn PROGEN_GLOBAL_PORT
    2 s  108 LOAD.PUB.SYS               142 B          On JUNK
    3 s  331 pm_cleanup                 100 B   212    nn JUNK
    4   5660 port_facility_process       13 A          nn PFP_PORT0
    5 s  129 port_facility_process      152 C    29     nn pfp_port1
    6   5252 port_facility_process      152 C     4     nn pfp_port2
    7 s  120 port_facility_process      152 C    83     nn pfp_port3
    8    216 port_purger_process        152 C     3     nn DELAYED_PURGER_PORT
    9     18 fsipc_post_timeout         152 C           nn BIPC_TIMER_PORT
   10   2525 xm_checkpoint_server        30 A     2     nn XM_CHECK_PT_PORT
   11 s   92 xm_static_checkpoint_serve 152 C    79     nn XM
   12     20 xm_static_checkpoint_serve 152 C           nn XM
   13     20 xm_static_checkpoint_serve 152 C           nn XM
   14     20 xm_static_checkpoint_serve 152 C           nn XM
   15     76 spsnet_process_init        100 B           nn SPUNET_PORT
   16     78 repeater_process_init      152 C           nn REPEATER_PORT
   17     49 io_mgr_process              13 A           nn MsgPort#-113
   18     47 io_mgr_process              13 A           nn MsgPort#-117
   19    706 io_mgr_process              13 A           nn MsgPort#-121
   20 s   16 avr_process_init           100 B           nn AVR_PORT
   21     60 mms_process_init           100 B           nn MMS_PORT
   24   5080 cm: NMMON                  149 B           cn MESSAGE
   25   1383 cm: MESSENGER              120 B           cn IO, TIMER, MESSAGE
   26   3517 NMFILE.PUB.SYS             149 B           Cn MESSAGE
   27    645 NMLOGMON.PUB.SYS           149 B           Nn MsgPort#-161
   28    370 NMLOGICS.PUB.SYS           148 B           Nn NMS_LOG_GLOBAL_PORT
   29    348 NMTRCMON.PUB.SYS           149 B           Cn MESSAGE
   30 s  369 NMCONSOL.PUB.SYS           149 B   225     Nn MsgPort#-171
   31   5547 io_mgr_process              13 A           nn MsgPort#-176
   32     56 io_mgr_process              13 A           nn MsgPort#-180
   33    438 io_mgr_process              13 A           nn MsgPort#-184
   34    157 io_mgr_process              13 A           nn MsgPort#-189
   35     52 io_mgr_process              13 A           nn MsgPort#-193
   36    647 io_mgr_process              13 A           nn MsgPort#-197
   37    477 JSMAIN.PUB.SYS             152 B           Nn MsgPort#-32776
   38    310 LOG.PUB.SYS                 50 B           Nn SYSLOG
```

```
 39     74 SYSMAIN.PUB.SYS              49 B           Nn SYSMAIN_PORT
 40   2846 NETCP.NET.SYS               149 B           Nn MEMORY_WAIT
 41    275 SPOOLMOM.PUB.SYS            100 B           Nn SPOOLER_MOM_PORT
 42    231 OUTSPOOL.PUB.SYS            152 C           Nn MsgPort#-32771
 44   6107 SESSION.PUB.SYS             100 B           Nn SESSIONMAIN_PORT
 45   2645 JOB.PUB.SYS                 100 B           Nn 2 ports@$1fc.4164496
              waiting on: JOBMAIN_PORT, JOB_QUEUE_PORT
 48     70 OUTSPOOL.PUB.SYS            100 B           Nn OTHER_IO_WAIT
 50   4380 TCPSIP.NET.SYS              149 B           Nn MsgPort#-339
 51    342 ICMPSERV.NET.SYS            152 C           Nn MEMORY_WAIT
 52    331 SOCKREG.NET.SYS             149 B           Cn TERMINAL_READ_WAIT
 53     52 PT2PNSTN.NET.SYS            149 B           Nn SigPort#-32780
 54   4266 DSDAD.NET.SYS               149 B           Nn TERMINAL_READ_WAIT
 55     95 VTSERVER.NET.SYS            149 B           Nn TERMINAL_READ_WAIT
 56     88 VTSERVER.NET.SYS            149 B           Nn TERMINAL_READ_WAIT
 57   2198 VTSERVER.NET.SYS            149 B           Nn TERMINAL_READ_WAIT
 58    172 VTSERVER.NET.SYS            152 C    17     Nn TERMINAL_READ_WAIT
 59    151 VTSERVER.NET.SYS            152 C           Nn TERMINAL_READ_WAIT
 60 s   10 VTSERVER.NET.SYS            152 C           Nn TERMINAL_READ_WAIT
 61   1002 VTSERVER.NET.SYS            149 B           Nn TERMINAL_READ_WAIT
 62   1409 VTSERVER.NET.SYS            149 B           Nn TERMINAL_READ_WAIT
 63    606 CI.PUB.SYS                  202 D           Nn SON, FATHER
 65   3855 INETD.NET.SYS               216 D           Pn MESSAGE_WAIT
 67    474 JSMAIN.PUB.SYS              152 B           Nn MsgPort#-32798
 69    973 CI.PUB.SYS                  202 D           Nn SON, FATHER
 70    515 JSMAIN.PUB.SYS              152 B           Nn MsgPort#-32799
 71 s 1654 SCOPEXL.SCOPE.SYS           100 B   649     Nn TIMER
 72 s  175 CI.PUB.SYS                  202 D           Nn TIMER
 75 s   62 SCOPEXL2.SCOPE.SYS          152 C           Nn TIMER
 76    520 CI.PUB.SYS                  152 C           Nn TERMINAL_READ_WAIT
 77 s   62 SOS.PUB.LPS                 100 B   504     Nn TIMER
107 s  630 SOSTASK.PGMS.SOSDEV         253 E s  24   6 Nn TIMER
120    537 JSMAIN.PUB.SYS              152 B           Nn MsgPort#-32800
133    649 STRCNTL.NET.SYS             149 B           Nn FILE_UNBLOCKED
155    857 CI.PUB.SYS                  202 D           Nn SON, FATHER
158    610 JSMAIN.PUB.SYS              152 B           Nn MsgPort#-32807
163     27 STRCNTL.NET.SYS             149 B           nn OTHER_WAIT
170     27 STRCNTL.NET.SYS             152 E           nn OTHER_WAIT
182    748 JSMAIN.PUB.SYS              152 B           Nn MsgPort#-32801
185 s   62 SOS.PUB.LPS                 100 B   503     Nn TIMER
189    585 JSMAIN.PUB.SYS              152 B           Nn MsgPort#-32812
```

```
  199    826 CI.PUB.SYS                202 D           Nn SON, FATHER
  201   1324 SH.HPBIN.SYS              152 C           Pn TERMINAL_READ_WAIT
  204    514 JSMAIN.PUB.SYS            152 B           Nn MsgPort#-32808
  210     21 JSMAIN.PUB.SYS            152 B           Nn MsgPort#-32814
  236 s   13 CI.PUB.SYS                152 C           Nn SON, FATHER
  245    645 SOSLOAD.PGMS.SOSDEV       240 E           Pn FILE_UNBLOCKED
  283   1103 CI.PUB.SYS                240 E           Nn SON, FATHER
  295    574 JSMAIN.PUB.SYS            152 B           Nn MsgPort#-32802
  323 s  639 SOSTASK.PGMS.SOSDEV       253 E s  24   6 Nn TIMER
  338    640 CI.PUB.SYS                152 C           Nn SON, FATHER
  380    686 JSMAIN.PUB.SYS            152 B           Nn MsgPort#-32806
  396   1514 SHOT.PUB.LPSTOOLS         152 C   186     Nn (executing)
  439 s   63 SOS.PUB.LPS               100 B s   1     Nn TIMER
  473    499 JSMAIN.PUB.SYS            152 B           Nn MsgPort#-32811
  485    918 CI.PUB.SYS                202 D           Nn SON, FATHER
  510 s  670 SOSTASK.PGMS.SOSDEV       253 E s  25   7 Nn TIMER
  524    538 JSMAIN.PUB.SYS            152 B           Nn MsgPort#-32809
  649     28 STRCNTL.NET.SYS           149 B           nn OTHER_WAIT

Summary:               Last 349 secs Since SHOT start  Since bootup
                       ------------- ----------------  ------------
CPU % busy                        21                 -             -
Code Page faults/sec            0.02              0.03             -
Data Page faults/sec            0.24              0.23             -
# Processes started               12                12         9,803
Process Creates/min             2.06              2.03           1.0

(Warning: MI on)

Popular wait reasons          #PINs
----------------------------  -----
TERMINAL_READ_WAIT               12
PORT_WAIT                        49

SHOT [Jobs]:
```

Figure 16.5 *SHOT ALL Command*

Figure 16.6 is an example that shows the use of restriction. In this case, the SHOT display is for jobs only.

```
SHOT [All]: jobs only
SHOT @ TUE, SEP 11, 2001, 10:20 AM

? PIN  CPU   Process Name               Pri Q Delta  %  Ic Wait State
- --- ------ -------------------------- --- - ----- --- -- --------------------
#J2    JINETD,MANAGER.SYS, MANAGER, total CPU = 4 seconds, ldev 10, jsmain 37.
   Step: <not available>
    63    606 CI.PUB.SYS                202 D           Nn SON, FATHER
     65   3855 INETD.NET.SYS            216 D           Pn MESSAGE_WAIT
#J4    SCOPEJOB,MANAGER.SYS, MANAGER, total CPU = 1754 seconds, ldev 10,
        jsmain 67.
   Step: <not available>
    69    973 CI.PUB.SYS                202 D           Nn SON, FATHER
     71 s 1655 SCOPEXL.SCOPE.SYS        100 B   825     Nn TIMER
      75 s   62 SCOPEXL2.SCOPE.SYS      152 C    14     Nn TIMER
#J5    SENDPFG,MGR.LPS, PUB, total CPU = 572 seconds, ldev 10, jsmain 70.
   Step: <not available>
    72 s  175 CI.PUB.SYS                202 D           Nn TIMER
#J105  SOSMONJ,MGR.LPS, PUB, total CPU = 751 seconds, ldev 10, jsmain 158.
   Step: <not available>
   485    918 CI.PUB.SYS                202 D           Nn SON, FATHER
    185 s   63 SOS.PUB.LPS              100 B   989     Nn TIMER
#J110  LOAD4,MGR.SOSDEV, PUB, total CPU = 2025 seconds, ldev 10, jsmain 295.
   Step: <not available>
   283   1103 CI.PUB.SYS                240 E           Nn SON, FATHER
    245    645 SOSLOAD.PGMS.SOSDEV      240 E           Pn FILE_UNBLOCKED
     510 s  696 SOSTASK.PGMS.SOSDEV     253 E s  25  20 Nn TIMER
     323 s  663 SOSTASK.PGMS.SOSDEV     253 E s  24  19 Nn TIMER
     107 s  654 SOSTASK.PGMS.SOSDEV     253 E s  24  19 Nn TIMER


Summary:               Last 127 secs Since SHOT start  Since bootup
                       ------------- ----------------  ------------
CPU % busy                        61                 -             -
Code Page faults/sec            0.01              0.01             -
Data Page faults/sec            0.82              0.26             -
# Processes started                6                18         9,809
Process Creates/min             2.83              1.51           1.0

(Warning: MI on)

SHOT [Jobs]:
```

Figure 16.6 *Restricted SHOT Display*

Figure 16.7 shows a typical SHOT **sessions only** display.

```
SHOT [Jobs]: sessions only
SHOT @ TUE, SEP 11, 2001, 10:26 AM

? PIN  CPU   Process Name               Pri Q Delta  %  Ic Wait State
- --- ------ -------------------------- --- - ----- --- -- --------------------
#S38   JIM,MGR.SOSDEV, PUB, total CPU = 168 seconds, ldev 4, jsmain 120.
   Step: <not available>
   236 s   13 CI.PUB.SYS               152 C           Nn SON, FATHER
    201   1324 SH.HPBIN.SYS            152 C           Pn TERMINAL_READ_WAIT
    60 s   10 VTSERVER.NET.SYS         152 C           Nn TERMINAL_READ_WAIT

#S42   JIM,MGR.SOSDEV, PUB, total CPU = 0 seconds, ldev 5, jsmain 204.
   Step: <not available>
    76    520 CI.PUB.SYS               152 C           Nn TERMINAL_READ_WAIT
    59    151 VTSERVER.NET.SYS         152 C           Nn TERMINAL_READ_WAIT

#S48   RODICA,MGR.LPSTOOLS, PUB, total CPU = 2 seconds, ldev 8, jsmain 473.
   Step: RUN shot
   338    640 CI.PUB.SYS               152 C           Nn SON, FATHER
    396   2250 SHOT.PUB.LPSTOOLS       191 C   264     Nn (executing)
    58    266 VTSERVER.NET.SYS         152 C    28     Nn TERMINAL_READ_WAIT

Summary:                Last 301 secs Since SHOT start  Since bootup
CPU % busy                        1   +               -             -
Code Page faults/sec             0.01+            0.01+            -
Data Page faults/sec             1.17+            0.51+            -
# Processes started             12               30          9,821
Process Creates/min               2.39             1.71           1.0

(Warning: MI on)

Popular wait reasons          #PINs
----------------------------  -----
TERMINAL_READ_WAIT                5
CHILD_WAIT                        2

SHOT [Jobs]:
```

Figure 16.7 *Sessions-only SHOT Display*

In Figure 16.8 we see how to use SHOT's **TRace pin** command.

