User Manual

TOOLBOX

SYSTEM MANAGER'S

DEVELOPER'S

LUND

PERFORMANCE SOLUTIONS

Performance Beyond Expectation.

LPS-Tools

System Managers Toolbox and Developers Toolbox

User Reference Manual

Software Version A.01

Hewlett-Packard HP 3000/HP 9000 Specialists

Phone (541) 926-3800
FAX (541) 926-7723
E-mail address: support@lund.com

Lund Performance Solutions

LPS-Tools/System Managers Toolbox and Developers Toolbox• Version A.01

Printed in the United States of America

For technical support please contact:

Lund Performance Solutions

Phone: (541) 926-3800
FAX: (541) 926-7723
Email address: support@lund.com

Table of Contents

Section 2
Developers Toolbox

Introduction

Welcome to ***LPS-Tools*** from Lund Performance Solutions. We believe that you will find both the ***System Managers Toolbox*** and ***Developers Toolbox*** valuable aids in maximizing the efficiency of your day-to-day programming and system management operations. Thank you for choosing the ***LPS-Tools*** utilities and the staff of Lund Performance Solutions to assist you in maximizing your *HP3000*'s performance.

About *LPS-Tools*

The utilities that comprise the ***System Managers Toolbox*** and ***Developers Toolbox*** evolved over several years. Each utility was devised to streamline, increase performance, and help make day-to-day operations and repetitive tasks on the *HP3000* easier and more efficient. The ***System Managers Toolbox*** consists of fifteen utilities that assist in all aspects of managing the *HP3000* environment: file management, system management, and performance management. The ***Developers Toolbox*** consists of eight unique utilities that are designed to help with programming tasks, including optimized replacements for frequently called intrinsics and program modification assistance. All of the utilities that comprise the toolboxes were designed by *HP3000* professionals with years of experience. Further, these toolboxes were developed with the idea of improving existing MPE utilities and providing solutions that simply have not existed.

How Can I Make the Best Use of *LPS-Tools*?

Even if initially you are using only one or two of the utilities in a toolbox, take the time to go through this document and familiarize yourself with all of the tools in each toolbox. Over time you will find many uses for many of the tools in the toolboxes. If at any time you have any questions about their use or suggestions for improvement, please contact our Technical Support department at (541) 926-3800, or FAX the **Software Enhancement Request** form found in the back of this manual to (541) 926-7723. Also refer to Appendices C, D, E, F and G for information on functionality common to many of the tools, and on-line help and editing features.

Support Policy

When you purchase support, an advantage you receive is our staff. We are glad to help you with questions on how best to use the tools in your environment and how to improve performance overall on your *HP3000*. Our annual support fee will ensure that you get timely updates, bug fixes, documentation and extra technical help via the telephone. Whether you are demonstrating ***LPS-Tools***, or it is still under the initial warranty period, or you have purchased a support agreement, we will be glad to help you with your questions. Our support staff is available Monday through Friday from 8:00 am through 5:00 pm Pacific Time.

Supported Hardware and Software

LPS-Tools is currently supported for MPE/iX version 4.0 or greater *HP3000* systems. (For additional information, refer to Appendix A, "*Unsupported Operating Systems.*") All Hewlett-Packard terminal types are supported. For information on how to navigate the user interface, refer to Appendix D, "*Standard Windowing Terms and Features*," and Appendix E, "*Standard Function Keys*."

Before You Start

You will find the installation instructions in the next chapter. 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*," and Appendix F, "*The MODIFY Editor*." For information on the standard setting you would use for each tool, please see Appendix G, "*Setting Options*."

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 23 chapters and eight appendices. There is a chapter devoted to each tool, and each chapter is organized alphabetically within the toolbox. Section 1 contains the ***System Managers Toolbox*** tools and Section 2 details the ***Developers Toolbox*** tools.

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.

Installation

Installing the ***System Managers Toolbox*** or the ***Developers Toolbox*** (or both) is very simple. Here is a brief overview:

- Restore all the files from the tape to the LPSTOOLS account
- Stream the installation job
- The installation job creates or modifies the LPSTOOLS account structure
- That's it!

Follow these same instructions whether you're installing our products for the first time or updating an existing copy.

Step-by-Step Installation Instructions

1. If the LPSTOOLS account already exists on your system, you should first back it up. Certain files may be overlaid by new versions. After the installation is complete, you may want to restore any files you have customized. Make sure no user is running any of the ***LPS-Tools***.

2. Mount the installation tape.

3. Log on to the system as MANAGER.SYS:

   :HELLO MANAGER.SYS,PUB

4. Restore all files to the LPSTOOLS account.

   :FILE T; DEV=TAPE
   :RESTORE *T; @.@.@TAPE;ACCOUNT=LPSTOOLS;CREATE;CREATOR=MGR; SHOW

   Note: It is very important that you type the restore command exactly as shown. We distribute our files in an account whose name ends with "TAPE", but they must be restored to the LPSTOOLS account.

5. A tape reply request will appear on the console (unless you have auto-reply):

   ?time/job/pin/LDEV# for "T" on devclass (NUM)?

   Reply to the tape request (again, unless you have auto-reply):

   =REPLY pin,ldev

6. Insert your MANAGER.SYS passwords into the installation jobstream LPSINST.JOB.LPSTOOLS.

7. Stream the installation job. The job LPSINST builds/modifies the LPSTOOLS account structure as needed.

   :STREAM LPSINST.JOB.LPSTOOLS

   Note: Our installation process does not put a password on the LPSTOOLS account. You should add a password to this account to ensure system security.

Section 1

System Managers Toolbox

Chapter 1

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 can 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 based on 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, PH.

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.

Command Code	Description
CAPability	Alters program capabilities
CLose	Writes changes to a selected file
DL	Changes size of DL area
Exit	Terminates ACAP
FATAL	Modifies FATAL error bit for CM programs
HEAP	Changes size of program heap
HELP	Invokes ACAP help
Look	Synonym for Peek
MAXDATA	Changes size of MAXDATA value
NONFATAL	Modifies NONFATAL error bit for CM programs
OCTcomp	Modifies OCT flag for CM programs
ODD	Reports on programs with unusual NMHEAPs, NMSTACKs, POSIX, or priority fields.
Open	Opens a program to modify
Peek	Displays information about <progfile>
PRIVSEGs	Modifies privileged segment bit for CM programs
SET/REset	Enables and disables options
STACK	Changes program stack size
ZERODB	Modifies ZERODB flag for CM programs

Command Definitions

Listed below are detailed descriptions of the ACAP commands.

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: IA, BA, DS, PH, MR, PM, OLD, ALL. 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. *Note*: The OLD capability is only valid during a continuous session with a single program.

[+,-,=] These operators are used to designate addition, subtraction, and assignment of capabilities, respectively.

CLose

This command is used to write changes to the open file.

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 Managers Toolbox*** to make a spare copy of MYPROG and name it MYPROGPM. Then use ACAP to add PM to MYPROGPM).

DL = #

The DL (Data Limit) command is valid only for compatibility mode programs. The number is entered in half-words. Use this command to change the size of the DL area that your program will use the next time it runs.

FATAL | NOFATAL

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

HEAP = # (system default = -1) | default

This command is only valid for native mode programs. Using the HEAP command will change the size of a program's heap the next time it is run. The number entered for the heap size is in bytes.

Look <progfile>

This command is a synonym for **Peek**.

MAXDATA = #

This command is valid only for compatibility mode programs. The number is entered in half-words. Use this command to change the size of MAXDATA that your program will use the next time it runs.

NONFATAL | NONONFATAL

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

OCTcomp <on | off | # | old>

This command is used to modify the state of the OCT flag in a compatibility mode program. Options for this command are "**on**," "**off**," "**#**" or "**old**." Changing the state of the OCT flag to either ON or OFF enables or disables the execution of the program through the OCTCOMP compiler.

OCT # Forces the OCT flag word to a specified value. This option should be used with care and only on occasions where you know exactly what changing the OCT value will accomplish.

OCT old The OLD option is used to restore the OCT setting to its original value (for the current ACAP process).

ODD <progfile>

Use this command to report on program files that feature non-standard or unusual applications of NMHEAP, NMSTACK, POSIX, or priority fields.

Open <progfile> [<editcommands>]

Use OPEN to select the program to modify. ACAP must open the program for read/write access. If it is successful, ACAP displays the fully-qualified program name in the square brackets preceding the ACAP prompt. For example, if you opened MAGNET successfully, the prompt would be "[MAGNET.PUB.LPSTOOLS] ACAP:".

If it is unsuccessful, ACAP reports this information to the screen. Typical problems are: 1) write access to the program file is not allowed, or 2) you are trying to open a non-program file. The first problem can usually be traced to a) the program is being used, or b) insufficient capabilities. Additionally, ACAP edit commands can be given when the OPEN command is issued.

Note: ACAP will automatically open a file if it is specified at the time you run ACAP. For example, typing "**acap magnet**" at the colon prompt will start the ACAP program and open the MAGNET program file at the same time.

Peek <progfile>

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

PRIVSEGs | NOPRIVSEGs

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

SET | REset

Selecting this option tells ACAP to validate all changes made to compatibility mode program attributes (MAXDATA, DL, STACK). The validation is based on the calculation that says "A program's DB storage + DL + stack cannot exceed 30720 16-bit words."

STACK = # (system default = -1)

This command is valid for both native mode and compatibility mode programs. Using the STACK command provides a way to modify the stack size a program will use the next time it is run. Use the value minus one (-1) to set the stack size to the system default. For native mode programs, the number is entered in bytes. For compatibility mode programs, the number is entered in half-words.

ZERODB | NOZERODB

The two forms of this command are used to select whether or not the initially defined user DL-DB area and uninitialized portions of the DB-Q (initial) are set to zero at load time.