```
SHOT [Jobs]: tr 60
This is your first TRace command ... may take a moment...

Stack Trace for PIN $3c (#60):
       PC=a.0017199c enable_int+$2c
NM* 0) SP=41842fb0 RP=a.00787004 notify_dispatcher.block_current_process+$338
NM  1) SP=41842fb0 RP=a.00788e44 notify_dispatcher+$268
NM  2) SP=41842f30 RP=a.001b8a64 wait_for_active_port+$e8
NM  3) SP=41842e30 RP=a.001b96c8 receive_from_port+$544
NM  4) SP=41842db0 RP=a.0075d5e4 extend_receive+$494
NM  5) SP=41842bb0 RP=a.00224d30 nowait_io_comp.get_any_io+$84
NM  6) SP=41842a70 RP=a.00225e20 nowait_io_comp+$2e0
NM  7) SP=41842870 RP=a.00225b0c ?nowait_io_comp+$8
         export stub: a.002c4408 IOWAIT+$bc
NM  8) SP=418423b0 RP=a.002c4338 ?IOWAIT+$8
         export stub: 242.0002bf54
NM  9) SP=418422f0 RP=242.0002f65c
NM  a) SP=41842230 RP=242.0002f4f8
         export stub: 242.00014aa8
NM  b) SP=418421b0 RP=242.00000000
     (end of NM stack)

SHOT [Jobs]:
```

Figure 16.8 *TRACE PIN Command*

The following example shows how running SHOT's TREE command on pin #1 (PROGEN) displays the entire process tree.

```
SHOT [Jobs]: tree 1
SHOT @ TUE, SEP 11, 2001, 10:32 AM

? PIN  CPU   Process Name               Pri Q Delta  %  Ic Wait State
- --- ------ -------------------------- --- - ----- --- -- --------------------
     1 s   21 PROGEN.PUB.SYS             13 A           Nn PROGEN_GLOBAL_PORT
     2 s  110 LOAD.PUB.SYS              142 B           On JUNK
     3 s  332 pm_cleanup                100 B   157     nn JUNK
     4   5678 port_facility_process      13 A    12     nn PFP_PORT0
     5 s  129 port_facility_process     152 C    32     nn pfp_port1
     6   5256 port_facility_process     152 C     1     nn pfp_port2
     7 s  120 port_facility_process     152 C    89     nn pfp_port3
     8    216 port_purger_process       152 C           nn DELAYED_PURGER_PORT
     9     18 fsipc_post_timeout        152 C           nn BIPC_TIMER_PORT
    10   2531 xm_checkpoint_server       30 A     2     nn XM_CHECK_PT_PORT
    11 s   92 xm_static_checkpoint_se   152 C    40     nn XM
    12     20 xm_static_checkpoint_se   152 C           nn XM
    13     20 xm_static_checkpoint_se   152 C           nn XM
    14     20 xm_static_checkpoint_se   152 C           nn XM
    15     76 spsnet_process_init       100 B           nn SPUNET_PORT
    16     78 repeater_process_init     152 C           nn REPEATER_PORT
    17     49 io_mgr_process             13 A           nn MsgPort#-113
    18     47 io_mgr_process             13 A           nn MsgPort#-117
    19    708 io_mgr_process             13 A           nn MsgPort#-121
    20 s   16 avr_process_init          100 B           nn AVR_PORT
    21     60 mms_process_init          100 B           nn MMS_PORT
    24   5080 cm: NMMON                 149 B           cn MESSAGE
    26   3517 NMFILE.PUB.SYS            149 B           Cn MESSAGE
    27    645 NMLOGMON.PUB.SYS          149 B           Nn MsgPort#-161
    28    370 NMLOGICS.PUB.SYS          148 B           Nn NMS_LOG_GLOBAL_PORT
    29    348 NMTRCMON.PUB.SYS          149 B           Cn MESSAGE
    30 s  370 NMCONSOL.PUB.SYS          149 B   240     Nn MsgPort#-171
    40   2846 NETCP.NET.SYS             149 B           Nn MEMORY_WAIT
    51    342 ICMPSERV.NET.SYS          152 C           Nn MEMORY_WAIT
    52    331 SOCKREG.NET.SYS           149 B           Cn TERMINAL_READ_WAIT
    53     52 PT2PNSTN.NET.SYS          149 B           Nn SigPort#-32780
    50   4380 TCPSIP.NET.SYS            149 B           Nn MsgPort#-339
    54   4268 DSDAD.NET.SYS             149 B           Nn TERMINAL_READ_WAIT
    55     95 VTSERVER.NET.SYS          149 B           Nn TERMINAL_READ_WAIT
    56     88 VTSERVER.NET.SYS          149 B           Nn TERMINAL_READ_WAIT
    62   1409 VTSERVER.NET.SYS          149 B           Nn TERMINAL_READ_WAIT
```

```
   57   2198 VTSERVER.NET.SYS          149 B           Nn TERMINAL_READ_WAIT
   61   1002 VTSERVER.NET.SYS          149 B           Nn TERMINAL_READ_WAIT
  133    649 STRCNTL.NET.SYS           149 B           Nn FILE_UNBLOCKED
  163     27 STRCNTL.NET.SYS           149 B           nn OTHER_WAIT
  170     27 STRCNTL.NET.SYS           152 E           nn OTHER_WAIT
  649     28 STRCNTL.NET.SYS           149 B           nn OTHER_WAIT
   25   1390 cm: MESSENGER             120 B           cn IO, TIMER, MESSAGE
   31   5547 io_mgr_process             13 A           nn MsgPort#-176
   32     56 io_mgr_process             13 A           nn MsgPort#-180
   33    439 io_mgr_process             13 A           nn MsgPort#-184
   34    157 io_mgr_process             13 A           nn MsgPort#-189
   35     52 io_mgr_process             13 A           nn MsgPort#-193
   36    647 io_mgr_process             13 A           nn MsgPort#-197
   38    328 LOG.PUB.SYS                50 B           Nn SYSLOG
   39     74 SYSMAIN.PUB.SYS            49 B           Nn SYSMAIN_PORT
   41    275 SPOOLMOM.PUB.SYS          100 B           Nn SPOOLER_MOM_PORT
   42    231 OUTSPOOL.PUB.SYS          152 C           Nn MsgPort#-32771
   48     70 OUTSPOOL.PUB.SYS          100 B           Nn OTHER_IO_WAIT
   44   6107 SESSION.PUB.SYS           100 B           Nn SESSIONMAIN_PORT
  120    537 JSMAIN.PUB.SYS            152 B           Nn MsgPort#-32800
  236 s   13 CI.PUB.SYS                152 C           Nn SON, FATHER
  201   1324 SH.HPBIN.SYS              152 C           Pn TERMINAL_READ_WAIT
   60 s   10 VTSERVER.NET.SYS          152 C           Nn TERMINAL_READ_WAIT
  204    514 JSMAIN.PUB.SYS            152 B           Nn MsgPort#-32808
   76    520 CI.PUB.SYS                152 C           Nn TERMINAL_READ_WAIT
   59    151 VTSERVER.NET.SYS          152 C           Nn TERMINAL_READ_WAIT
  473    499 JSMAIN.PUB.SYS            152 B           Nn MsgPort#-32811
  338    640 CI.PUB.SYS                152 C           Nn SON, FATHER
  396   5464 SHOT.PUB.LPSTOOLS         153 C s   3     Nn (executing)
   58    350 VTSERVER.NET.SYS          152 C    84     Nn TERMINAL_READ_WAIT
  210     21 JSMAIN.PUB.SYS            152 B           Nn MsgPort#-32814
   45   2649 JOB.PUB.SYS               100 B           Nn 2 ports@$1fc.4164496
                waiting on: JOBMAIN_PORT, JOB_QUEUE_PORT
   37    477 JSMAIN.PUB.SYS            152 B           Nn MsgPort#-32776
   63    606 CI.PUB.SYS                202 D           Nn SON, FATHER
   65   3855 INETD.NET.SYS             216 D           Pn MESSAGE_WAIT
   67    474 JSMAIN.PUB.SYS            152 B           Nn MsgPort#-32798
   69    973 CI.PUB.SYS                202 D           Nn SON, FATHER
   71 s 1657 SCOPEXL.SCOPE.SYS         100 B   784     Nn TIMER
   75 s   62 SCOPEXL2.SCOPE.SYS        152 C           Nn TIMER
   70    515 JSMAIN.PUB.SYS            152 B           Nn MsgPort#-32799
```

```
   72 s  175 CI.PUB.SYS                202 D           Nn TIMER
  182    748 JSMAIN.PUB.SYS            152 B           Nn MsgPort#-32801
  158    610 JSMAIN.PUB.SYS            152 B           Nn MsgPort#-32807
  485    918 CI.PUB.SYS                202 D           Nn SON, FATHER
  185 s   64 SOS.PUB.LPS               100 B   506     Nn TIMER
  295    742 JSMAIN.PUB.SYS            152 B           Nn MsgPort#-32802
  380    554 JSMAIN.PUB.SYS            152 B           Nn MsgPort#-32806
  226   1041 CI.PUB.SYS                240 E           Nn SON, FATHER
  501    655 SOSLOAD.PGMS.SOSDEV       240 E           Pn FILE_UNBLOCKED
  254 s   36 SOSTASK.PGMS.SOSDEV       253 E s  27   7 Nn TIMER
  371 s   34 SOSTASK.PGMS.SOSDEV       253 E s  26   6 Nn TIMER
  394 s   34 SOSTASK.PGMS.SOSDEV       253 E s  26   6 Nn TIMER
  524    538 JSMAIN.PUB.SYS            152 B           Nn MsgPort#-32809
  199    826 CI.PUB.SYS                202 D           Nn SON, FATHER
  439 s   64 SOS.PUB.LPS               100 B   459     Nn SHORT'WAIT
  189    585 JSMAIN.PUB.SYS            152 B           Nn MsgPort#-32812
  155    857 CI.PUB.SYS                202 D           Nn SON, FATHER
   77 s   64 SOS.PUB.LPS               100 B   497     Nn TIMER
SHOT [Jobs]:
```

Figure 16.9 *TREE Command*

SHOT's default display can easily be modified using the ADM command (see Figure 16.10) to show only those process activities that are of interest to you.

```
SHOT [Delta]: adm pin,name,pri,queue,execmode,swdepth,wait
ADM (PIN, NAME, PRI, QUEUE, EXECMODE, SWDEPTH, WAIT)
SHOT [Delta]:
SHOT @ TUE, SEP 11, 2001, 10:44 AM

PIN Process Name                Pri Q Ic S Wait State
--- --------------------------- --- - -- - ------------------------------------
  7 port_facility_process       152 C nn   pfp_port3
 30 NMCONSOL.PUB.SYS            149 B Nn   MsgPort#-171
 58 VTSERVER.NET.SYS            152 C Nn   TERMINAL_READ_WAIT
 71 SCOPEXL.SCOPE.SYS           100 B Nn   TIMER
254 SOSTASK.PGMS.SOSDEV         253 E Nn   TIMER
371 SOSTASK.PGMS.SOSDEV         253 E Nn   TIMER
394 SOSTASK.PGMS.SOSDEV         253 E Nn   TIMER
521 SHOT.PUB.LPSTOOLS           152 C Nn   (executing)

Summary:                 Last 55 secs Since SHOT start  Since bootup
                         ------------ -----------------  ------------
CPU % busy                         22                 -             -
Code Page faults/sec             0.00              0.00             -
Data Page faults/sec             0.02              0.02             -
# Processes started                 0                 0         9,840
Process Creates/min              0.00              0.00           1.0

(Warning: MI on)

Popular wait reasons           #PINs
-----------------------------  -----
TIMER_WAIT                         3
PORT_WAIT                          2

SHOT [Delta]:
```

Figure 16.10 *ADM Command*

The SWITCHDEPTH column (headed with the letter **S**) indicates how many times a process has switched between Native Mode and Compatibility Mode (see Figure 16.11).

```
SHOT [All]:

SHOT @ TUE, SEP 11, 2001, 10:58 AM

PIN Process Name                  Pri Q Ic S Wait State
--- -------------------------- --- - -- - -------------------------------------
  2 LOAD.PUB.SYS                  142 B On 1 JUNK
 25 cm: MESSENGER                 120 B cn 1 IO, TIMER, MESSAGE
 48 OUTSPOOL.PUB.SYS              100 B Nn 2 OTHER_IO_WAIT
 63 CI.PUB.SYS                    202 D Nn 2 SON, FATHER
 69 CI.PUB.SYS                    202 D Nn 2 SON, FATHER
155 CI.PUB.SYS                    202 D Nn 2 SON, FATHER
199 CI.PUB.SYS                    202 D Nn 2 SON, FATHER
226 CI.PUB.SYS                    240 E Nn 2 SON, FATHER
338 CI.PUB.SYS                    152 C Nn 2 SON, FATHER
485 CI.PUB.SYS                    202 D Nn 2 SON, FATHER

Summary:                 Last 109 secs Since SHOT start  Since bootup
                         ------------- ----------------  ------------
CPU % busy                       27   +               -             -
Code Page faults/sec              0.00+            0.00+            -
Data Page faults/sec              0.17+            0.19+            -
# Processes started               0               18          9,858
Process Creates/min               0.00             1.18           1.0

(Warning: MI on)

Popular wait reasons          #PINs
----------------------------  -----
TERMINAL_READ_WAIT               12
PORT_WAIT                        49

SHOT [All]:
```

Figure 16.11 *SWITCHDEPTH Column*

NOTE Program optimization on an MPE/iX machine often involves migrating from Compatibility Mode to Native Mode. SHOT's SWDEPTH option can be used to determine how many times your application is switching between Compatibility Mode and Native Mode. Excessive switching can result in significant performance penalties.

Figure 16.12 is an example that shows the use of the PROG command.

```
SHOT [All]: prog

Building list of programs...sorting...

#Procs   Program Name                  Capabilities
------   ----------------------------- ------------
    10 : CI.PUB.SYS                   NM:
     1 : DSDAD.NET.SYS                NM:
     1 : ICMPSERV.NET.SYS             NM:
     1 : INETD.NET.SYS                NM:
     1 : JOB.PUB.SYS                  NM:
    13 : JSMAIN.PUB.SYS               NM:
     1 : LOAD.PUB.SYS                 OCT: PrivSeg
     1 : LOG.PUB.SYS                  NM:
     1 : NETCP.NET.SYS                NM:
     1 : NMCONSOL.PUB.SYS             NM:
     1 : NMFILE.PUB.SYS               CM:  PM ph PrivSeg
     1 : NMLOGICS.PUB.SYS             NM:
     1 : NMLOGMON.PUB.SYS             NM:
     1 : NMTRCMON.PUB.SYS             CM:  PM ds mr ph PrivSeg
     2 : OUTSPOOL.PUB.SYS             NM:
     1 : PROGEN.PUB.SYS
     1 : PT2PNSTN.NET.SYS             NM:
     1 : SCOPEXL.SCOPE.SYS            (could not open, error 93)
     1 : SCOPEXL2.SCOPE.SYS           (could not open, error 93)
     1 : SESSION.PUB.SYS              NM:
     1 : SH.HPBIN.SYS                 NM:
     1 : SHOT.PUB.LPSTOOLS            NM:
     1 : SOCKREG.NET.SYS              CM:  PM ds mr ph
     3 : SOS.PUB.LPS                  NM:
     1 : SOSLOAD.PGMS.SOSDEV          NM:
     3 : SOSTASK.PGMS.SOSDEV          NM:
     1 : SPOOLMOM.PUB.SYS             NM:
     4 : STRCNTL.NET.SYS              NM:
     1 : SYSMAIN.PUB.SYS              NM:
     1 : TCPSIP.NET.SYS               NM:
     8 : VTSERVER.NET.SYS             NM:
```

Figure 16.12 *PROG Command*

SHOT Error Messages

Each SHOT error message is described in the next table.

Table 16.8 *SHOT Error Messages*

Message	Cause	Action
DEBUG requires SM or PM capability	Using the DEBUG command from within SHOT requires that you have PM.	Use GRANT from the **System Manager's Toolbox** to give yourself PM, or logon as a user that has PM.
You must have SM capability to do this.	The SHOT commands ABORTCONNS, ABORTPROCIO , BREAK, RESUME, KILL require that a user has SM.	Use GRANT from the **System Manager's Toolbox** to give yourself PM, or logon as a user that has PM.
You must have SM or OP capability to do this.	The SHOT command PRIORITY requires that a user have SM or OP.	Use GRANT from the **System Manager's Toolbox**, or logon as a user that has either SM or OP.
You must have SM or PM capability to do this.	The SHOT command TRACE requires that a user have SM or PM.	Use GRANT from the **System Manager's Toolbox** to give yourself PM, or logon as a user that has SM or PM.

17

THE TINDEX TOOL

TINDEX verifies that data on a tape backup is readable and then produces a report on all the data that has been verified. Use TINDEX to verify tapes before sending tapes to other sites. You may want to include the TINDEX report as a courtesy. TINDEX is ideal if you are backing up critical data or archiving seldom used accounts.

Operation

Conceptually, TINDEX is similar to programs like VALIDATE and VSTORE. Operationally, however, TINDEX differs in its ease of use, powerful command set, and flexible operation.

TINDEX prints a directory of the files on various kinds of tapes: STORE (NM and CM style), DBLOAD, Classic MEMDUMP, SPOOK, Classic DUS (Diagnostic Utilities Tape), HPPA INSTALL, MPE XL MEMDUMP, HPUX Core tapes. TINDEX also has limited support for UNIX TAR tapes.

While TINDEX is running, you can press **Ctrl+Y** to display the name of the file currently being processed. Additionally, a dot (.) is printed each time 50 files have been processed as that you can more easily monitor how TINDEX is progressing.

NOTE Be careful if you use TINDEX with a tape that has no file marks on it as it may spin the tape off of the end of the reel.

Background on Filenames

This section discusses operational issues related to long creator names and to the Hierarchical File Name Syntax (HFS).

Long Creator Names

Prior to MPE XL, whenever a disk file was created, MPE would record the user-id in the file label. The account name of the creator was not recorded. As of MPE/iX (and earlier for a few files), MPE records both the user-id and the account name.

This means that the first 8-bytes of the creator name is a user-id and the second 8-bytes is an account name. Files with "long" creator names are flagged with a plus (+) instead of a period (.) in front of the account name. If the account portion of the creator's name does not match the account the file is in, then the full creator name will be shown on the next line of output.

These files are typically either from an MPE/iX 4.0 (or later) system or are spoolfiles from an MPE XL 2.2 (or later) system. **:RESTORE** on pre-MPE/iX 4.5 systems has trouble restoring non-spoolfiles with long creator names and may require use of the options:

```
CREATE=CREATOR;CREATOR=<desiredname>
```

Hierarchical File System (HFS)

With the release of MPE/iX 4.5, MPE supports Hierarchical File System names (i.e., **/usr.lib/thisisalongname/too**). STORE was modified to handle the names of such files in a special manner, which is somewhat backwards-compatible with pre-Posix MPE XL :RESTORE.

When the first HFS file is seen by STORE, it generates a new file and puts it on the tape. This file appears to be called "HFSMAP._HFSGRP._HFSACCT". This file contains lines that show a mapping between the HFS name and a name like "F#######", where "#######" is a number that increments for each HFS file stored.

Here is a sample line from HFSMAP:

```
F0000000._HFSGRP._HFSACCT<-/MINER/SOURCE/foo
```

The above line means that the HFS file /MINER/SOURCE/foo was placed on the STORE tape as if its name is **F0000000._HFSGRP._HFSACCT**.

TINDEX shows the names of the HFS files as **F#######._HFSGRP._HFSACCT**, and then shows the mapping from **F#######** to HFS names.

TINDEX Report

By default, TINDEX reports on a select set of information about each file. This information includes: filename, accessed date, modified date, lockword (depending on your capabilities), and creator. Fields can be selected by including the fieldname in the option list:

```
:tindex mgr dates
```

Or you can suppress information by preceding the fieldname with the letters **NO**:

```
:tindex mgr nodates
```

Printer Output & LPSLP

TINDEX will read the specified tape and look at each file on the tape. Detailed information about each file is reported to a printer file whose formal name is LPSLP. By default, each output line

defaults to a width of 132 characters. To provide greater output control, LPSLP can be file equated to other devices. Example file equations are listed below:

`:file lpslp;dev=100`	Send output to ldev 100
`:file lpslp=$null`	No output
`:file lpslp;dev=lp,1`	Deferred output
`:file lpslp;dev=100`	Change the file designator

Usage

TINDEX can be run from either the supplied UDC or via a fully-qualified statement.

- UDC

The following UDC allows easy invocation of TINDEX:

```
tindex WHAT=" ",A=" ",B=" ",C=" ",D=" ",E=" "
RUN TINDEX.PUB.LPSTOOLS;INFO="!WHAT !A !B !C !D !E"
**************
```

Example usage:

```
tindex elvis long verify
```

Or, if you prefer to specify options via the PARM bits:

```
tindex WHAT=" ", PARM=0, MISC=" "
RUN TINDEX.PUB.LPSTOOLS;INFO="!WHAT !MISC";PARM=!PARM
**************
```

Example usage:

```
tindex elvis, 1
```

Note that the above two UDC examples show "elvis" as the tape name. Requesting a tape called "T", "TAPE", or "MYTAPE" provides the operator with little extra information. By requesting a tape called "ELVIS", the operator immediately knows which tape should be mounted.

- RUN

```
:RUN TINDEX.PUB.LPSTOOLS;INFO="tapename option1 option2 ..."
```

The **tapename** is the name of the tape. If a **tapename** is not specified, then your user name is used. If you wish to specify one or more options, you must provide a **tapename**. Multiple options can be specified by separating each with a blank space.

Capabilities

Program capabilities required include IA, BA, DS, PM and PH. User SM or OP is required if you want TINDEX to display file lockwords on the report.

Building TINDEX Reports

TINDEX's report capabilities can be formatted in a variety of ways. Specifying format and report contents is done through TINDEX options. A number of these options are specified through the INFO parameter. A few of the options can be specified by setting various PARM bits.

Table 17.1 *TINDEX Options*

Option	Description
132	132 character portrait output
ACCESSED	Displays access date
ALPHASORT	Inserts a blank line after first letter of filename change
BLKSZ	Displays block size
BUILDPV	Creates private volume account structure
COMPARE	Compares modification date
CONTENTS	Dumps tape directory to disk file
CREATED	Displays creation date
CREATOR	Displays creator
DATES	Synonym for ACCESSED, CREATED, MODIFIED
DEFERLPSLP	Assigns output priority 1 to LPSLP
DEVICE	Displays device name for every file
EOF	Displays EOF
EXTENTS	Displays extents
EXPLAIN	Displays extra information about long creator names
FCODE	Displays filecode
FGA	Displays filename.group.account
FILENUM	Displays filenumber

Option	Description
FROMDISK	FROMDISK is an internal debugging option
FULLQUICK	Displays directory only, one filename per line
HEADER	Displays header at the top of each page
HELP	Displays on-line help
HFS	Displays HFS file information
LABELLED	Labeled tape
LAND132	132 character landscape output
LAND176	176 character landscape output
LASTEXTENT	Displays last extend information
LIMIT	Displays file limit
LOCKWORD	Displays lockwords
LONG	More detailed output
MATRIX	Displays file security matrix
MINIMUM	Minimum output
MODIFIED	Displays modification date
NEWDISK	Outputs to disk instead of printer
NOTHING	Only displays filename.group.account
ONLINE	Puts tape drive on-line
PAGESIZE	Changes output page size
PORT132	132 character portrait output
PV	Synonym for BUILDPV
QUICK	Directory listing only
REARM	Periodically rearms **Ctrl-Y**
RECSZT	Displays record size
RESTOREQUICK	Produces a RESTORE compatible listing
SECTORS	Displays number of sectors

Option	Description
SECURITY	Displays file security matrix
SHOWNEW	Compares tape files against disk, only display newer
SHOWOLD	Compares tape files against disk, only display older
SHOWSAME	Compares tape files against disk, only display same
SKIP	Use fcontrol - forward skip file
TAPECONT	Dumps tape directory to disk
TRYNM	Tells TINDEX to run STORE to read the tape
TRYXC	Tells TINDEX to try to process 7980XC compressed tapes
TYPE	Displays user label information
USERLABELS	Displays user label information
VERIFY	Verifies file label and file data

Options Definitions

Listed below is a detailed description for each TINDEX option. In cases where a command is suppressed by adding "NO" as a prefix, "NO" is shown in brackets ([ ]).

In the list below, each option that controls an output column in a STORE tape report is flagged <report>. Most of these options apply only to STORE tapes, not to any other kind of tape (i.e.: TAR, SLT).

[NO] ACCESSED

<report>

ACCESSED tells TINDEX that you want to see the access date for every file (if available). NOACCESSED tells TINDEX to suppress showing the access date. ACCESSED is true by default.

[NO] ALPHASORT

ALPHASORT tells TINDEX to put a blank line between any two filenames whose first letter is different. Normally, most STORE tapes are created by storing files alphabetically within groups, and groups alphabetically within accounts. TINDEX defaults to printing a blank line after every group or account change is noticed. If your STORE tape has files in alphabetic order by filename only (rare, but it happens), then the default action can result in a large number of blank

lines. TINDEX tries to deduce when the files on a tape are in this "alphasort" order (i.e.: not in group.account order). If TINDEX fails to deduce correctly, the ALPHASORT keyword tells TINDEX to put blank lines only after the first letter of the file-part changes. NOALPHASORT reverts to the default mechanism.

[NO] BLKSZ

<report>

BLKSZ tells TINDEX that you want to see the block size for every file. NOBLKSZ tells TINDEX to suppress showing the block size. BLKSZ is the default.

NOTE BLKSZ may not be visible on "narrow" output.

[NO] BUILDPV

This command has the following syntax:

```
BUILDPV [ = pvname]
```

BUILDPV tells TINDEX to build a flat disk file (PV) which contains lines like:

```
newacct SPLASH , MGR

newacct SPLASH , MGR; onvs = PV

newgroup ASMNM.SPLASH

Hit <return> to continue, / to stop:

newgroup ASMNM.SPLASH; onvs = PV

altgroup ASMNM.SPLASH; onvs = PV

altgroup ASMNM.SPLASH; homevs = PV
```

The name of the private volume can be specified (=pvname), or it will default to **PV**. The output is written to a file whose formal name is PV.

[NO] CHECKSUM

<report>

CHECKSUM tells TINDEX to compute a checksum of the data for every file. The checksum information is displayed as the first column of output on the listing.

CHECKSUM tries to skip those fields in a file label that might change without the underlying file's data changing (i.e.: access date, ldev).

CHECKSUM tries to skip those few bytes in record 0 of CM PROG files that get modified every time the program is run (in record 0).

Due to a flaw in the implementation of "TRANSPORT" mode STORE on MPE XL, the checksums generated may vary from STORE to STORE, even if the files have not been modified.

Default: NOCHECKSUM

[NO] COMPARE

This command has the following syntax:

```
COMPARE [ = <ACcessdate | CReatedate | MOdifydate> ]
```

COMPARE tells TINDEX to compare the modification date of every file on the tape against a file of the same name on disk. If the "=" option is used, a different date from tape may be chosen instead of modification date. However, the disk file's modification date is always used, no matter which of the three possible dates from the tape copy of the file was selected.

The result is displayed as a special character in the first column after the account name. The characters used are:

<	older than disk file
>	newer than disk file
=	same date as disk file
blank	disk file exists, could not compare
*	disk file exists, could not compare
?	error occurred fetching disk file info.

CONTents

This command has the following syntax:

```
CONTents < <eol> | tapecontname >
```

Synonym for TAPECONT (see below)

[NO] CREATED

<report>

CREATED tells TINDEX that you want to see the creation date for every file (if available). NOCREATED tells TINDEX to suppress showing the creation date. CREATED is true by default.

[NO] CREATOR

<report>

CREATOR tells TINDEX to report the creator for each file (if available). NOCREATOR suppresses this column. CREATOR is true by default.

[NO] DATES

<report>

DATES is a synonym for ACCESSED, CREATED, MODIFIED. NODATES is a synonym for NOACCESSED, NOCREATED, NOMODIFIED

[NO] DEFERlpslp

DEFERLPSLP tells TINDEX to open LPSLP with output priority 1. This may be overridden with a file equation. The default is NODEFERLPSLP.

[NO] DEVICE

<report>

DEVICE tells TINDEX that you want to see the device name for every file. NODEVICE tells TINDEX to suppress showing the device name. DEVICE is the default.

NOTE Device names of "DISC" are automatically replaced by 8 blanks.

DEVICE may not be visible on "narrow" output.

[NO] EOF

<report>

EOF tells TINDEX that you want to see the end-of-file record number for every file. NOEOF tells TINDEX to suppress showing the end-of-file. EOF is the default.

NOTE EOF may not be visible on "narrow" output.

[NO] EXTENTS

<report>

EXTENTS tells TINDEX that you want to see the # of extents for every file (if available). NOEXTENTS tells TINDEX to suppress showing the # extents. EXTENTS is the default.

NOTE EOF may not be visible on "narrow" output.

[NO] EXPLAIN

<report>

Suppresses explanation of long creator names.

DEFAULT: EXPLAIN

[NO] FCODE

<report>

FCODE tells TINDEX that you want to see the file code for every file. NOFCODE tells TINDEX to suppress showing the block size. FCODE is the default.

NOTE Lower case file codes are "synthetic", and are not recognized by MPE. These include "qedit" (file code 111), as well as several dozen common file codes recognized by SIGSYSPROG.

[NO] FGA

<report>

FGA tells TINDEX that you want to see the "file.group.account" name of every file. NOFGA tells TINDEX to suppress showing the "file.group.account".

DEFAULT: FGA

NOTE It might not be very useful to suppress FGA!

[NO] FILENUM

<report>

FILENUM tells TINDEX to report the file number of each file it finds. NOFILENUM suppresses this column. FILENUM is true by default.

[NO] FROMDISC

FROMDISC is an internal debugging option.

[NO] FULLQUICK

FULLQUICK tells TINDEX to produce a modified form of the QUICK option's output (see QUICK). QUICK puts multiple file names on the same line, as long as they are in the same group and account.

FULLQUICK puts one file name per line, in the form:

```
####: file    .group   .account
```

Where **####** is a counter of the number of files reported so far.

See also: RESTOREQUICK.

NOTE It might not be very useful to suppress FGA!

[NO] HEADER

HEADER (default) tells TINDEX to produce a short header at the top of every page of output.

NOHEADER disables the page headers.

HELP

The HELP option causes the entire help file to be displayed.

[NO] LABelled

The LABELLED option tells TINDEX to make a special effort to read a labelled STORE tape. Normally, to get a directory listing of a labelled STORE tape, try running TINDEX and pointing it at the tape without any special file equates.

If this does NOT work, try running TINDEX with the LABEL option. This will cause TINDEX to ask for an unlabeled tape for Read & Write access, which allows a labelled tape to be 'sneaked' past MPE. Don't worry, TINDEX will NOT write to the tape!

If a labelled tape has a lockword, you will be asked to supply it, regardless of your capabilities.

[NO] LAND132

<report>

The LAND132 option tells TINDEX that your output is going to an HP LaserJet (or compatible) and that you want landscape orientation with 132 characters per line. Output of more than 132 characters will be truncated unless the LONG option is used.

[NO] LAND176

<report>

The LAND176 option tells TINDEX that your output is going to an HP LaserJet (or compatible) and that you want landscape orientation with 176 characters per line. Output of more than 176 characters will be truncated unless the LONG option is used.

[NO] LASTEXTENT

<report>

LASTEXTENT tells TINDEX that you want to see information about the last extent of every file (if available). NOLASTEXTENT tells TINDEX to suppress this information. LASTEXTENT is the default.

NOTE LASTEXTENT may not be visible on "narrow" output.

[NO] LIMIT

<report>

LIMIT tells TINDEX that you want to see the file limit for every file. NOLIMIT tells TINDEX to suppress showing the file limit. LIMIT is the default.

NOTE LIMIT may not be visible on "narrow" output.

[NO] LOCKword

<report>

LOCKWORD tells TINDEX to report the lockword for each file (if available, and if appropriate for your capabilities). NOLOCKWORD suppresses this column. LOCKWORD is true by default.

LONG

<report>

Causes TINDEX to print much more information about each file. LONG output will print 1 or 2 lines of information per file (depending on the recsize of the LPSLP file).

Example:

```
RUN TINDEX.PUB.LPSTOOLS;INFO="mytape LONG"
```

A nice way to use the LONG option and an HP2680A printer (the laser page printer), is to use an environment file that provides more characters per line than 132. For example, if you have an environment file (LPWIDE) that allowed 200 characters per line, you could use it and the LONG option as follows:

```
FILE LPSLP; DEV=EPOC; CCTL; ENV=LPWIDE.HPENV.SYS
RUN TINDEX.PUB.LPSTOOLS;INFO="mytape LONG"
```

[NO] MATRIX

<report>

[NO] SECURITY

MATRIX tells TINDEX that you want to see the file security matrix for every file. NOMATRIX tells TINDEX to suppress showing the file security matrix. MATRIX is the default. SECURITY is a synonym for MATRIX.

NOTE MATRIX may not be visible on "narrow" output.

[NO] MINIMUM

MINIMUM tells TINDEX that you want to have the information about the stored files on the minimum number of pages of output. This option is intended for the system operator who routinely validates STORE tapes with TINDEX, and only wants a minimum sized TINDEX output to save. NOMINIMUM is the default.

[NO] MODIFIed

<report>

MODIFIED tells TINDEX that you want to see the modification date for every file (if available). NOMODIFIED tells TINDEX to suppress showing the modification date. MODIFIED is true by default.

NEWDISK

The NEWDISK option causes TINDEX to make a human readable copy of the TINDEX output as a permanent disk file named NEWDISK.

[NO] NOTHING

<report>

NOTHING tells TINDEX that you don't want any information about files to be displayed except FGA. NOTHING is useful when you want to turn off a lot of the REPORT options, as you can use NOTHING and then turn on selected options. NONOTHING turns on all REPORT options.

[NO] ONLINE

This command has the following syntax:

```
[NO] ONLINE [=] ldev#
```

ONLINE tells TINDEX that you want have it try to set the tape drive on-line at the start of TINDEX. ONLINE uses the HPDEVCONTROL intrinsic, which has various problems.

PAGEsize

This command has the following syntax:

```
PAGEsize [=] #
```

<report>

Tells TINDEX to use a different value for determining the number of lines per printed page. Normally, TINDEX uses 60 (unless the record size of the printer is 200 or more characters, in which case 90 used). However, some printers default to a smaller page size, which can result in wasted pages being produced. If your printed output consists of a page of data followed by a page with 2 lines, followed by a page of data, followed by 2 lines (etc.), trying specifying PAGE=58 or PAGE=57.

[NO] PORT132 or 132

<report>

The PORT132 option tells TINDEX that your output is going to an HP LaserJet (or compatible) and that you want portrait orientation with 132 characters per line. Output of more than 132 characters will be truncated unless the LONG option is used.

[NO] QUICK

<report>

At the front of every STORE tape is a 'directory' which lists just the name (file.group.account) of every file found on the tape. The QUICK option causes TINDEX to print just this directory (instead of the more descriptive normal listing). As you might guess, this is very quick, but you gain speed at the cost of information. A QUICK option can print only the names of the files, it cannot print additional information.

For example:

```
RUN TINDEX.PUB.LPSTOOLS;INFO="mytape QUICK"
```

NOTE QUICK cannot be used in conjunction with most other reporting options.

[NO] REARM

REARM tells TINDEX to periodically re-arm the control-Y trap. This option should be unnecessary, but can be useful when running on a version of MPE XL that has problems "losing" control-Y.

[NO] RECSZT

<report>

RECSZT tells TINDEX that you want to see the record size of every file. NORECSZT tells TINDEX to suppress this information. RECSZT is the default. RECSIZE is a synonym for RECSZT.

NOTE RECSZT may not be visible on "narrow" output.

[NO] RESTOREquick

RESTOREQUICK tells TINDEX to produce a quick directory (from the information at the start of the reel) in a format that RESTORE would like (one file name per line, with no embedded blanks).

RESTOREQUICK always implies NOHEADER.

NOTE RESTOREQUICK cannot be used in conjunction with most other reporting options.

[NO] SECTORS

<report>

SECTORS tells TINDEX that you want to see the number of sectors each file occupies. NOSECTORS tells TINDEX to suppress this information. SECTORS is the default.

NOTE SECTORS may not be visible on "narrow" output.

[NO] SHOWNEW

SHOWNEW tells TINDEX that you only want to see those files on tape that are newer than disk files with the same name.

[NO] SHOWOLD

SHOWOLD tells TINDEX that you only want to see those files on tape that are older than disk files with the same name.

[NO] SHOWSAME

SHOWSAME tells TINDEX that you only want to see those files on tape that are the same date as disk files with the same name.

By default, SHOWxxx compares the tape file's modification date against the disk file's modification date. The tape file's creation date or access date may be selected with the COMPARE option.

Only one of SHOWNEW, SHOWOLD, and SHOWSAME may be chosen.

SKIP

The SKIP option tells TINDEX to use the 'fcontrol' option called 'forward skip file' to go from one file to the next, regardless of the type of device your tape is mounted on.

This option is meaningful only when your tape is really an HP9144 cartridge tape. This is a very slow device, and seems to take a long time when doing 'forward skip file' operations. TINDEX notices when you are using an HP9144 device, and defaults to using a 'read every record' method of getting to the end of each file for this device. Specifying SKIP allows you to tell TINDEX to use 'forward skip file' anyway.

NOTE If VERIFY is true (and it is, by default), the SKIP option is ignored. Thus, to use SKIP and NOVERIFY, the options must be specified in the order: NOVERIFY, SKIP

TAPECONT

This command has the following syntax:

TAPECONT filename

The TAPECONT option causes TINDEX to save a copy of the undecoded tape directory on disk in a file whose name follows the TAPECONT option. The following example will create a text file called "DIRCOPY":

```
RUN TINDEX.PUB.LPSTOOLS;INFO="fulldump TAPECONT dircopy"
```

This file cannot be easily read without special tools.

[NO] TRYNM

TRYNM tells TINDEX to run STORE.PUB.SYS as a child process to read and report the contents of an NM STORE tape.

Default: NOTRYNM.

[NO] TRYXC

The TRYCX option tells TINDEX to try to process compressed 7980XC tapes.

NOTE If the tape drive is not automatically decompressing such tapes, it is unlikely that TINDEX will be able to make much sense out of the data.

Default: NOTRYXC.

[NO] TYPE

<report>

TYPE tells TINDEX that you want to see the file type information for every file. NOTYPE tells TINDEX to suppress the file type information. TYPE is the default. "File type" information is similar to the "TYP" column of the LISTF,2 command.

NOTE TYPE may not be visible on "narrow" output.

[NO] USERLABELS

<report>

USERLABELS tells TINDEX that you want to see the user label information for every file. NOUSERLABELS tells TINDEX to suppress showing user label information. USERLABELS is the default.

NOTE USERLABELS may not be visible on "narrow" output.

[NO] VERify

The VERIFY option (true by default) tells TINDEX to check that every bit on the tape is readable.

The NOVERIFY option tells TINDEX to not bother verifying that every bit on the tape is readable. When NOVERIFY is specified, TINDEX reads only the file label of each file and then skips the rest of the file's data.

TINDEX PARM Bits

The following table is a summary of current PARM option bits:

Table 17.2 *PARM option bits*

Bit	Meaning	Bit Value
7	PORT132	256
8	(reserved)	
9	COMPARE (partial implementation)	64
10	NOVERIFY	32
11	SKIP	16
12	LABELLED	8
13	NEWDISK	4
14	QUICK	2
15	LONG	1

TOOLBOX STANDARDS

The ToolBox collections from Lund Performance Solutions have a uniform user interface. As a result, in addition to the commands specific to each Toolbox tool, most tools allow the commands described in “TOOLBOX STANDARDS” on page 311.

TINDEX Examples

The first example is a typical TINDEX output listing. Notice how TINDEX reports the results of its findings in a clear, concise format. TINDEX concludes its reports with a Storage summary by group.