ACAP Command Examples

This section provides some examples of the various ACAP commands and their syntax. We have also included 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 "**myfile**") in a single command line specification.

```
:acap "myfile + pm"

ACAP [2.0] - LPS Toolbox [A.01a]          (c) 1995 Lund Performance Solutions

For Help at the ACAP prompt enter   ?
Opened: MYFILE.PUB.HENSLEY
(NM)    old Cap = ba,ia,ph; nmHeap = -1; nmStack = 2000000
(NM)    Cap = ba,ia,PM,ph; nmHeap = -1; nmStack = 2000000
Closed program file.

END OF PROGRAM
:
```

Figure 1.1 - Adding PM Capability

Note: If ACAP is run with an INFO string, then the following steps occur. First, ACAP is executed. Next, the OPEN command is called to execute INFO string instructions. And finally, the EXIT command is called to close the procedure. For Example:

:run acap.pub.lpstools
OPEN "infostring prog"
CAP + pm
CLose "infostring prog"
EXIT

Thus, a UDC could be written that would give PM capability to a program:

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

Figure 1.2 demonstrates the use of ACAP's "Peek" and "OCT" commands. *Note*: MPE/iX 4.5 users have the option of using POSIX file references.

```
:acap
ACAP [2.0] - LPS Toolbox [A.01a]          (c) 1995 Lund Performance Solutions

For Help at the ACAP prompt enter   ?

ACAP: peek /SYS/PUB/SPL
(CM)    Cap = ba,ia,ds,ph; Maxdata = 32767; DL = 0; Stack = 3500; OCT'ed

ACAP: open spl.pub.sys
Opened: SPL.PUB.SYS
(CM)    Cap = ba,ia,ds,ph; Maxdata = 32767; DL = 0; Stack = 3500; OCT'ed

[SPL.PUB.SYS] ACAP: oct off
Updated to:
(CM)    new Cap = ba,ia,ds,ph; Maxdata = 32767; DL = 0; Stack = 3500
      ; OCT'ed/disabled

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

ACAP: exit
:
```

Figure 1.2 - 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.

```
:acap
ACAP [2.0] - LPS Toolbox [A.01a]          (c) 1995 Lund Performance Solutions

For Help at the ACAP prompt enter   ?

ACAP: open myfile
Opened: MYFILE.PUB.HENSLEY
(NM)   Cap = ba,ia; nmHeap = -1; nmStack = 2000000

[MYFILE.PUB.HENSLEY] ACAP: +pm+ph;stack=5000000
Updated to:
(NM)   new Cap = ba,ia,PM,ph; nmHeap = -1; nmStack = 5000000

[MYFILE.PUB.HENSLEY] ACAP: close
Closed program file.

ACAP: exit
:
```

Figure 1.3 -Multiple Commands on a Single Line

ACAP Error Messages

Message	***DL value less than -1 may cause problems.***
Cause	User entered a value for DL that is less than -1.
Action	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.
Message	***Error fetching file system information***
Cause	ACAP could not obtain file information about the program the user specified.
Action	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.
Message	***Error reading record #0 of NMPRG file:***
Cause	ACAP could not obtain the program header information.
Action	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.
Message	***Expected a capability***
Cause	User entered "CAP=" without giving a new value for CAP.
Action	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	***Expected DL value:***
Cause	User entered "DL=" without giving a new value for DL.
Action	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	***Expected MAXDATA value:***
Cause	User entered "MAXDATA=" without giving a new value for MAXDATA.
Action	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.
Message	***Expected nmHEAP value:***
Cause	User entered "HEAP=" without giving a new value.
Action	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.
Message	***Expected OCT FLAGS value:***
Cause	User entered "OCT" without an option specified.
Action	Valid options for the "OCT" command are ON, OFF, #, OLD where # denotes a number.
Message	***Expected STACK value:***
Cause	User entered "STACK=" without giving a new value.
Action	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.
Message	***Expected: -, +, or = after CAP***
Cause.	Used an invalid capability operator
Action	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	***Failed to open***
Cause	When ACAP tried to open the program specified, it encountered an error.
Action	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.
Message	***Failed to open ASDF: Nonexistent permanent file***
Cause	User specified a program file that ACAP was not able to open.
Action	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 "LISTF @,2" to examine the files in the current group.
Message	***Failed to post changes to program file:***
Cause	When ACAP tried to post changes to the program file it failed.
Action	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.

Message	***File is not an NMPRG or PROG file!***
Cause	User tried to open a non-program file.
Action	ACAP can only modify the attributes of native mode program or compatibility mode program files. Use the MPE command "LISTF @,2" to review the files in your directory for `NMPRG' or `PROG' filecodes.
Message	***HEAP value less than -1 may cause problems.***
Cause	User entered a value for HEAP that is less than -1.
Action	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.
Message	***MAXDATA value less than -1 may cause problems.***
Cause	User entered a value for MAXDATA that is less than -1.
Action	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.
Message	***MAXDATA must be in range -32768...32767:***
Cause	User entered a value for MAXDATA that is less than -32768 or greater than 32767.
Action	Input a valid value within the range specified above.
Message	***Note: program is in use, so we are sharing it!***
Cause	Selected program is being accessed elsewhere in the system.
Action	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.
Message	***Oops...this program file has an unexpected header record***
Cause	The structure of this program's header does not conform to known information.
Action	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.
Message	***Option not available for NMPRG***
Cause	User tried to alter a NMPRG attribute with a PROG attribute.
Action	Valid native mode attributes for modification are: NMHEAP, NMSTACK, and CAPs.
Message	***Program file must be OPENed first***
Cause	Open program before edit.
Action	User tried to alter program attributes without first selecting a program to alter.
Message	***Program file not open!***
Cause	User must open program first.
Action	The user must use the "open" command to select a program file to alter before any of its attributes may be modified.
Message	***STACK value less than -1 may cause problems.***
Cause	User has entered a value for STACK that is less than -1.
Action	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.

Message	***Unable to obtain write access to program.***
Cause	ACAP could not obtain write access to the program file.
Action	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.

Message	***Unknown capability, expected one of:***
Cause	User entered an invalid capability.
Action	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	***Unknown edit option***
Cause	User entered unknown command.
Action	Input a valid ACAP command specified on the Command Summary page.

Chapter 2

The BETIMES Tool

The BETIMES tool allows you to change the current time or date on an *HP3000* S/9xx without having to reboot the computer.

Operation

It is not uncommon for most computer centers to need to change the system time several times a year. Doing this on the *HP3000* S/9xx using standard methods involves many separate steps. A typical scenario would include scheduling time for the reboot, shutting the system down, and then performing the reboot. During the reboot stage the time could then be changed. With BETIMES, the process is dynamic and requires only a few minutes to accomplish the task.

There are many reasons why you may need to change the system time. For most computer centers the switch between Daylight Savings and Standard time creates a need to change the system clock biannually. Other reasons include testing time-sensitive software, or correcting an improperly set clock. BETIMES allows the date and/or time to be changed via a simple user interface. The new date or time can be entered directly with the DATE or TIME commands, or you can simply enter the difference between the current setting and the new setting. For example, if the desired setting is one hour less than the current setting, then the date or time could be changed by just that amount. Incremental updates are accomplished using the ADD and SUBTRACT commands.

For MPE/iX versions 5.0 and newer, BETIMES uses the MPE/iX SETCLOCK command, which can be overridden with the RESET SETCLOCK command.

Capabilities

Program capabilities required include IA, BA, PM, DS, and PH. BETIMES requires that the user have SM or OP capability.

HP3000 Time Settings

On *HP3000* S/9xx computers there are two clocks: a hardware clock, and a software clock. The hardware clock maintains the time in GMT (Greenwich Mean Time) and is battery backed-up. The software clock maintains the time in 24-hour "local" format. Modifying the hardware clock involves rebooting and running the ISL tool program, CLKUTIL. Refer to the HP manual entitled, *System Startup and Shutdown,* for directions on using the CLKUTIL program.

Without BETIMES, modifying the software clock must be accomplished during a 10 to 15-second window in the start-up dialogue of the reboot process. Should you miss this opportunity, you would need to reboot the system again until you successfully change the clock.

In addition to the hardware and software clocks, a GMT offset is stored in an EEPROM in the computer. BETIMES does not update the GMT offset due to the EEPROM's characteristics (an EEPROM can only be written to a finite number of times). Since BETIMES does not write to the EEPROM, the software clock needs to be reset each time the computer is rebooted.

The correct hardware time setting for your time zone is obtained by adding the GMT offset (the amount by which your time zone differs from GMT) to the actual GMT time.

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>"

Two additional UDCs are supplied for bi-annual use:

SPRINGAHEAD (Adds an hour to the /iX clock)
FALLBEHIND (Subtracts an hour from the /iX clock)

Command Summary

Following is a summary list of BETIMES commands.

Command Code	Description
ADD	Adds to current date and time
CAPture	Captures screen memory
DATE	Sets to specified value
DO	Reuses the last input line
Exit	Terminates BETIMES
HELP	Invokes BETIMES help
LISTRedo	Lists the REDO stack for the ***LPS-Tool***
NOW	Displays current date and time
REDO	Allows you to edit the last input line
RESET SETCLOCK	Overrides the MPE/iX SETCLOCK command
SUBtract	Subtracts from current date and time
TIME	Synonym for DATE
USE[Q] <file>	Reads in a disk file

Command Definitions

This section describes BETIMES commands in detail.

ADD <number> <quantity> [<number> <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

<number> a decimal number (with the above limits).

CAPTURE [PARTIAL [FLAT]]
[FLAT]

The CAPTURE command will generate a printout or a disk copy for all (or a portion) of the screen memory. Use the PARTIAL option to capture a portion of screen memory. Use the FLAT option to capture to a disk file. FLAT is the formal file designator. It may be file equated to another name. For example, if you are running SHOT and you want to perform a screen capture to the file FOO, you would type the following statements:

```
BETIMES: :file flat=foo
BETIMES: capture flat
```

: <COMMAND>

A colon (:) followed by an MPE command or UDC name is passed to the HPCICOMMAND intrinsic.

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.

DO [#]
[RELATIVE #]
[TEXT]

The DO command causes BETIMES to re-use a selected input line without re-editing. If no options follow DO, then the last line is reused. If a number (e.g., DO 5) is specified, then whatever happens to be on that line in the REDO stack is reused.

If a RELATIVE number (e.g., DO -3) is used, then that line is reused. *Note*: A value of -1 means most recent, -2 means second most recent, and so on.

If TEXT is specified, then the most recent command that started with the same text (regardless of case) is used.

EXIT or "/"

Terminates the ***LPS-Tools*** program immediately.

HELP

The HELP command displays 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 set of characters.

Typing help ? will display the entire help file for any tool.

Help Examples:

HELP STANDARDS	Displays information about the ***Toolbox*** standard interface.
Help SE	Displays information about the SET command, and any other command beginning with "SE".

LISTREDO [ALL]
[*]

Lists the REDO stack. If the ALL option is used, the REDO stack is displayed for all input for every tool you may have used in the current session. If the asterisk (*) option is used, the REDO stack is displayed for input for the ***LPS-Tool*** you are currently using. The asterisk (*) option is the default setting.

The REDO stack holds up to 40 commands. Each ***LPS-Tool*** "remembers" its REDO stack between invocations during a single session. So, even if you exit a ***LPS-Tool***, your REDO stack will be preserved the next time you restart the ***LPS-Tool***.

NOW

This command simply displays the current date and time.

REDO [#]
[RELATIVE #]
[TEXT]

The REDO command is identical to the DO command except that you can edit the displayed input line. The MODIFY editor, which is documented in an appendix, is used to alter the input line as required. Once you have finished editing the input line, press **Return**. REDO can be abandoned by pressing "Ctrl-Y" while editing.

If a number (e.g., REDO 5) is specified, then whatever happens to be on that line in the REDO stack is reused.

If a RELATIVE number (e.g., REDO -3) is specified, then that line is retrieved and reused. A value of -1 means most recent, -2 means second most recent, and so on.

If TEXT is specified, then the most recent command that started with the same text (regardless of case) is reused.

RESET SETCLOCK

For MPE/iX versions 5.0 this command is used to override the MPE/iX SETCLOCK command.

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).

USE[Q] <filename>

The USE command causes the ***LPS-Tool*** to read subsequent input from the specified disk file. USE echoes input; USEQ does not. USE files may not be nested. Using USE files is a great way to store your own ***LPS-Tools*** settings.

BETIMES Examples

Example 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, at the BETIMES prompt enter:

SUB 1 Y

Or:

SUB 1 year

Example 3 shows how the SPRINGAHEAD UDC adjusts for Daylight Savings.

```
:springahead
BETIMES [2.0] - LPS Toolbox [A.01a]          (c) 1995 Lund Performance Solutions

For Help at the BETIMES prompt enter   ?

MPE/iX 5.0 Push (or later)

You appear to have 1 CPU
Updated time to: 1995/12/15 12:52:51
:
```

Figure 2.1 - SPRINGAHEAD UDC

Example 4 shows how the FALLBEHIND UDC adjusts for Standard time.

```
:fallbehind
BETIMES [2.0] - LPS Toolbox [A.01a]          (c) 1995 Lund Performance Solutions

For Help at the BETIMES prompt enter   ?

MPE/iX 5.0 Push (or later)

You appear to have 1 CPU
Updated time to: 1995/12/15 11:52:59
:
```

Figure 2.2 - FALLBEHIND UDC

BETIMES Error Messages

Message	***OOPS: TIMER intrinsic no longer works!***
Cause	BETIMES is not compatible with the version of /iX (or XL) on which it is running.
Action	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.
Message	***Unable to locate TIMER globals.***
Cause	Probably due to changes HP made to the operating system between major updates (i.e., patches).
Action	Call Lund Performance Solutions Technical Support. Please provide the Build-id for your operating system, machine series, and BETIMES version when you call.
Message	***You need SM or OP capability to run this program.***
Cause	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.
Action	Have the system manager change the time, or run the GRANT ***System Managers Toolbox*** to give yourself SM capability.

Chapter 3

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 command 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.

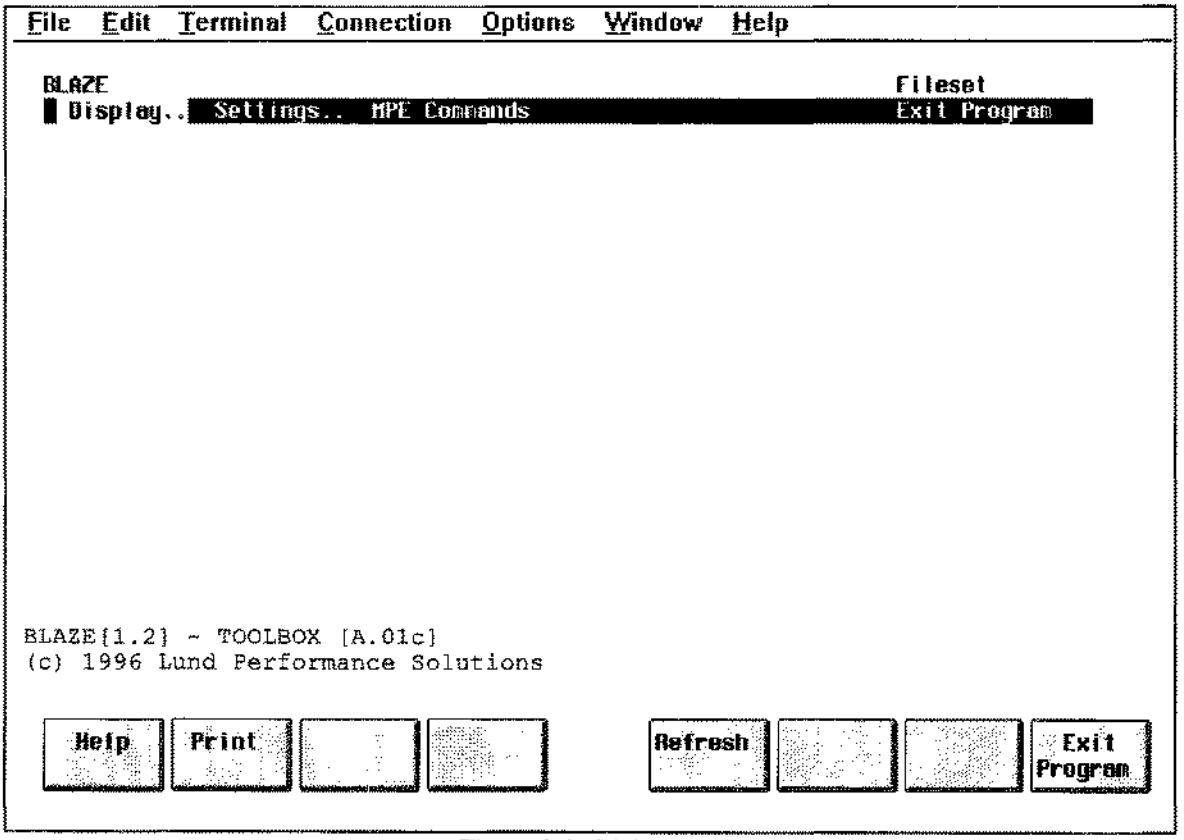


Figure 3.1 - Main Menu

The Display.. Menu

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

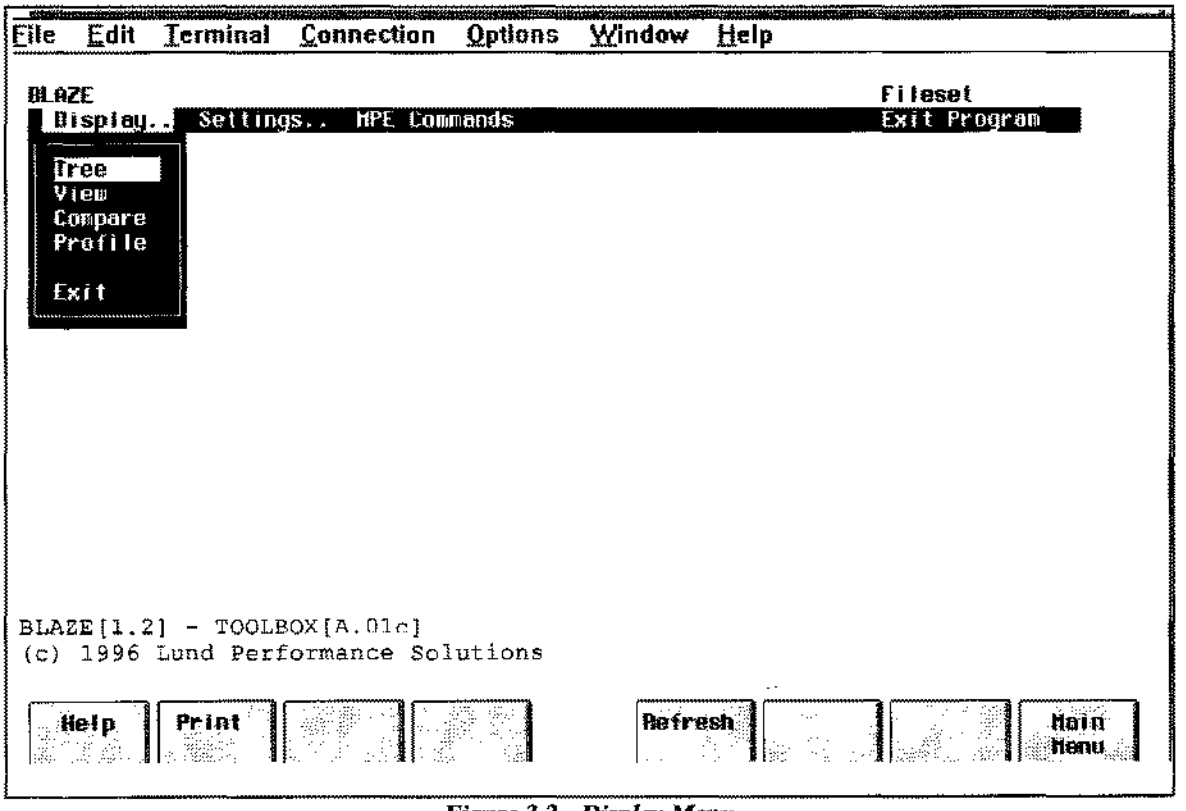


Figure 3.2 - Display Menu

The Settings.. Menu

The Settings menu is for user-customization 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.

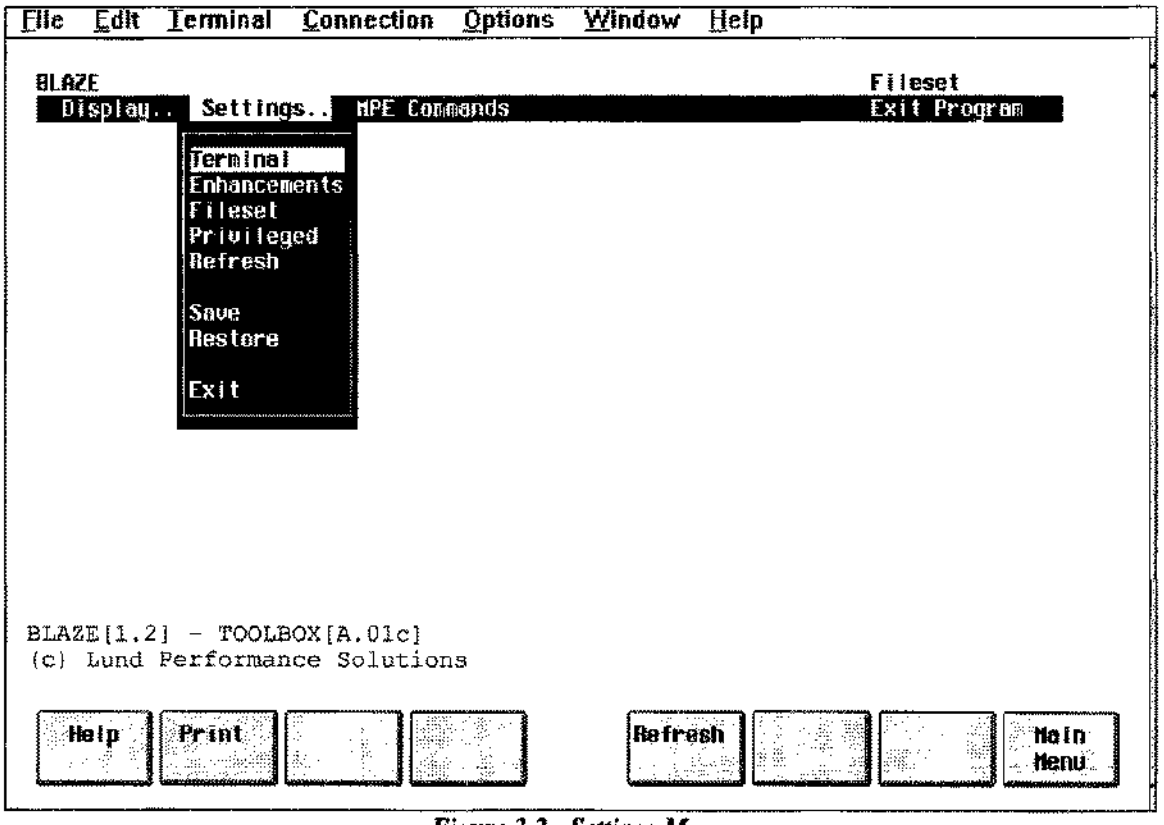


Figure 3.3 - Settings Menu

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.