```
: tindex "lpstape nothing creator lockword created modified accessed"

This product is licensed to: lps
TINDEX [2.23] - LPS Toolbox [A.09F]         (c) 1995 Lund Performance Solutions

Please mount reel # 1 for TAPE LPSTAPE. (reply to PIN 70)

(Printer file has 132 characters per line)
Will VERIFY by reading every tape record.


NM STORE:
   Created             : MON, OCT 16, 2000,  2:37 PM
   Options: recovery; show; fastsearch; rename; copyacd; No_Compression
There are 72 files on your tape set.
Looking for start of first user file...
Starting to read file labels now...

Tape created: 2000-10-16 @  2:37 PM

Will print one dot (.) per 10 files:
.......
***END OF TAPE SET***

Largest file: #17 = (4,320 sectors (1 MBs)

Total # of sectors on tape:  37,376 (9 MBs); total # of files: 72

VERIFY --> no errors
Bengal:/LPS/PUB: showout

DEV/CL   DFID       JOBNUM  FNAME    STATE FRM SPACE RANK PRI #C
LP       #0107      #S57    LPSLP    READY         32    D 8   1
8        #08        #S57    $STDLIST OPENED

3 FILES:
    0 ACTIVE
    2 READY; INCLUDING 2 SPOOFLES, 2 DEFERRED
    1 OPENED; INCLUDING 0 SPOOFLES
    0 LOCKED; INCLUDING 0 SPOOFLES
    2 SPOOFLES:  112 SECTORS
```

```
OUTFENCE = 14
: printo 107

 INSTMI   .PUBSYS  .LPS      15 Sep00 15 Sep00 15 Sep00 MGR
 INSTOS   .PUBSYS  .LPS      15 Sep00 15 Sep00 15 Sep00 MGR
 LPSCHECK.PUB      .LPS      15 Sep00 15 Sep00 15 Sep00 MGR
 LPSEXTND.PUB      .LPS      15 Sep00 15 Sep00 15 Sep00 MGR
 LPSINST .JOB      .LPS      15 Sep00 15 Sep00 15 Sep00 MGR
 COPYFS   .UTIL    .LPS      15 Sep00 15 Sep00 15 Sep00 MGR
 DBLOADJ .UTIL     .LPS      15 Sep00 15 Sep00 15 Sep00 MGR
 DBLOADNG.UTIL     .LPS      15 Sep00 15 Sep00 15 Sep00 MGR
 DBLOADTX.UTIL     .LPS      15 Sep00 15 Sep00 15 Sep00 MGR
 FILERPT .UTIL     .LPS      15 Sep00 15 Sep00 15 Sep00 MGR
 FILERPTD.UTIL     .LPS      15 Sep00 15 Sep00 15 Sep00 MGR
 FILERPTJ.UTIL     .LPS      15 Sep00 15 Sep00 15 Sep00 MGR
 FILUSERS.UTIL     .LPS      15 Sep00 15 Sep00 15 Sep00 MGR
 ONLINE   .UTIL    .LPS      15 Sep00 15 Sep00 15 Sep00 MGR
 PCLINK2 .UTIL     .LPS      15 Sep00 15 Sep00 15 Sep00 MGR
 PSCREEN .UTIL     .LPS      15 Sep00 15 Sep00 15 Sep00 MGR
 QUAD     .UTIL    .LPS      15 Sep00 15 Sep00 15 Sep00 MGR
 RAMUSAGE.UTIL     .LPS      15 Sep00 15 Sep00 15 Sep00 MGR
 SYSLOG   .UTIL    .LPS      15 Sep00 15 Sep00 15 Sep00 MGR
 DWNLDPFG.PUB      .LPS      15 Sep00 15 Sep00 15 Sep00 MGR
 HOLIDAYS.PUB      .LPS      15 Sep00 15 Sep00 15 Sep00 MGR
 ITEMLIST.PUB      .LPS      15 Sep00 15 Sep00 15 Sep00 MGR
 LOGHELP .PUB      .LPS      15 Sep00 15 Sep00 15 Sep00 MGR
 FCPHELP .PUB      .LPS      15 Sep00 15 Sep00 15 Sep00 MGR
 PERFCOLJ.PUB      .LPS      15 Sep00 15 Sep00 15 Sep00 MGR
 PERFREDJ.PUB      .LPS      15 Sep00 15 Sep00 15 Sep00 MGR
 FCPCOLJ .PUB      .LPS      15 Sep00 15 Sep00 15 Sep00 MGR
 PFGITEMS.PUB      .LPS      15 Sep00 15 Sep00 15 Sep00 MGR
 RCITEMS .PUB      .LPS      15 Sep00 15 Sep00 15 Sep00 MGR
 REDITEMS.PUB      .LPS      15 Sep00 15 Sep00 15 Sep00 MGR
 REPRTDEF.PUB      .LPS      15 Sep00 15 Sep00 15 Sep00 MGR
 RPTCARDJ.PUB      .LPS      15 Sep00 15 Sep00 15 Sep00 MGR
 SL94000A.PUB      .LPS      15 Sep00 15 Sep00 15 Sep00 MGR
 SOS      .PUB     .LPS      15 Sep00 15 Sep00 15 Sep00 MGR
 SOSADVIC.PUB      .LPS      15 Sep00 15 Sep00 15 Sep00 MGR
 SOSFULL .PUB      .LPS      15 Sep00 15 Sep00 15 Sep00 MGR
 SOSHELP .PUB      .LPS      15 Sep00 15 Sep00 15 Sep00 MGR
 SOSJOB   .PUB     .LPS      15 Sep00 15 Sep00 15 Sep00 MGR
```

```
SOSKIP  .PUB     .LPS      15 Sep00 15 Sep00 15 Sep00 MGR
SOSLOGX .PUB     .LPS      15 Sep00 15 Sep00 15 Sep00 MGR
SOSLOGXJ.PUB     .LPS      15 Sep00 15 Sep00 15 Sep00 MGR
SOSMONJ .PUB     .LPS      15 Sep00 15 Sep00 15 Sep00 MGR
SOSPRANJ.PUB     .LPS      15 Sep00 15 Sep00 15 Sep00 MGR
SOSPRANL.PUB     .LPS      15 Sep00 15 Sep00 15 Sep00 MGR
SOSPRDMP.PUB     .LPS      15 Sep00 15 Sep00 15 Sep00 MGR
SOSRCOM .PUB     .LPS      15 Sep00 15 Sep00 15 Sep00 MGR
SOSSNOOP.PUB     .LPS      15 Sep00 15 Sep00 15 Sep00 MGR
XFERPFG .PUB     .LPS      15 Sep00 15 Sep00 15 Sep00 MGR
ANALRPT .SAMPLE  .LPS      15 Sep00 15 Sep00 15 Sep00 MGR
CPUUTIL .SAMPLE  .LPS      15 Sep00 15 Sep00 15 Sep00 MGR
FILESPC .SAMPLE  .LPS      15 Sep00 15 Sep00 15 Sep00 MGR
DISCINFO.SAMPLE  .LPS      15 Sep00 15 Sep00 15 Sep00 MGR
DISCUTIL.SAMPLE  .LPS      15 Sep00 15 Sep00 15 Sep00 MGR
DISCVOL .SAMPLE  .LPS      15 Sep00 15 Sep00 15 Sep00 MGR
GLOBAL  .SAMPLE  .LPS      15 Sep00 15 Sep00 15 Sep00 MGR
MEMUTIL .SAMPLE  .LPS      15 Sep00 15 Sep00 15 Sep00 MGR
MODESWIT.SAMPLE  .LPS      15 Sep00 15 Sep00 15 Sep00 MGR
RESPHIST.SAMPLE  .LPS      15 Sep00 15 Sep00 15 Sep00 MGR
RESPTIME.SAMPLE  .LPS      15 Sep00 15 Sep00 15 Sep00 MGR
STOPS   .SAMPLE  .LPS      15 Sep00 15 Sep00 15 Sep00 MGR
WORKDETL.SAMPLE  .LPS      15 Sep00 15 Sep00 15 Sep00 MGR
WORKLOAD.SAMPLE  .LPS      15 Sep00 15 Sep00 15 Sep00 MGR
SNMPSEND.UTIL    .LPS      15 Sep00 15 Sep00 15 Sep00 MGR
SCOPUTIL.PUB     .LPS      15 Sep00 15 Sep00 15 Sep00 MGR
SCOPEXTJ.PUB     .LPS      15 Sep00 15 Sep00 15 Sep00 MGR
SCOPUTJ .PUB     .LPS      15 Sep00 15 Sep00 15 Sep00 MGR
SCOPALLJ.PUB     .LPS      15 Sep00 15 Sep00 15 Sep00 MGR
SCOPFORM.PUB     .LPS      15 Sep00 15 Sep00 15 Sep00 MGR
SCOPITEM.PUB     .LPS      15 Sep00 15 Sep00 15 Sep00 MGR
SCOPEXTL.PUB     .LPS      15 Sep00 15 Sep00 15 Sep00 MGR
PSITEMS .PUB     .LPS      15 Sep00 15 Sep00 15 Sep00 MGR
TOTDIV  .PUB     .LPS      15 Sep00 15 Sep00 15 Sep00 MGR


***END OF TAPE SET***

Total # of sectors on page:  37,376 (  9 MBs)

Largest file: #17 = (4,320 sectors (1 MBs)
```

```
Total # of sectors on tape:  37,376 (9 MBs); total # of files: 72

Storage summary...
Group    .Account  Files   Sectors      File# Page  Reel
--------.-------- ------- ---------- ------- ---- -------
PUBSYS   .LPS            2        352       2    1       1
PUB      .LPS            2      2,592       4    1       1
JOB      .LPS            1        128       5    1       1
UTIL     .LPS           14     13,792      19    1       1
PUB      .LPS           29     15,328      48    1       1
SAMPLE   .LPS           14        368      62    2       1
UTIL     .LPS            1         32      63    2       1
PUB      .LPS            9      4,784      72    2       1
@        .LPS           72     37,376      72    2       1


VERIFY --> no errors
```

Figure 17.1 *TINDEX Output Listing*

This is a sample output using the TINDEX COMPARE option. The comparison character is squeezed in between the ACCOUNT and CREATOR columns. In this sample, the dates were all the same. So, the only comparison character displayed is the equal sign (=).

```
: tindex "lpstape nothing creator lockword created modified accessed compare"

This product is licensed to: lps
TINDEX [2.23] - LPS Toolbox [A.09f]         (c) 1995 Lund Performance Solutions

Please mount reel # 1 for TAPE LPSTAPE. (reply to PIN 71)

(Printer file has 132 characters per line)
Will VERIFY by reading every tape record.
Will compare modify dates of files


NM STORE:
   Created             : MON, OCT 16, 2000,  2:37 PM
   Options: recovery; show; fastsearch; rename; copyacd; No_Compression
There are 72 files on your tape set.
Looking for start of first user file...
Starting to read file labels now...

Tape created: 2000-10-16 @  2:37 PM

Will print one dot (.) per 10 files:
.......
***END OF TAPE SET***

Largest file: #17 = (4,320 sectors (1 MBs)

Total # of sectors on tape:  37,376 (9 MBs); total # of files: 72

VERIFY --> no errors

: showout

DEV/CL   DFID       JOBNUM  FNAME    STATE FRM SPACE RANK PRI #C
LP       #0108      #S57    LPSLP    READY         32    D 8   1
8        #08        #S57    $STDLIST OPENED