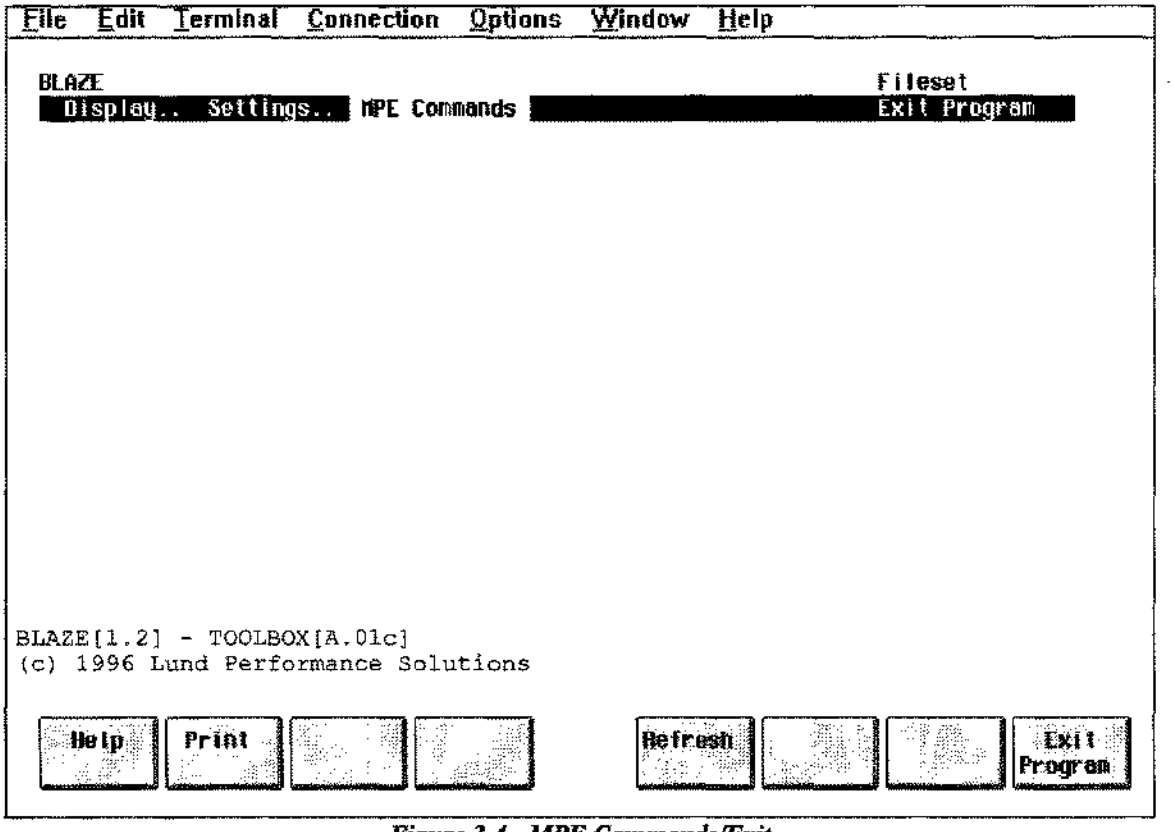


Figure 3.4 - MPE Commands/Exit

Display Selection Menus

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.

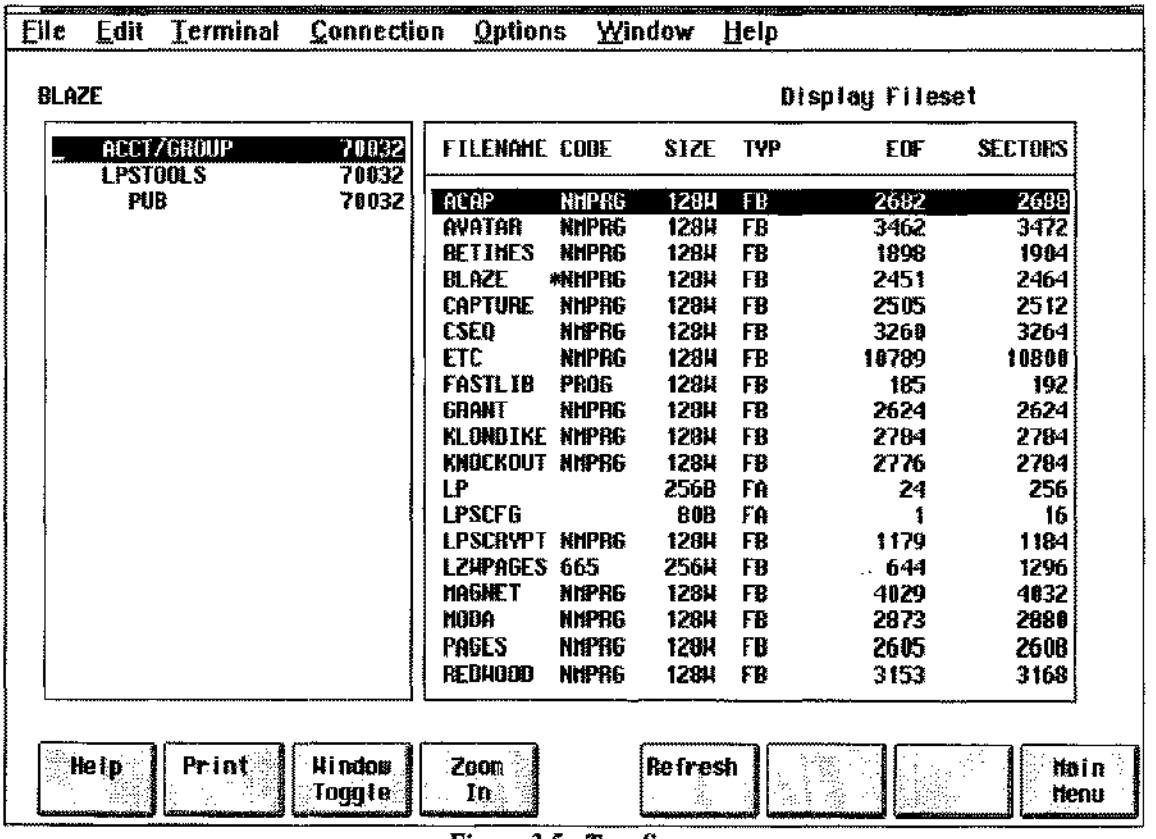


Figure 3.5 - Tree Screen

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.

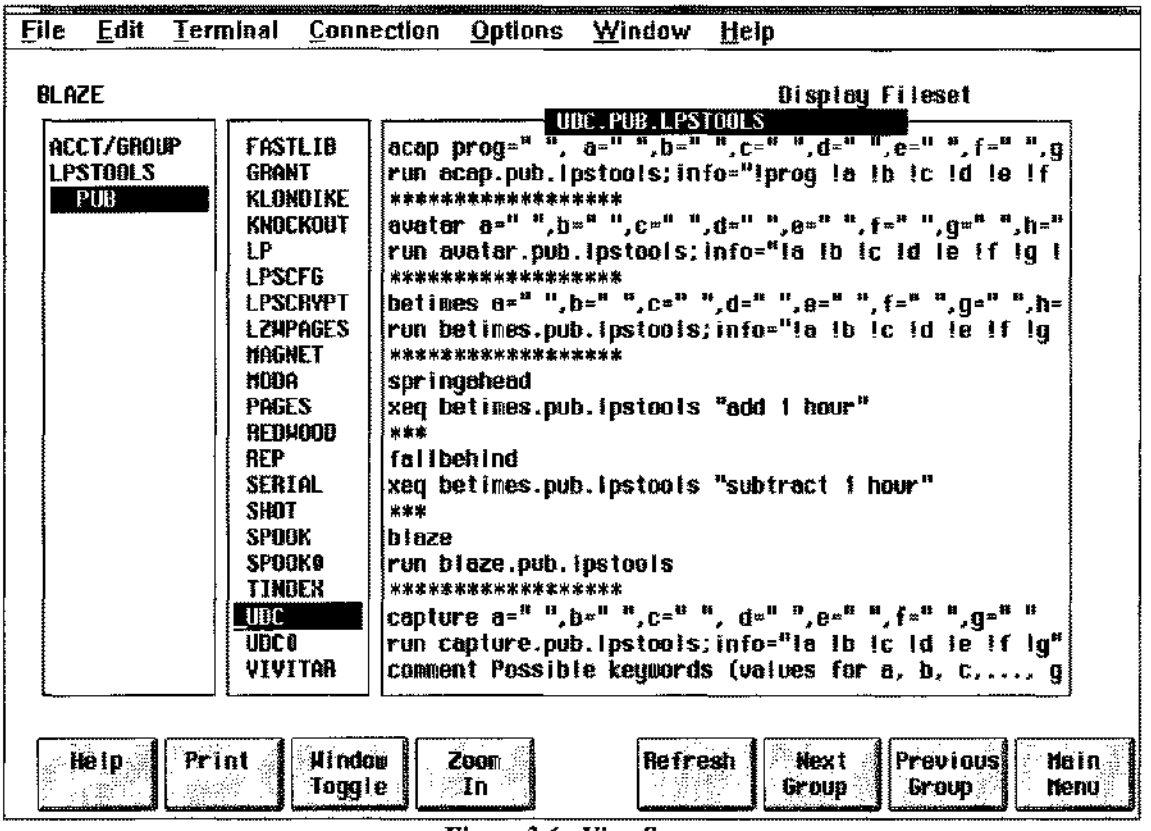


Figure 3.6 - View Screen

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.

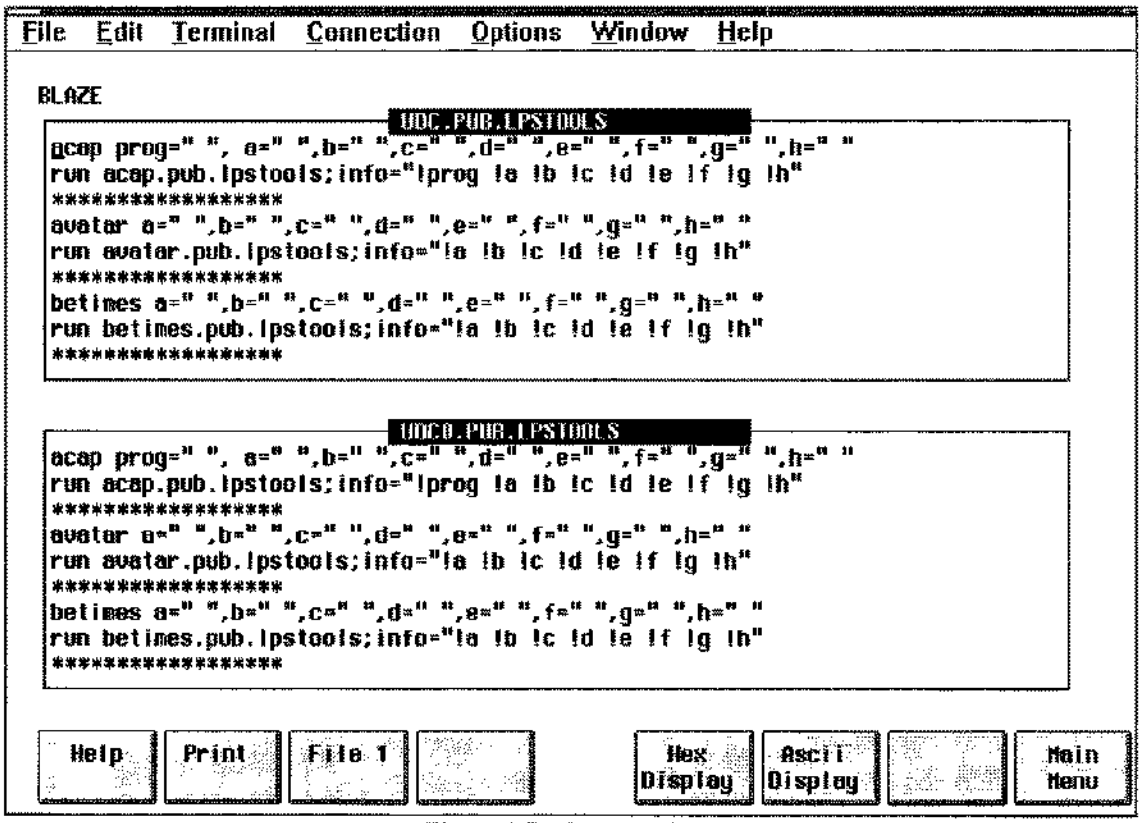


Figure 3.7 - Compare Screen

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.

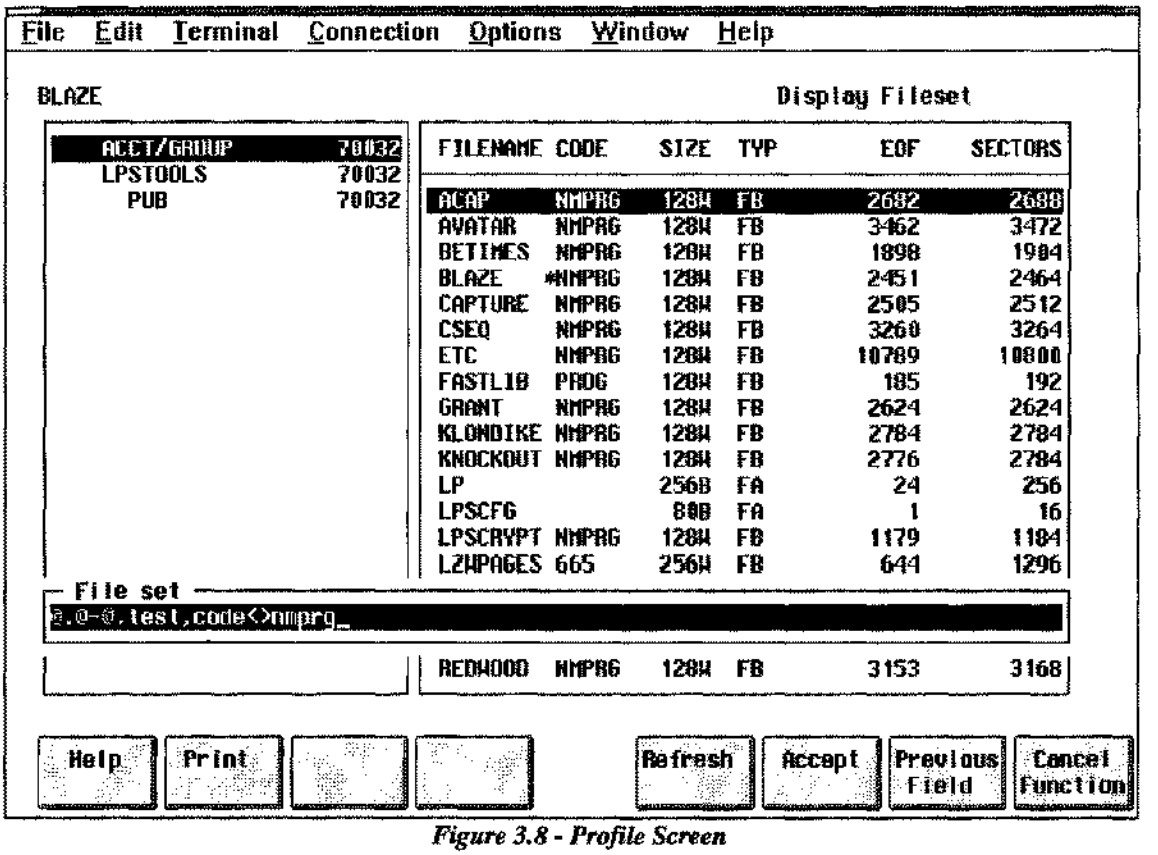


Figure 3.8 - Profile Screen

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 3.9 - Fileset Specification Diagram

BLAZE Parameters

One of the most powerful features of BLAZE is its fileset specification syntax. The syntax diagram in *Figure 3.9* 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 plus (+) 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 of 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 length 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, include 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 files. For example, list all files in this group except message files, and native mode relocatable libraries:

@, msg = off, code <> nmrl

RIO

This definition represents Relative I/O files. This filter lets you specify whether or not to include RIO files. 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 filename 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.

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)

TASK	LETTER	DESCRIPTION
	X	eXpand (activate the previous file subset)
	N	Next subset (activate the next file subset)
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 3.10*). In this window you can define a new file specification using the syntax described earlier in the "*Fileset 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.

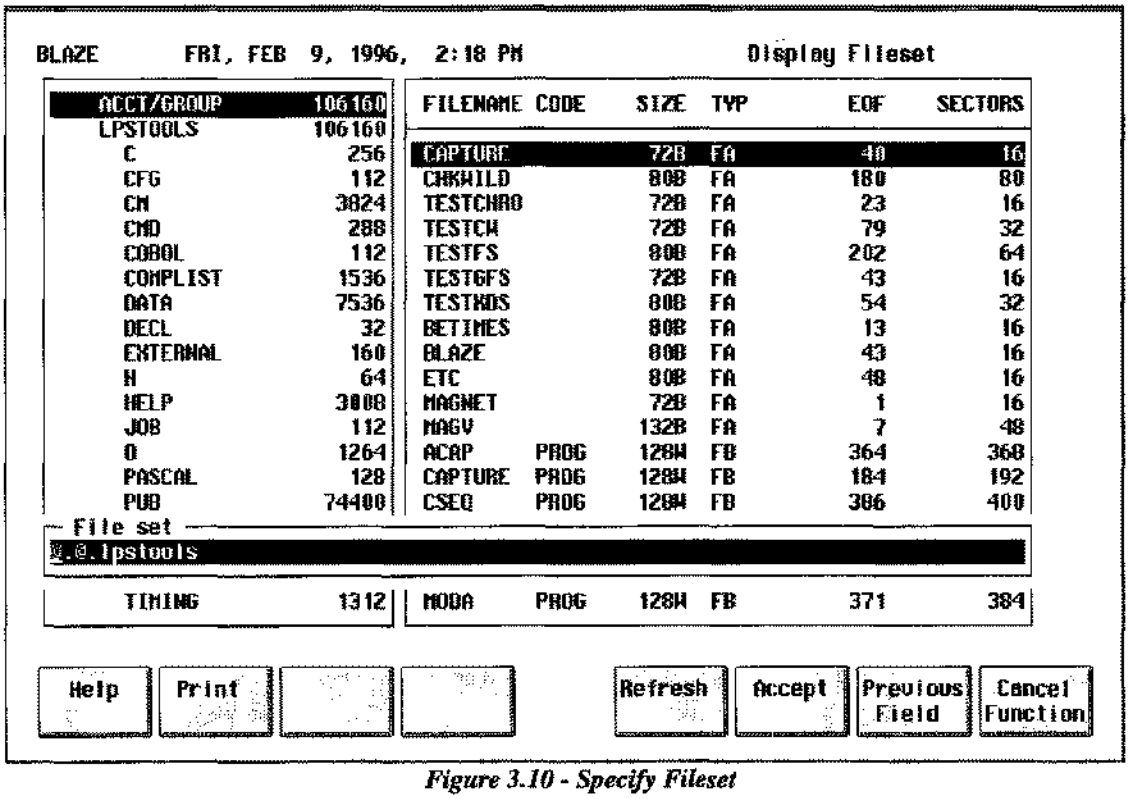


Figure 3.10 - Specify Fileset

M

The M command pops up a window on top of the current display (see *Figure 3.11*). This window is titled "*Words to search.*" The parameters entered here are passed programmatically to MAGNET in the ***System Managers 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 the 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 the 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.

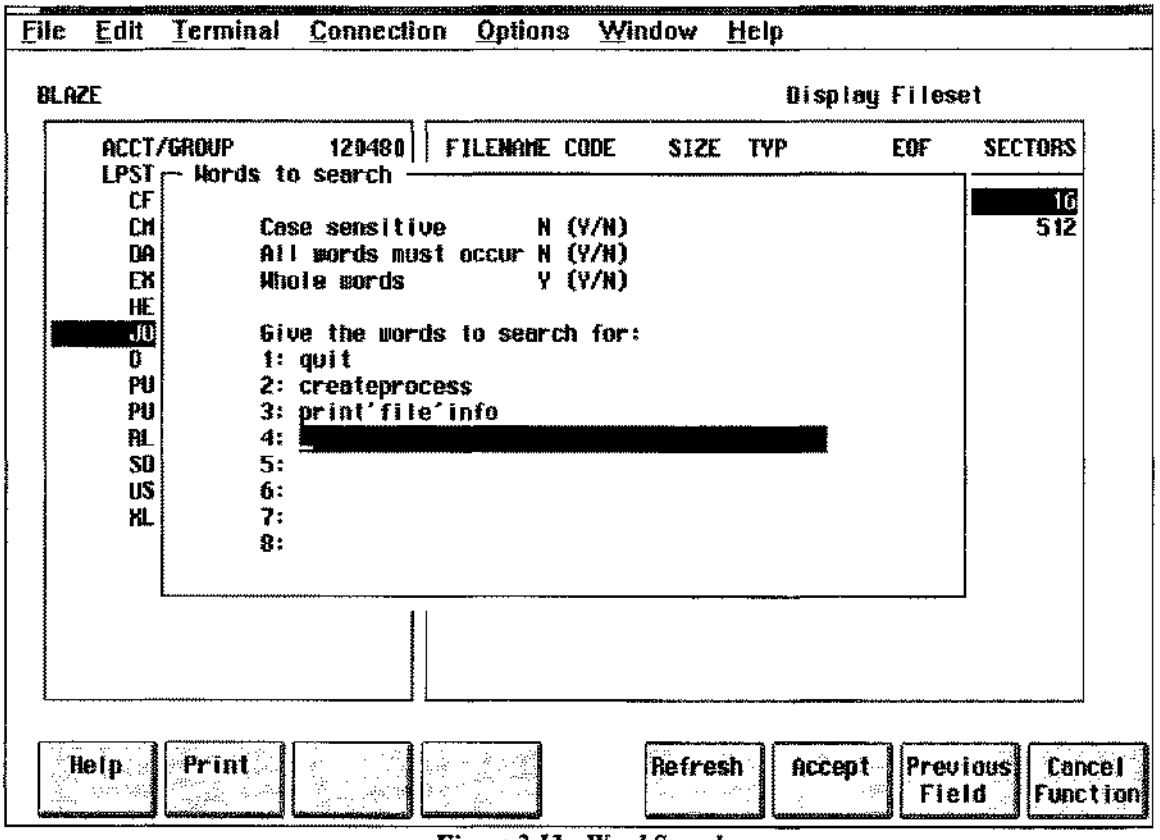


Figure 3.11 - Word Search

Figure 3.11 illustrates how you can locate the files that use a few common intrinsics. Something like this might be handy when migrating 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 Untag 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 the 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 the 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 3.12*). 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.