4 FILES:
    0 ACTIVE
    3 READY; INCLUDING 3 SPOOFLES, 3 DEFERRED
```

```
    1 OPENED; INCLUDING 0 SPOOFLES
    0 LOCKED; INCLUDING 0 SPOOFLES
    3 SPOOFLES:  144 SECTORS
OUTFENCE = 14

: printo 108

 INSTMI   .PUBSYS  .LPS     <15 Sep00 15 Sep00 15 Sep00 MGR
 INSTOS   .PUBSYS  .LPS     <15 Sep00 15 Sep00 15 Sep00 MGR
 LPSCHECK.PUB      .LPS     <15 Sep00 15 Sep00 15 Sep00 MGR
 LPSEXTND.PUB      .LPS     <15 Sep00 15 Sep00 15 Sep00 MGR
 LPSINST .JOB      .LPS     <15 Sep00 15 Sep00 15 Sep00 MGR
 COPYFS   .UTIL    .LPS     <15 Sep00 15 Sep00 15 Sep00 MGR
 DBLOADJ .UTIL     .LPS     <15 Sep00 15 Sep00 15 Sep00 MGR
 DBLOADNG.UTIL     .LPS     <15 Sep00 15 Sep00 15 Sep00 MGR
 DBLOADTX.UTIL     .LPS     <15 Sep00 15 Sep00 15 Sep00 MGR
 FILERPT .UTIL     .LPS     <15 Sep00 15 Sep00 15 Sep00 MGR
 FILERPTD.UTIL     .LPS     <15 Sep00 15 Sep00 15 Sep00 MGR
 FILERPTJ.UTIL     .LPS     <15 Sep00 15 Sep00 15 Sep00 MGR
 FILUSERS.UTIL     .LPS     <15 Sep00 15 Sep00 15 Sep00 MGR
 ONLINE   .UTIL    .LPS     <15 Sep00 15 Sep00 15 Sep00 MGR
 PCLINK2 .UTIL     .LPS     <15 Sep00 15 Sep00 15 Sep00 MGR
 PSCREEN .UTIL     .LPS     <15 Sep00 15 Sep00 15 Sep00 MGR
 QUAD     .UTIL    .LPS     <15 Sep00 15 Sep00 15 Sep00 MGR
 RAMUSAGE.UTIL     .LPS     <15 Sep00 15 Sep00 15 Sep00 MGR
 SYSLOG   .UTIL    .LPS     <15 Sep00 15 Sep00 15 Sep00 MGR
 DWNLDPFG.PUB      .LPS     <15 Sep00 15 Sep00 15 Sep00 MGR
 HOLIDAYS.PUB      .LPS     <15 Sep00 15 Sep00 15 Sep00 MGR
 ITEMLIST.PUB      .LPS     <15 Sep00 15 Sep00 15 Sep00 MGR
 LOGHELP .PUB      .LPS     <15 Sep00 15 Sep00 15 Sep00 MGR
 FCPHELP .PUB      .LPS     <15 Sep00 15 Sep00 15 Sep00 MGR
 PERFCOLJ.PUB      .LPS     <15 Sep00 15 Sep00 15 Sep00 MGR
 PERFREDJ.PUB      .LPS     <15 Sep00 15 Sep00 15 Sep00 MGR
 FCPCOLJ .PUB      .LPS     <15 Sep00 15 Sep00 15 Sep00 MGR
 PFGITEMS.PUB      .LPS     <15 Sep00 15 Sep00 15 Sep00 MGR
 RCITEMS .PUB      .LPS     <15 Sep00 15 Sep00 15 Sep00 MGR
 REDITEMS.PUB      .LPS     <15 Sep00 15 Sep00 15 Sep00 MGR
 REPRTDEF.PUB      .LPS     <15 Sep00 15 Sep00 15 Sep00 MGR
 RPTCARDJ.PUB      .LPS     <15 Sep00 15 Sep00 15 Sep00 MGR
 SL94000A.PUB      .LPS     <15 Sep00 15 Sep00 15 Sep00 MGR
 SOS      .PUB     .LPS     <15 Sep00 15 Sep00 15 Sep00 MGR
```

```
SOSADVIC.PUB     .LPS     <15 Sep00 15 Sep00 15 Sep00 MGR
SOSFULL .PUB     .LPS     <15 Sep00 15 Sep00 15 Sep00 MGR
SOSHELP .PUB     .LPS     <15 Sep00 15 Sep00 15 Sep00 MGR
SOSJOB  .PUB     .LPS     <15 Sep00 15 Sep00 15 Sep00 MGR
SOSKIP  .PUB     .LPS     <15 Sep00 15 Sep00 15 Sep00 MGR
SOSLOGX .PUB     .LPS     <15 Sep00 15 Sep00 15 Sep00 MGR
SOSLOGXJ.PUB     .LPS     <15 Sep00 15 Sep00 15 Sep00 MGR
SOSMONJ .PUB     .LPS     <15 Sep00 15 Sep00 15 Sep00 MGR
SOSPRANJ.PUB     .LPS     <15 Sep00 15 Sep00 15 Sep00 MGR
SOSPRANL.PUB     .LPS     <15 Sep00 15 Sep00 15 Sep00 MGR
SOSPRDMP.PUB     .LPS     <15 Sep00 15 Sep00 15 Sep00 MGR
SOSRCOM .PUB     .LPS     <15 Sep00 15 Sep00 15 Sep00 MGR
SOSSNOOP.PUB     .LPS     <15 Sep00 15 Sep00 15 Sep00 MGR
XFERPFG .PUB     .LPS     <15 Sep00 15 Sep00 15 Sep00 MGR
ANALRPT .SAMPLE  .LPS     <15 Sep00 15 Sep00 15 Sep00 MGR
CPUUTIL .SAMPLE  .LPS     <15 Sep00 15 Sep00 15 Sep00 MGR
FILESPC .SAMPLE  .LPS     <15 Sep00 15 Sep00 15 Sep00 MGR
DISCINFO.SAMPLE  .LPS     <15 Sep00 15 Sep00 15 Sep00 MGR
DISCUTIL.SAMPLE  .LPS     <15 Sep00 15 Sep00 15 Sep00 MGR
DISCVOL .SAMPLE  .LPS     <15 Sep00 15 Sep00 15 Sep00 MGR
GLOBAL  .SAMPLE  .LPS     <15 Sep00 15 Sep00 15 Sep00 MGR
MEMUTIL .SAMPLE  .LPS     <15 Sep00 15 Sep00 15 Sep00 MGR
MODESWIT.SAMPLE  .LPS     <15 Sep00 15 Sep00 15 Sep00 MGR
RESPHIST.SAMPLE  .LPS     <15 Sep00 15 Sep00 15 Sep00 MGR
RESPTIME.SAMPLE  .LPS     <15 Sep00 15 Sep00 15 Sep00 MGR
STOPS   .SAMPLE  .LPS     <15 Sep00 15 Sep00 15 Sep00 MGR
WORKDETL.SAMPLE  .LPS     <15 Sep00 15 Sep00 15 Sep00 MGR
WORKLOAD.SAMPLE  .LPS     <15 Sep00 15 Sep00 15 Sep00 MGR
SNMPSEND.UTIL    .LPS     <15 Sep00 15 Sep00 15 Sep00 MGR
SCOPUTIL.PUB     .LPS     <15 Sep00 15 Sep00 15 Sep00 MGR
SCOPEXTJ.PUB     .LPS     <15 Sep00 15 Sep00 15 Sep00 MGR
SCOPUTJ .PUB     .LPS     <15 Sep00 15 Sep00 15 Sep00 MGR
SCOPALLJ.PUB     .LPS     <15 Sep00 15 Sep00 15 Sep00 MGR
SCOPFORM.PUB     .LPS     <15 Sep00 15 Sep00 15 Sep00 MGR
SCOPITEM.PUB     .LPS     <15 Sep00 15 Sep00 15 Sep00 MGR
SCOPEXTL.PUB     .LPS     <15 Sep00 15 Sep00 15 Sep00 MGR
PSITEMS .PUB     .LPS     <15 Sep00 15 Sep00 15 Sep00 MGR
TOTDIV  .PUB     .LPS     <15 Sep00 15 Sep00 15 Sep00 MGR

***END OF TAPE SET***
```

```
***END OF TAPE SET***

Total # of sectors on page:  37,376 (  9 MBs)

The COMPARE option places a flag character just after a file's
account name.  The following is a summary of COMPARE results:

  Flag  Meaning                                  # Occurrences
  ----  ---------------------------------------  ------------
    >   Newer than disk file                              0
    <   Older than disk file                             72
    =   Same date as disk file                            0
    -   Disc file does not exist                          0
    ?   Error during comparison                           0


Largest file: #17 = (4,320 sectors (1 MBs)

Total # of sectors on tape:  37,376 (9 MBs); total # of files: 72

Storage summary...
Group    .Account  Files   Sectors      File# Page  Reel
--------.-------- ------- ---------- ------- ---- -------
PUBSYS   .LPS            2        352       2    1       1
PUB      .LPS            2      2,592       4    1       1
JOB      .LPS            1        128       5    1       1
UTIL     .LPS           14     13,792      19    1       1
PUB      .LPS           29     15,328      48    1       1
SAMPLE   .LPS           14        368      62    2       1
UTIL     .LPS            1         32      63    2       1
PUB      .LPS            9      4,784      72    2       1
@        .LPS           72     37,376      72    2       1


VERIFY --> no errors
```

Figure 17.2 *COMPARE Option*

TINDEX Error Messages

The following message list is a summary of important messages that TINDEX may display while you are in the process of a TINDEX operation. Self-explanatory messages are not included.

Table 17.3 *TINDEX Error Messages*

Message	Cause	Action
Error Reading Tape	Can occur for a variety of reasons.	Ensure integrity of media, clean tape mechanism.

Message	Cause	Action
Failed on Forward-Skip-File on tape	Error using SKIP option on cartridge, serial disk media.	Don't use SKIP option. Ensure integrity of media.
Too many tape errors	More than 100 verify errors or more than 9 tape errors were detected.	Ensure integrity of media, clean tape mechanism.

UNSUPPORTED OPERATING SYSTEMS

If *System Manager's Toolbox* is run on a version of the operating system that it doesn't know, it will terminate with either one of these messages:

```
This is an unknown version of MPE/iX
```

```
This version of MPE/iX is unfamiliar
```

The reason for these messages is that some of the tools may be sensitive to MPE/iX operating system changes. When these changes are detected, one of the warning messages will be displayed. If you get one of these messages, you may want to contact LPS to determine if the version of MPW/iX that you are running is compatible with tools operations.

There are two ways to override the operating system check, both of which involve setting a JCW.

At the MPE/iX prompt, type:

```
:setjcw LPSMPEOK 1
```

This allows the tool to acknowledge the unknown operating system's presence without terminating.

Or, you may type:

```
:setjcw LPSMPEOK 3
```

This allows the tool to quietly continue.

B

MPE File Codes

This appendix has been included in order to provide you with a convenient way to look up file code information that is displayed when you use Toolbox utilities like BLAZE, REP, or any other tool that presents filecode information.

File codes are recorded in the file label and are available to process accessing the file through the FFILEINFO or FGETINFO intrinsic. Although any user can specify a positive integer ranging from 0 to 32767 or the mnemonic name for this parameter, certain reserved integers and mnemonic have particular system-defined meanings. This table defines the MPE reserved integer and mnemonic values.

NOTE Default is file code 0.

Table B.1 *MPE reserved integer and mnemonic values*

Integer	Mnemonic	Meaning
1024	USL	User Subprogram Library
1025	BASD	Basic Data
1026	BASP	Basic Program
1027	BASFP	Basic Fast Program
1028	RL	Compatibility Mode Relocatable Library
1029	PROG	Compatibility Mode Program File
1030	NMPROG	Native Mode Program File
1031	SL	Segmented Library
1032	NMXL	Native Mode Executable Library
1033	NMRL	Native Mode Relocatable Library

Integer	Mnemonic	Meaning
1035	VFORM	VPLUS Forms File
1036	VFAST	VPLUS Fast Forms File
1037	VREF	VPLUS Reformat File
1040	XLSAV	Cross Loader ASCII File (SAVE)
1041	XLBIN	Cross Loader Relocated Binary File
1042	XLDSP	Cross Loader ASCII File (DISPLAY)
1050	EDITQ	Edit Quick File
1051	EDTCQ	Edit KEEPQ File (COBOL)
1052	EDTCT	Edit TEXT File (COBOL)
1054	TDPDT	TDP Diary File
1055	TDPQM	TDP Proof Marked File QMARKED
1056	TDPP	TDP Proof Marked non-COBOL File
1057	TDPCP	TDP Proof Marked COBOL File
1058	TDPQ	TDP Work File
1059	TDPXQ	TDP Work File COBOL
1060	RJEPN	RJE Punch File
1070	QPROC	QUERY Procedure File
1080	KSAMK	KSAM Key File
1083	GRAPH	GRAPH Specification File
1084	SD	Self-describing File
1090	LOG	User Logging Log File
1100	WDOC	HPWORD Document
1101	WDICT	HPWORD Hyphenation Dictionary
1102	WCONF	HPWORD Configuration File
1103	W2601	HPWORD Attended Printer Environment
1110	PCELL	IFS/3000 Character Cell File

Integer	Mnemonic	Meaning
1112	PENV	IFS/3000 Environment File
1113	PCCMP	IFS/3000 Compiled Character Cell File
1114	RASTR	Graphics Image in RASTER Format
1130	OPTLF	OPT/3000 Log File
1131	TEPES	TEPE/3000 Script File
1132	TEPEL	TEPE/3000 Log File
1133	SAMPL	APS/3000 Log File
1139	MPEDL	MPEDC/DRP Log File
1140	TSR	HPToolset Root File
1141	TSD	HPToolset Data File
1145	DRAW	Drawing File for HPDRAW
1146	FIG	Figure File for HPDRAW
1147	FONT	Reserved
1148	COLR	Reserved
1149	D48	Reserved
1152	SLATE	Compressed SLATE File
1153	SLATW	Expanded SLATE Work File
1156	DSTOR	RAPID/3000 DICTDBU Utility Store File
1157	TCODE	Code File for Transact/3000 Compiler
1158	RCODE	Code File for Report/3000 Compiler
1159	ICODE	Code File for Inform/3000 Compiler
1166	MDIST	HPDESK Distribution List
1167	MTEXT	HPDESK Text
1168	MARPA	ARPA Messages File
1169	MARPD	ARPA Distribution List
1170	MCMND	HPDESK Abbreviated Commands File

Integer	Mnemonic	Meaning
1171	MFRTM	HPDESK Diary Free Time List
1172	None	Reserved
1173	MEFT	HPDESK External File Transfer Messages File
1174	MCRPT	HPDESK Encrypted Item
1175	MSERL	HPDESK Serialized (Composite) Item
1176	VCSF	Version Control System File
1177	TTYPE	Terminal Type File
1178	TVFC	Terminal Vertical Format Control File
1192	NCONF	Network Configuration File
1193	NTRAC	Network Trace File
1194	NTLOG	Network Log File
1195	MIDAS	Reserved
1211	NDIR	Reserved
1212	INODE	Reserved
1213	INVRT	Reserved
1214	EXCEP	Reserved
1215	TAXON	Reserved
1216	QUERF	Reserved
1226	VC	VC File
1227	DIF	DIF File
1228	LANGD	Language Definition File
1229	CHARD	Character Set Definition File
1230	MGCAT	Formatted Application Message Catalog
1236	BMAP	Base Map Specification File
1242	BDATA	HP Business BASIC/V Data File

Integer	Mnemonic	Meaning
1243	BFORM	HP Business BASIC/V Field Order File for VPLUS
1244	BSAVE	HP Business BASIC/V SAVE Program File
1245	BCNFG	Configuration File for Default Options for HP Business
BASIC Programs		
1246	BKEY	Function Key Definition File for Terminal
1258	PFSTA	Pathflow STATIC File
1259	PFDYN	Pathflow Dynamic File
1270	RFDCA	Revisable From DCA Data Stream
1271	FFDCA	Final Form DCA Data Stream
1272	DIU	Document Interchange Unit File
1273	PDOC	HPWORD/150 Document
1275	DFI	DISOSS Filing Information File
1276	SRI	Search Restart Information File
1401	CWPTX	Chinese Word Processor Text File
1421	MAP	HPMAP/3000 Map Specification File
1422	GAL	Reserved
1425	TTX	Reserved
1428	RDIL	HP Business Report Writer (BRW) Dictionary File CM
1429	RSPEC	BRW Specification File
1430	RSPCF	BRW Specification File
1431	REXCL	BRW Execution File
1432	RJOB	BRW Report 509 File
1433	ROUTI	BRW Intermediate Report File
1434	ROUTD	BRW Dictionary Output

Integer	Mnemonic	Meaning
1435	PRINT	BRW Print File
1436	RCONF	BRW Configuration File
1437	RDICN	BRW NM Dictionary File
1438	REXNUM	BRW NM Execution File
1441	PIF	Reserved
1461	NMOBJ	Native Mode Object File
1462	PASLIB	Pascal XL Source Library
1476	TIFF	Tag Image File Format
1477	RDF	Revisable Document Format
1478	SOF	Serial Object File
1479	GPF	Chart File for Charting Gallery Chart
1480	GPD	Data File for Charting Gallery Chart
1483	VCGPM	Virtuoso Core Generator Processed Macro File
1484	FRMAT	Formatter
1485	DUMP	Dump Files Created and Used by IDAT and DPAN
1486	NNMD0	New Wave Mail Distribution List
1491	X4HDR	X.400 Header for HP Desk Manager
1500	WP1	Reserved
1501	WP2	Reserved
1502	LO123	Lotus 123 Spread Sheet
1514	FPCF	Form Tester Command Spec File
1515	INSP	Spooler XL Input Spoolfile
1516	OUTSP	Spooler XL Output Spoolfile
1517	CHKSP	Spooler XL Checkpoint Spoolfile
1521	DSKIT	HPDesk Intrinsics Transaction File

Integer	Mnemonic	Meaning
1526	MSACK	Man Server Acknowledgment
1527	MSNON	Man Server Non-Delivery Notification
1528	MSTRC	Man Server Trace File
3333		Reserved

C

LISTF FILESET

In some commands, you may substitute wildcard characters for certain parameters, or parts of parameters, in the list. The wildcard characters count toward the eight character limit for user, group, account, and file names. These wildcard characters are defined in the table below.

Wildcard Characters Definitions

Table C.1 *Wildcard Characters Definitions*

Character	Function
@	Specifies zero or more alphanumeric characters. When used by itself, @ denotes all possible members of the set.
#	Specifies one numeric character.
?	Specifies one alphanumeric character.

Wildcard Characters Examples

The above characters can be used as follows:

Table C.2 *Wildcard Characters Examples*

Example	Description
n@	Represents all items starting with the character "n".
@n	Represents all items ending with the character "n".
n@x	Represents all items starting with the character "n" and ending with the character "x".
n###	Represents all items starting with the character "n" followed by three digits, where each digit is represented by a single number (#) sign. (The "n" may be followed by up to seven number (#) signs.)

Example	Description
?n@	Represents all items whose second character is "n".
n?	Represents all two-character items starting with the character "n".
?n	Represents all two-character items ending with the character "n".

D

STANDARD WINDOWING TERMS AND FEATURES

This section explains the terminology used in describing windows-based tools. Standard features, like function keys, are discussed in the next appendix.

Table D.1 *Standard Windowing Terms*

Term	Description
User Input	Monospace typeface (i.e., enter EXFORMF into the filename field).
Window	A rectangular area that occupies a portion of the screen and is used to display information that can be viewed easily.
Menus	
Menu Bar	A single line menu (usually shaded) with the items displayed horizontally across the top of the display.
Pull-down Menu	A menu of various options that extends down from an item in a menu bar. Menu items whose names include an ellipsis (i.e., "Forms..") have pull-down menus attached to them. You may select items within a pull-down menu the same way you would in any menu.
Arrow Keys	Horizontal arrow keys are used to move along the menu bar and choose a menu to open. You may open a menu by highlighting it in the menu bar, then pressing **Enter/ Return**. A down arrow key can also be used to open a menu. To select an item within a menu, move up and down in the menu using the vertical arrow keys. You may also type the first letter of the menu item you want twice, in quick succession, and the cursor will jump to that item starts with the same letter, the cursor will jump to the first menu item it finds with that letter.

Term	Description
Menu Walking	If you have a pull-down menu already open, using the horizontal arrow keys will automatically open neighboring pull-down menus as you move to them.
Scrolling	
Vertical Scrolling	The fastest way to move through the file. Vertical scrolling is done by using the **Prev** and **Next** keys or the **PgUp** and **PgDn** keys on a PC.
Arrow Keys	The arrow keys also scroll, but only one line or column at a time.
Scroll Lock	If you have Scroll Lock enabled for your terminal or PC, your arrow keys will not function properly. It is best to leave the Scroll Lock off.

E

STANDARD FUNCTION KEYS

This section describes a few of the standard function keys typically found in a windows-based Toolbox. Non-standard function keys that are used for Toolbox-specific operations are not covered here. Only common keys, like **Help** and **Print**, are discussed here.

NOTE Function keys are context-sensitive. This means that depending on which screen is active, some or all of these functions will be available for you to use.

HELP

Context-sensitive **Help** is always assigned to the F1 key. When F1 is pressed, a pop-up help window appears on top of the current display. This window will have a title that describes the general subject of the help material. Within the window, the cursor keys, and keypad keys (**PgUp**, **Home**, etc.) can be used to navigate through the text.

For the most part, the help text displayed in the window is based on the action you are trying to accomplish. Once the text is displayed, you can browse through the entire Help subsystem.

The help text for a ***Toolbox*** utility is stored in a text file in the HELP group. If you want, this text can be modified to better suit your needs.

PRINT

Pressing **F2** outputs a "snap-shot" of the current screen display to either a printer of a disk file. The formal file designator for the output file is LP. Output can be directed to the system line printer by issuing the following file equation:

```
:FILE LP;DEV=LP
```

If no file equation is defined for LP, the output is directed to a disk file with the name LP. To direct output to a file with a different name, use a file equation of the form:

```
:FILE LPSLP=myfile
```

REFRESH

This function is not always available. When it is available, it is typically accessed through the F5 function key.

The purpose of this operation is to refresh the entire screen display. This is occasionally necessary due to "noisy" connections to the host computer or operator messages that may disrupt the screen.

Most windowed ***LPS-Tools*** usually operate in QUIET mode, so TELL messages will not corrupt the display. WARN messages, however, cannot be avoided.

ACCEPT

This function is not always available. When it is available, it is typically accessed through the F6 function key. The purpose of this function is to accept user input from a data entry form.

NOTE In **Character mode**, this key has the same effect as the **Return** key. In **Block mode**, this key is used instead of the **Return** key.

PREVIOUS and NEXT

PREVIOUS is used in data entry windows to return to the previous field or in menus to return to the previous menu option.

NEXT is used to move to the next data entry field or menu option.

CANCEL or EXIT

These functions are typically available through the F8 function key.

CANCEL is used to terminate the current activity and return you to the previous level of activity. EXIT simply terminates the program.

ZOOM

This function key provides two functions: ZOOM IN and ZOOM OUT. The function key label displays the active function.

ZOOM IN enlarges the current window to take up the entire screen, while ZOOM OUT returns the enlarged window to its original size.

F

THE MODIFY EDITOR

MODIFY is a single-line visual mode editor used for all REDO commands and, to a greater extent, in a few of the tools.

Operations

MODIFY displays your changes on the screen as you type. The cursor rests on the same line as the text you are editing. If you type any printable key, that key will either replace the character the cursor was on, or insert the key before that character, depending on the mode. Initially, you are in transparent mode. Here, a blank will simply cause the cursor to move one space to the right. Typing any other printable character terminates transparent-mode and puts you in overwrite-mode, so the character will replace the one the cursor is on.

The 3 basic modes are:

Table F.1 *Basic Modes*

Mode Types	To Enter Mode	To Exit Mode
transparent	^T	any printable char, ^B, ^O, or ^X
overwrite	^O	^T, ^B, or ^X
insert	^B or ^	^T, ^O, or ^X

You cannot create a line longer than the maximum specified by the calling program, nor can you accidentally "lose" characters off the right edge when using insert-mode. A beep will sound when you try to execute an illegal action.

The editor has an extensive set of commands, all of which are invoked via control-characters. MODIFY is case-sensitive. A few commands are meaningful only when this editor is used from within QEDIT from Robelle Consulting, Ltd. For more information on QEDIT, consult the documentation that comes with the product.

Char	Mnemonic	Description
^A	append	Goto end-of-line. Moves the cursor to just after the last character on the line. If the line is already at the maximum length, the cursor will be placed on the last character.
^B	before	Turn on insert-mode. Turns off overwrite-mode. If you enter a character while in insert-mode, it will be put before the character the cursor is on, and the rest of the line will move to the right one.
^	before	Control up-arrow (synonym of ^B). Use ^^ instead of ^B if you are on a system console.
^C	case	Change case of current character. If the current character is a lowercase letter, it will be changed to an uppercase letter and vice-versa.
^D	delete	Delete character. Typing ^D will cause the character under the cursor to be deleted and the rest of the line moved one space to the left.
^L^D	delete end	If the cursor is just past the last character (i.e., you just did a ^L or ^A), then the ^D will delete the last character of the line.
^E	erase	Erase to end of line. This will erase all of the text from the cursor to the end of line.
^F<c>	find	Find next occurrence of "c". The cursor will be moved to the first occurrence of the character "c" to the right of the cursor. If "c" is not found, you will hear a beep.
^F<n><c>	find	Find "nth" occurrence of "c" (1<=n<=8)
^G	goof	Undo all current modifications. Restores the line of text to its original form. **NOTE** ^V, ^K, ^T^D, and ^T^V cannot be undone.
^H	backspace	Move back one char (non-destructive)
^I	tab	Skip 10 characters to right.
^J	justify	Deletes blanks from cursor to the first non-blank (does not delete that character).

Char	Mnemonic	Description
^K	add	Requests QEDIT to add a line after current line. The current line will then be re-displayed for editing and you will get to edit the new line.
^L	lengthen	Goto end-of-line (synonym of ^A). Use ^L instead of ^A if you are on a Type Ahead Engine (TAE).
^M	return	Marks end of editing a line. Returns to the caller (i.e., QEDIT) the modified line. **NOTE** ^M is the same as the Return key.
^O	overwrite	Initiate overwrite-mode and also turn off insert-mode (^B). In overwrite-mode, if you enter a character it will replace the one on the screen (i.e., overwrite it).
^P<#> <dir>		Move up/down some number of lines of text (only applicable) from QEDIT). For example, "^P3-" moves back 3 lines.
^Q	query	Displays Help information.
^S<c>	scan	Find previous occurrence of "c". The cursor will be moved to the first occurrence of "c" to the left of the current cursor position. If "c" is not found, you will hear a beep.
^S<n><c>	scan	Find nth occurrence of "c" (1<=n<=8).
^T	transparent	Terminates insert-mode and overwrite-mode. After ^T, if you type blanks, the cursor will simply move right one space without affecting the text. Transparent-mode is always turned off automatically whenever a non-blank printable character is entered, then overwrite-mode is turned on.
^T^D	delete	If done at column 1, will request caller to delete the line.
^T^V	splice	If done at column 1, will request caller to join the next line to the end of the current line. The newly spliced line will be displayed for editing.

Char	Mnemonic	Description
^U	jUmpback	Move 10 characters to left. This is the opposite of ^I. As an aid to remembering them, ^I is the same as hitting the tab key, and ^U is just to the left of ^I on the keyboard.
^V	split	Split current line (at cursor) into two lines and modify both of them.
^X	eXamine	Examine (redisplay) current line.
^Y	Abort	Terminates modify mode without changing the current line.
^W	Wordproc	Shifts into "word processor" mode. In word processor mode, the next control character is used to select a function.

Word Processing Mode Functions

Table F.2 *Word Processing Mode Functions*

Char	Description
^W^C	Compress multiple blanks into single blanks.
^W^D	Delete Word. Deletes from the cursor to the next blank and then any following blanks up to (but not including) the next non-blank.
^W^H	Toggles a flag that remembers if you have an HP110 (flag is initially off). The flag is needed because the HP110 only implements a subset of the "standard" HP26xx escape sequences.
^W^L	Draws a ruled "line", like the "LT" command in QEDIT.
^W^N	Toggles "numbered" mode. A line-number prefix will be displayed in front of a line of text only both of the following are true: (1) line numbers have been requested (either via an M command from QEDIT or via ^W^N), and (2) the line-number was passed to QZMODIFY by QEDIT (i.e., you did an M command, not an MQ command).

Char	Description
^W<c>^D	Delete all characters from cursor up to, but not including character "c". **NOTE** "c" must be a "printable" ASCII character (character code > 31). If the cursor is currently on a "c", it is deleted immediately before looking for the first "c". If "c" is not found, nothing is deleted.
^W^P	Put next character into text. This is useful when you want to put a control-character into the text. All non-printable characters will be displayed as periods (.), so they will take up one space on the line.
^W^S^D	Down-case all letters from cursor to end of line.
^W^S^U	Up-case all letters from cursor to end of line.
^W^S^T	Toggle-case all letters from cursor to end of line.
^W^T	Toggles the TypeAhead Engine (if you have one) through three states: **disabled**, **enabled**, **ignored**.
^W^V	Prints the version id of this editor.
^W?	Display the ASCII character code for the character that the cursor is on, in decimal and octal.

Symbol Chart

The following is an explanation of the symbols used above:

Table F.3 *Symbol Chart*

Symbol	Explanation
<c>	Any single character. This character will be searched for. If <c> is ^W, the search will be for a "word" (words are delimited by blanks) instead of for a single character.
<#>	Zero or more digits. For example, **^P12+** would mean move forward 12 lines. **^P3-** would mean move back 3 lines.
<n>	One of: "^A, ^B, ..., ^H" and is interpreted as the number "1, 2, ..., 8" respectively.
<dir>	A "-" to move "back", or a "+" to move "forward".

NOTE When modifying a line longer than 79 characters, some commands (i.e., ^D, ^B, ^E) will not update any line of the screen display other than the one you are on. Whenever you want to see an accurate display of your text line, press "^X" to refresh the display.

You cannot use the special keys on an HP terminal (i.e., the cursor keys, insert char, delete char, clear). If you use them by accident, a ^X will refresh the display of the line you are editing.

TypeAhead

The remaining information applies only to those users who have TypeAhead Engines (from Telamon). The TypeAhead Engine (TAE) can be in one of three states from the editor's viewpoint: **disabled**, **enabled**, or **ignored**. Each is defined below:

ignored	Editor will not do anything to either encourage or discourage the use of the TAE. This is the initial state (in most cases, however, see below).
enabled	Editor will place the TAE in single-character mode at entry and restore it to line mode at exit. This means that the HP3000 won't lose typed ahead input anymore and that the special keys (i.e., cursor keys) will work nicely.
disabled	Editor will disable TypeAhead at entry (by sending ^A^V to the TAE) and enable it at exit. In this mode, the TAE is effectively taken out of the "circuit".

With QEDIT, you configure TAE-treatment as part of the SET MODIFY VEMODIFY command:

```
SET MOD VEMODIFY              {Ignore the TAE}
SET MOD VEMODIFY TAEOFF       {TAE exists, disable it.}
SET MOD VEMODIFY TAE          {TAE exists, enable it.}
```

Additional commands are available **only** when the TAE is present and enabled:

Table F.4 *Additional CommandsL*

Command	Explanation
^W^T	Toggles the TypeAhead Engine through three states: **disabled**, **enabled**, or **ignored**.
Left arrow	The HP26xx left arrow key will move the cursor one space to the left.

Command	Explanation
Right arrow	The HP26xx right arrow key will move the cursor one space to the right.
Up arrow	Move up to the prior line of text, leaving cursor in the same column. The CRT screen is scrolled DOWN, so the line you were just editing is moved down one.
Down arrow	Move down to the next line of text, leaving cursor in the same column. The CRT screen is scrolled UP, so the line you were just editing is moved up one.
Delete char	Deletes the character under the cursor (like ^D).
Insert char	Turns on insert mode (like ^B).
Home up	Move cursor to column 1 of current line.
Home down	Move cursor to last column of current line.
Insert line	Ask QEDIT to add a new line AFTER the current line.
Delete line	Ask QEDIT to delete the current line.
^leftarrow	Moves cursor LEFT to the blank just after the nearest token to the left of the cursor. Valid ONLY if a TypeAhead Engine is present and enabled. Only available on HP264x terminals.
^rightarrow	Moves cursor RIGHT until it hits the next token. Will not move past current end of text. Valid ONLY if a TypeAhead Engine is present and enabled. Only available on HP264x terminals.

G

SETTING OPTIONS

The following list covers the standard settings that you would commonly use with System Manager's Toolbox utility after you have started it and are at that tool's prompt. These options impact how the tools behave. Any user-defined customization is achieved through these special options.

The RESET and SET commands are used for enabling or disabling options. In general, SET is equivalent to "enable" and RESET is equivalent to "disable".

When to Use Setting Options

For tools that serve very pointed, specific tasks like finding a file or changing program capabilities. setting options never really becomes an issue because users are "in and out" of these programs so quickly. But for tools that have a more multi-dimensional purpose, a typical user session could last quite a while. So, knowing how these options can affect a given utility's operation is extremely useful.

For example, the EATEMPTY option, when enabled, ignores empty input lines and continues to display the results from the command last entered. If you need to look at several screens full of information then enabling this option is very useful.

TOOLBOX STANDARDS

The ToolBox collections from Lund Performance Solutions have a uniform user interface. As a result, in addition to the commands specific to each Toolbox tool, most tools allow the following commands: //, CAPTURE, CRON, CROFF, DO, LISTREDO, RESET, SET, USE.

//

// will terminate most tools immediately.

CAPTURE

This command has the following syntax:

```
CAPTURE <captureoptions>
```

The CAPTURE command will generate a hardcopy (or a disc copy) of all (or a portion) of the screen display. The ability to enter CAPTURE as a command to most tools can be enabled by entering SET CAPTUREOK and can be disabled by entering RESET CAPTUREOK.

The interactive tools maintain a shared session-local redo stack of 40 lines. The DO, LISTREDO, and REDO commands access this stack. The options REDOOK and REDOALL affect the operation of the commands.

DO

This command has the following syntax:

```
DO [ cmd# | relative_cmd# | start_text ]
```

The DO command causes the tool to re-use the selected saved input line without re-editing.

If no options follow DO, then the most recent line is reused.

If a cmd# (i.e.: DO 5) is used, then that command is retrieved and reused.

If a relative_cmd# (i.e.: DO -3) is used, then that line is retrieved and reused. A value of -1 means: most recent, -2 means second most recent, etc.

If start_text is specified, then the most recent command that started with the same text (regardless of case) is reused.

LISTREDO

This command has the following syntax:

```
LISTREDO [ALL | *]
```

Lists the REDO stack for a tool. The REDO stack is up to 40 lines long. If the REDOALL option is false, then only the saved input lines from the current tool will be listed, otherwise the last 40 lines (regardless of what tool saved them) will be listed.

If the ALL option is specified, then all saved input lines will be listed, regardless of REDOALL and tool identity.

If the "*" option is specified, then only the current tool's saved lines will be listed, regardless of REDOALL.

REDO

This command has the following syntax:

```
REDO [ cmd# | relative_cmd# | start_text ]
```

The REDO command is very similar to the DO command. After selecting a saved input line, it then displays it for editing. When editing is done, the line is used as input. The REDO can be abandoned by pressing CTRL+Y while editing.

HELP

This command has one of the following syntax:

```
HELP  or

HELP   ?  or

HELP   command
```

The HELP command (synonym: ?) provides help information about the program in general, or about a specific command. Commands may be abbreviated, in which case HELP will display information about every command that starts with the same characters.

HELP ? will display the entire help file for a tool.

Examples:

HELP STANDARDS	displays information about Toolbox standard interface.
? SE	displays information about the SET command, and any other command beginning with "SE".

SET/RESET

These commands have the following syntax:

```
SET     <options>

RESET   <options>
```

In addition to the various SET/RESET options provided by each tool, every tool supports the following options:

<options>:

BATCH CAPTUREOK COPYLP CRON CRONOK CRONPROMPT EATEMPTY EATPROMPT ECHO MPEOK PAGING QEDITOK REDO REDOALL TERMQUIET UPSHIFT USEOK 80 132

Some users like the set/reset paradigm for turning options on/off, while other users like the set option/NOoption paradigm. The SET and RESET commands provides both styles:

SET <option>	will set the option to true.
RESET <option>	will set the option to false.
SET NO<option>	will set the option to false.
RESET NO<option>	will set the option to true.

Some of the options that end in "OK" control whether or not certain commands will be automatically recognized by the Toolbox input routine.

These options are: CAPTUREOK, CRONOK, MPEOK, REDOOK, and USEOK.

[RE]SET BATCH

The BATCH option allows the user to tell a tool that it is in a job (SET BATCH) or in a session (RESET BATCH). Every tool initially determines the value of this option by calling the WHO intrinsic. The ability to override it with a SET/RESET command is intended as a development tool for Lund Performance Solutions.

[RE]SET CAPTUREOK

If CAPTUREOK is true, then the a "pscreen" (a hard copy of the screen's current contents) can be obtained by entering the command CAPTURE at most prompts. (See also: HELP CAPTURE)

CAPTUREOK can be turned on by entering: SET CAPTUREOK

CAPTUREOK can be turned off by entering: RESET CAPTUREOK

[RE]SET COPYLP

When COPYLP is true, then a copy of all terminal output (except for input prompts) is sent to LPSLP.

COPYLP can be turned on by entering: SET COPYLP

COPYLP can be turned off by entering: RESET COPYLP

[RE]SET CROFF

[RE]SET CRON

SET CROFF turns off the "CRON" option

When CRON is true, hitting a return with no other input on the line will cause a tool to re-use the last input line.

CRON can be turned on by entering: CRON, SET CRON, or RESET CROFF

CRON can be turned off by entering: CROFF, SET CROFF, or RESET CRON

NOTENOTE See CRONOK option for more information about the CRON and CROFF commands.

[RE]SET CRONOK

When CRONOK is true, the CRON and CROFF commands may be entered at any prompt.

When CRONOK is false, the CRON and CROFF commands are not allowed. (In this case, the [RE]SET CRON command can be used to turn CRON on and off.)

[RE]SET CRONPROMPT

When CRON is true, the tool will display the default input as part of the prompt if CRONPROMPT is true.

CRONPROMPT can be turned on by entering: SET CRONPROMPT

CRONPROMPT can be turned off by entering: RESET CRONPROMPT

[RE]SET EATEMPTY

When EATEMPTY is true (and CRON is false), the tool will not "see" empty input lines. Most tools set EATEMPTY to true by default.

EATEMPTY can be turned on by entering: SET EATEMPTY

EATEMPTY can be turned off by entering: RESET EATEMPTY

[RE]SET EATPROMPT

When EATPROMPT is true, then a tool will look at the beginning of every input line to see if you did something like:

move cursor up; hit ENTER

If EATPROMPT is true, and the start of the input line matches the text you were last prompted with, then that text is stripped from your input. After the stripping is done, the remainder of the input line is treated as though it was freshly typed in. Most tools set EATPROMPT to true by default.

EATPROMPT can be turned on by entering: SET EATPROMPT

EATPROMPT can be turned off by entering: RESET EATPROMPT

[RE]SET ECHO

If ECHO is true, then all input read by the tool input routine is automatically echoed to LPSOUT. ECHO is set/reset automatically at the start of each tool, and is normally not changed by users. It is documented here for completeness.

[RE]SET MPEOK

If MPEOK is true, then any input line starting with a colon (:) is passed to the HPCICOMMAND intrinsic. Most tools set MPEOK to true by default.

MPEOK can be turned on by entering: SET MPEOK

MPEOK can be turned off by entering: RESET MPEOK

[RE]SET PAGING

If PAGING is true, and if the tool is running in a session, then most output will be "paged" (i.e.: it will pause every 22 lines or so). The HELP subsystem ALWAYS temporarily sets paging to true for sessions.

PAGING can be turned on by entering: SET PAGING

PAGING can be turned off by entering: RESET PAGING

[RE]SET PSCREENOK

PSCREENOK is a synonym for CAPTUREOK.

[RE]SET QEDITOK

If QEDITOK is true and REDOOK is true, then the 2-character sequence <escape>v will be treated as a synonym for LISTREDO. This character sequence is loaded into softkey 7 by QEDIT and labelled "Listredo".

[RE]SET REDOALL

The REDO stack maintained by the tool programs is a shared stack of 40 lines.

If REDOALL is true and REDOOK is true, then LISTREDO, DO, and REDO will see the entire stack.

If REDOALL is false and REDOOK is true, then LISTREDO, DO, and REDO will see only those redo stack entries that came from the current tool.

REDOALL is reset by default.

[RE]SET REDOOK

If REDOOK is true, then most tools support the commands DO, LISTREDO, and REDO.

REDOOK can be turned on by entering: SET REDOOK

REDOOK can be turned off by entering: RESET REDOOK

[RE]SET UPSHIFT

If UPSHIFT is true, then input will be automatically shifted to uppercase.

UPSHIFT can be turned on by entering: SET UPSHIFT

UPSHIFT can be turned off by entering: RESET UPSHIFT

[RE]SET USEOK

If USEOK is true, then most tools will allow the USE command. (See also: HELP USE)

USEOK can be turned on by entering: SET USEOK

USEOK can be turned off by entering: RESET USEOK

USE

This command has the following syntax:

```
USE[Q] <filename>
```

The USE command causes the tool to read subsequent input from the specified disc file. USE will echo its input, USEQ will not.

Because USE files may not be nested, a USE command within a USE file will close the first USE file and switch to the second file.

INDEX

Symbols

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

Z