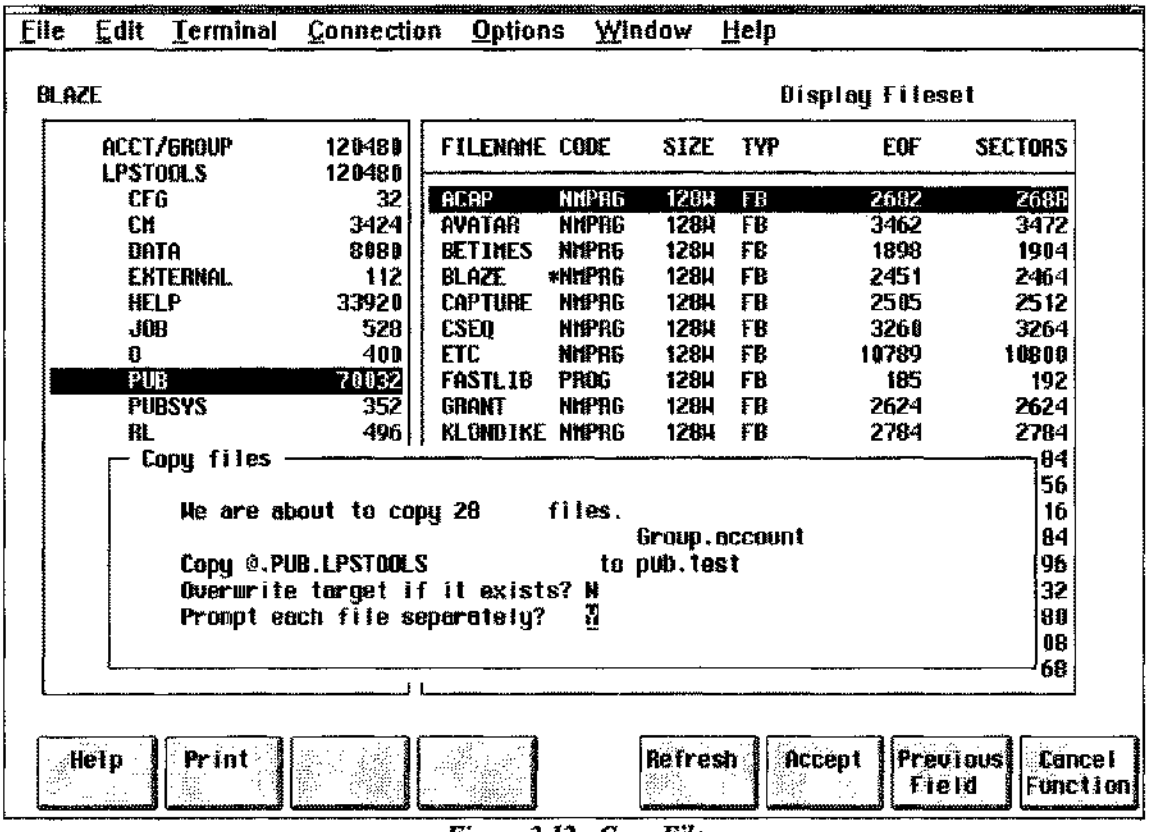


Figure 3.12 - Copy Files

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 3.13*). 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.

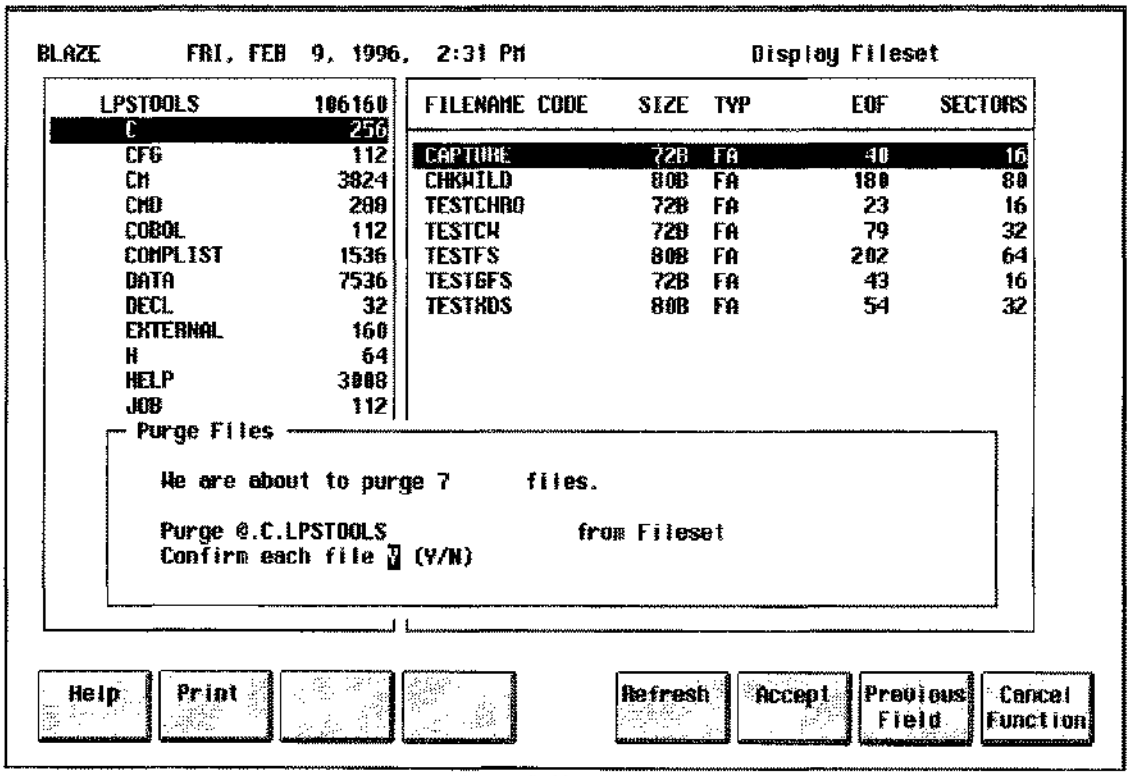


Figure 3.13 - Purge Files

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 3.14*). 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.

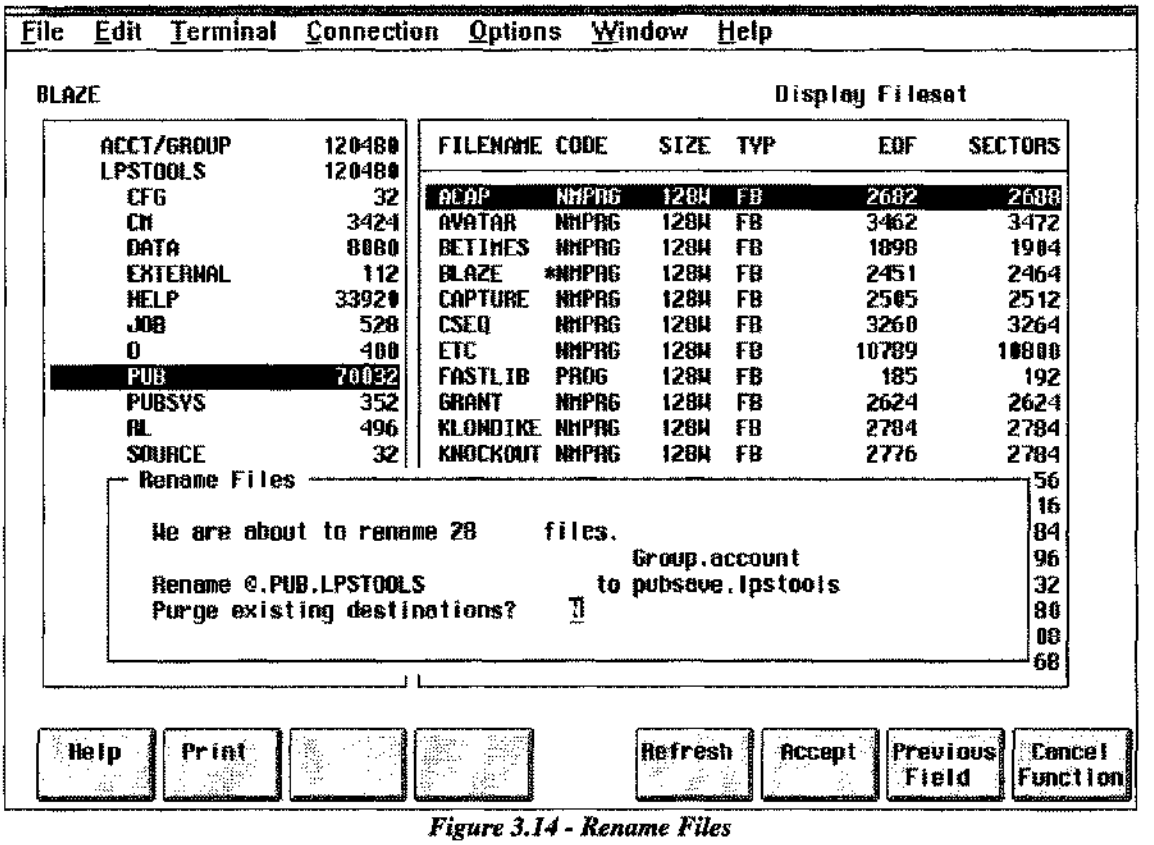


Figure 3.14 - Rename Files

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 3.15*). This window is titled "*Exec 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.

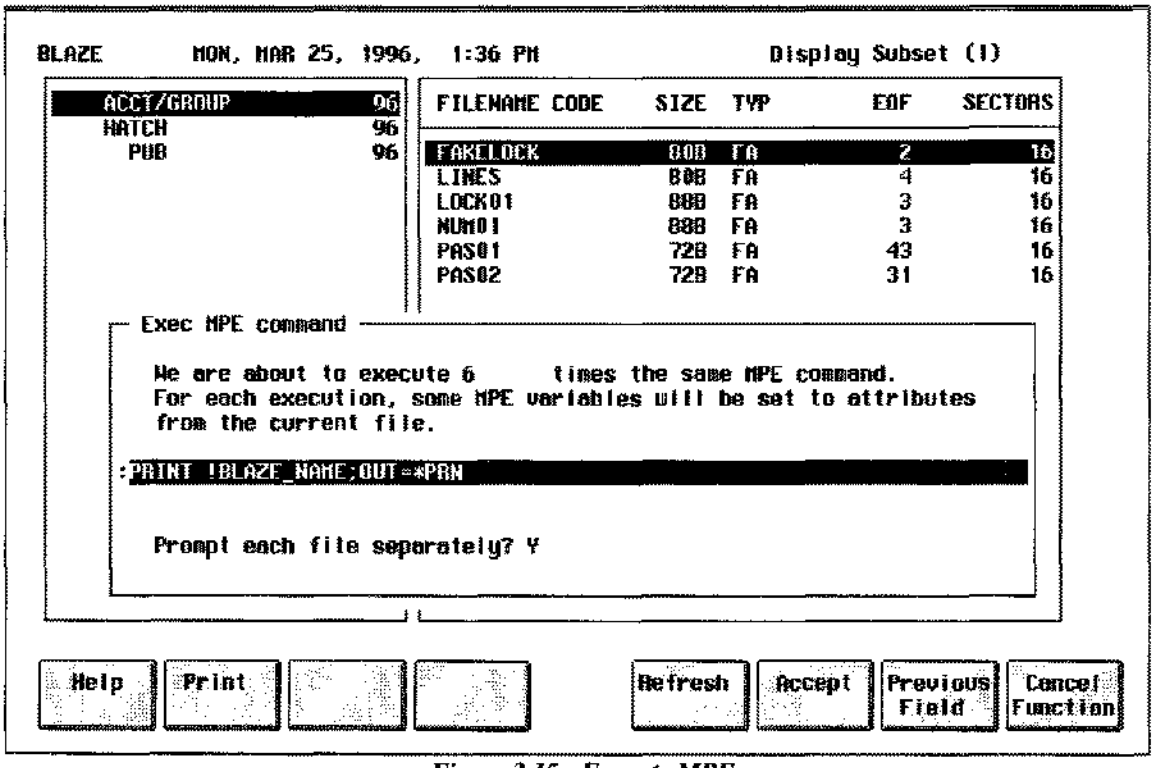


Figure 3.15 - Execute MPE

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 3.16*). 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.

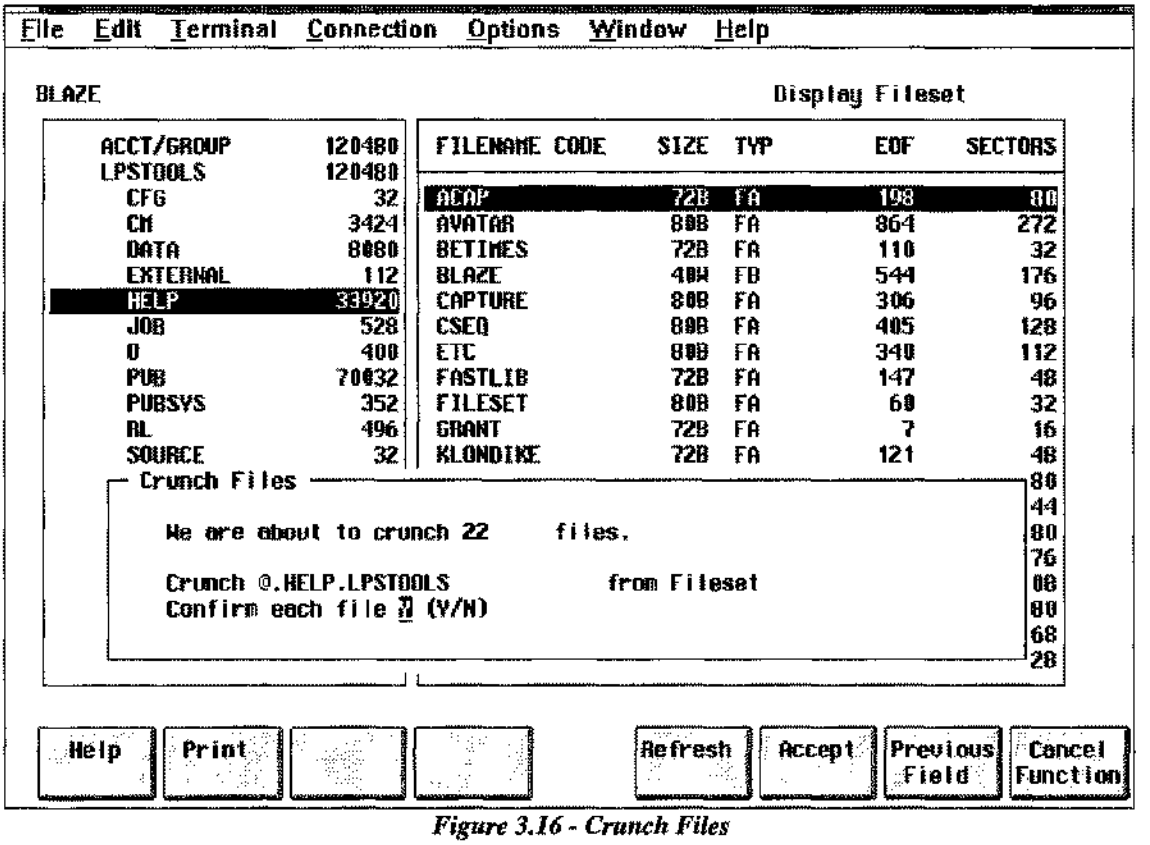


Figure 3.16 - Crunch Files

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 next** (>) 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 to pop-up on top of the current window on top of the display screen (see *Figure 3.18*). 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.

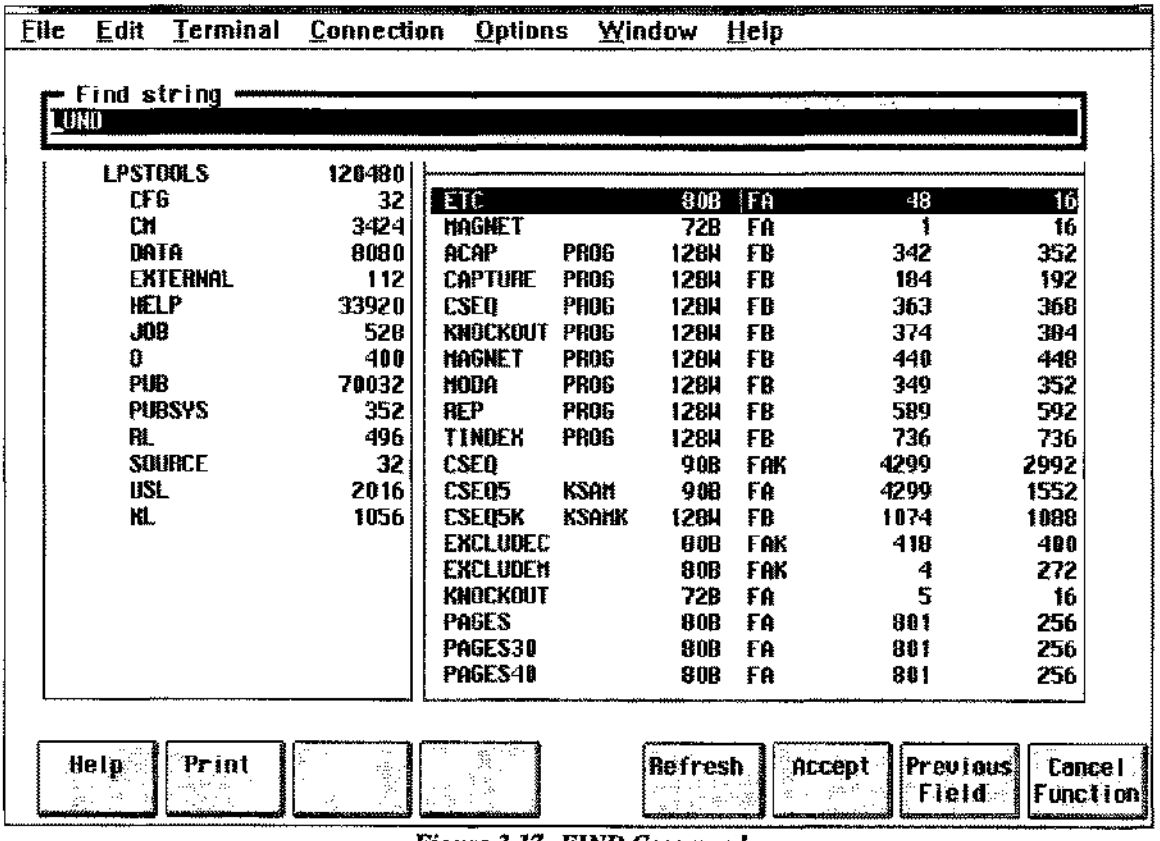


Figure 3.17- FIND Command

Help, BLAZE Single Letter Command Key Summary

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

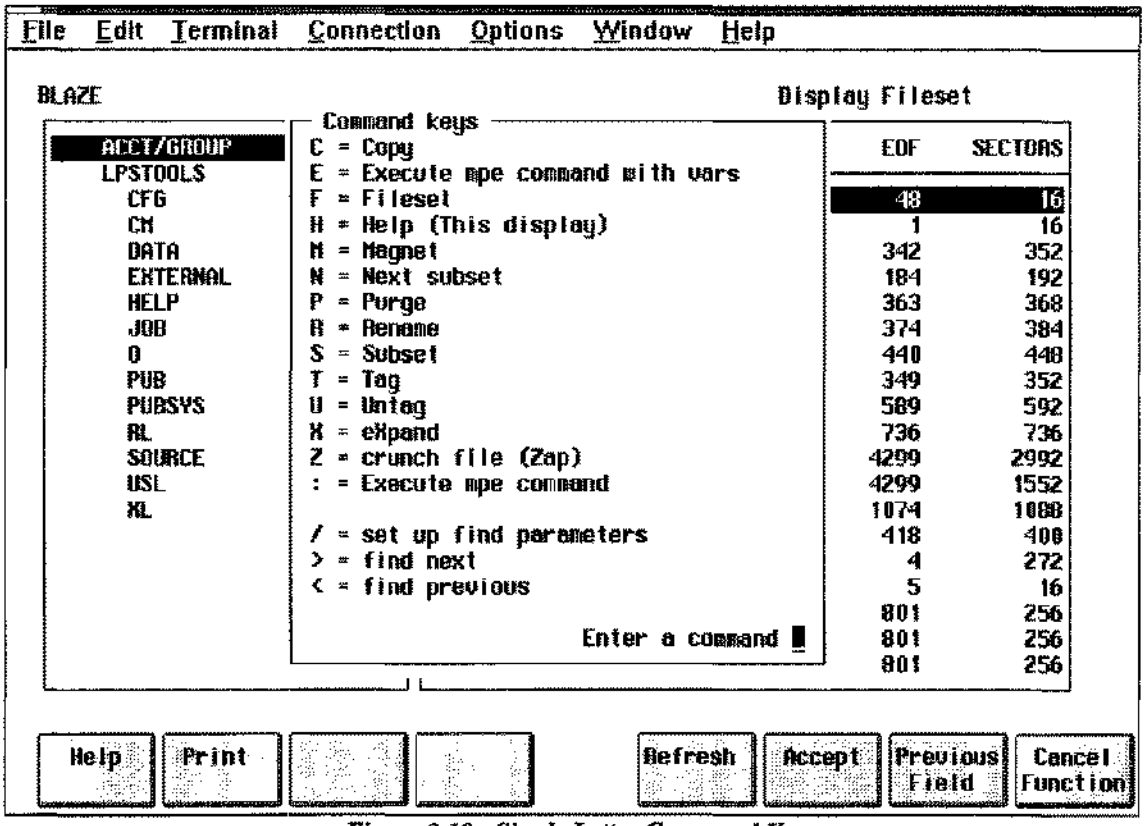


Figure 3.18 - Single Letter Command Keys

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 the 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 displays 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

Chapter 4

The CASPER Tool

CASPER is a replacement program for SPOOK, an essential spoolfile utility on the *HP3000* that became obsolete with the introduction of the Native Mode Spooler on MPE XL 2.1. CASPER provides access to Native Mode spoolfiles 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 Managers 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 JCW. 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 following table lists the various JCW settings:

JCW Settings (SPOOKFLAGS)

Meaning	Bit	Decimal Value
Extended SPOOK operation	15	1
Strict SPOOK emulation	14	2
MPEXL SPOOK emulation (pre 2.1)	13	4
MPE V SPOOK emulation	12	8
Want output paging (extended feature)	11	16

Meaning	Bit	Decimal Value
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.

***MPE XL 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.

***Not 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 Systems 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 and 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.

Command Code	Description
Alter	Alters the characteristics of specified output files
Copy	Copies all or part of a spoolfile to a file
Exit	Terminates operation
Find	Locates character string in TEXTed spoolfile
HELP	Invokes CASPER help
LALL	Synonym for LIST ALL
List	Lists all or part of a TEXTed file
LL	Lists last page of TEXTed file
Purge	Deletes one or more output spoolfiles
Quit	Terminates operation
RUN	Starts a MPE program
SET/REset	Enables or disables CASPER options
Show	Lists characteristics of input or output spoolfiles
Text	Opens an output spoolfile
Watch	Monitor spoolfile creation

Range Specification

Many of CASPER's commands accept range specifications. The following diagram shows the general structure for a range specification.

<range> =	**[*]** **[FIRST]** **[LAST]**	**offset /**	**[*]** **[FIRST]** **[LAST]**	**offset**

*	= Current line or file
offset	= Record number relative to recnumber
FIRST	= First record or line number
LAST	= Last record or line number

When a range is required for a specific command, that command's description will include the exact syntax structure.

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 <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 created 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]]]]
[;UNDEFER]]]]

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. The spoolfile's state will not change until an "ALTER ;UNDEFER" is issued.

UNDEFER This option changes the spoolfile's state to READY. The spoolfile's priority remains unchanged. This is the only way to change the state of a DEFERed spoolfile.

Copy <range> <file designator>

Copies one or more lines from the last TEXTed file to the permanent flat file given by **file designator.**

<range> = | **[*]** **[FIRST]** **[LAST]** | **offset /** | **[*]** **[FIRST]** **[LAST]** | **offset**

<file designator> Any valid MPE filename (i.e., $STDLIST, NUGLP, or a file equation of the form "*file"). For example:

copy all,myfile
copy first/first+30, *filename

Exit or "/"

The EXIT command suspends CASPER if its parent is not the top-level command interpreter, otherwise it will terminate. If you really want CASPER to terminate, use the QUIT command.

Find [@ | START] "pattern" [Up] [,<linerange>]

This command locates and displays the specified text string given by "pattern" in the last TEXTed spoolfile. *Note*: **Find** can only be used after the TEXT command has been successfully issued.

@ Scan entire line. Default: scan leading characters only (START).

START Scan leading characters only.

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

<linerange> = **[ALL]** | **[*]** **[FIRST]** **[LAST]** | **offset /** | **[*]** **[FIRST]** **[LAST]** | **offset**

If **linerange** is not given and START is not specified, then **Find** starts from the current line (*).

Examples of using the **Find** command include the following:

find @ 'LASER', 100/200
find @ "LASER", LAST-100
find @ "LASER", LAST-100/LAST-50
find @ "LASER",*/LAST
find @ "LASER",FIRST/50
find @ "LASER",*/100
find @ "LASER", ALL

HELP

This command invokes the CASPER help facility.

LALL

This command is a synonym for "LIST ALL".

List [<linerange>]["pattern" [Up]]
["pattern" [Up]]

This command lists the specified lines of the last "Texted" spoolfile. If a "pattern" is specified the lines containing the pattern within the desired range will be listed.

<linerange> =

[ALL]			
[*] **[FIRST]** **[LAST]**	**offset /**	**[*]** **[FIRST]** **[LAST]**	**offset**

linerange can optionally be qualified with a pattern to limit the number of lines displayed. If a **linerange** is not given, ALL will be used.

"pattern" [Up] Used to specify a text string to search for 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.

LL ["pattern" [Up]]

The LL command lists the last 20 lines of the current spoolfile. A pattern may be specified. LL is equivalent to "LIST LAST-20/LAST,"or "L L-20/L." 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 <spoolfilerange>

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

<spoolfilerange> **[spoolid [,spoolid] [...]]**
[spoolid [/spoolid]]
SAVEd

Spoolid is the number assigned to the spoolfile by the native mode 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 by separating them with commas, or a range of spoolids may be used.

If spoolid is negative, then 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 <progname>

Start an MPE program as a child process.

SET | REset

The SET and RESET commands are used to specify the options listed below.

SHOWNUMBERS	This option tells CASPER to display line numbers in LIST output. The default is SET SHOWNUMBERS
SHOWCCTL	This option is similar to the MODE CONTROLS=ON of SPOOK. The default is RESET SHOWCCTL
NUMBERED	Same as SHOWNUMBERS.
UNNUMBERED	Same as RESET SHOWNUMBERS, or RESET NUMBERS.
CONTINUE	When enabled, this option prevents CASPER from terminating after executing a single command that was entered via the INFO string parameter.
FINDANY	Forces the Find command to scan entire lines, by default. The Find command START option overrides this default for the duration of the session.
INTERPRETCCTL	Output control sequences.
SUSPEND	When enabled, this option suspends CASPER when an Exit command is issued, and CASPER's parent is not CI.PUB.SYS. This is handy when executing CASPER in another application, like VESOFT's MPEX.

Show [user[.acct]] [;@O] [;READY | ;OPEN]
[SAVEd]
[+] [user[.acct]]

Displays output spoolfile lists. The default (no parameters) is to show spoolfiles created by the current user. A list for all output spoolfiles may be displayed at the console.

When a Show command is issued, CASPER saves the output lines in a scratch file, so subsequent Text and Purge commands can use "relative" spoolfile numbers. Spoolfiles that are accessed through the "relative" commands are called "SAVEd spoolfiles" to distinguish them from regular MPE spoolfiles.

Negative numbers are used to identify SAVEd spoolfiles. The first SAVEd spoolfile displayed is relative number -1, the second is -2, and so on. Normally, each Show command (except Show SAVEd) resets the SAVEd spoolfile number list. However, specifying the "+" option appends Show command output to the SAVEd spoolfile number list.

user	User name for the creator of the output, or "@" for all users.
acct	Account name for the creator of the output, or "@" for all accounts.
@O	Long format; shows state, priority, number of copies, spoolfile records, and output device.
READY	Show the files in the ready state.
OPEN	Show the files in the open state.
SAVEd	Display the list of "saved" spoolfile numbers.
+	Append matching spoolfiles to SAVEd spoolfile list for a user/account.

SReady |+| [user[acct]] [;@O] [;state]

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

Text <spoolid> | < First | Last | Next | Prior >

Selects a spoolfile for use by the "Find" and "List" commands.

Spoolid is the number assigned to the spoolfile by the native mode spooler (decimal number with one or more digits. Use the **Show** command to locate spoolfiles.) The "**#O**" portion of the spoolid is not required.

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

First	Select first spoolfile from SAVEd list.
Last	Select last spoolfile from SAVEd list.
Next	Select the next spoolfile from the SAVEd list.
Prior	Select the previous spoolfile from the SAVEd list.

TN |T N

Selects the next relative spoolfile from the SAVEd spoolfile.

Watch <seconds>

After selecting a spoolfile via the text command, the user can monitor the creation of the spoolfile. Information written to the spoolfile is echoed by CASPER to the screen in 3 second increments unless otherwise specified. Once the spoolfile is built, CASPER quits monitoring.

CASPER Examples

Figure 4.1 and *Figure 4.2* 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 [1.12] - LPS Toolbox A.01a                    (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 4.1- The Contents of SPOOKHDR.DATA.LPSTOOLS

```
setjcw spookflags 16+2
:spook

SPOOK [1.12] - LPS Toolbox A.01a                    (Copyright (c) Lund)
> e
:
```

Figure 4.2 - Setting SPOOKFLAGS for STRICT MODE (2) and PAGING (16)

```
:spook

SPOOK [1.12] - LPS Toolbox A.01a                          (Copyright (c) Lund)
> s@.sys
#FILE   #JOB     FNAME     STATE   OWNER
#O266   #J'185   OFFLINE   READY   MANAGER.SYS
#O271   #J'188   OFFLINE   READY   MANAGER.SYS
#O301   #J'192   OFFLINE   READY   MANAGER.SYS
#O130   #J'64    $STDLIST  READY   FTP.SYS
#O304   #J'1     $STDLIST  READY   MANAGER.SYS
#O311   #S'7     OFFLINE   READY   MANAGER.SYS
#O314   #J'9     $STDLIST  READY   MANAGER.SYS
#O305   #J'2     $STDLIST  READY   FTP.SYS
#O317   #J'1     $STDLIST  READY   MANAGER.SYS
#O323   #J'5     OFFLINE   READY   MANAGER.SYS
#O318   #J'2     $STDLIST  READY   FTP.SYS
#O325   #J1      $STDLIST  READY   MANAGER.SYS
#O331   #J5      OFFLINE   READY   MANAGER.SYS
#O337   #J6      OFFLINE   READY   MANAGER.SYS
#O326   #J2      $STDLIST  OPEN    FTP.SYS
#O265   #J'185   $STDLIST  READY   MANAGER.SYS
#O270   #J'188   $STDLIST  READY   MANAGER.SYS
#O300   #J'192   $STDLIST  READY   MANAGER.SYS
#O302   #J'199   $STDLIST  READY   MANAGER.SYS
#O303   #J'200   $STDLIST  READY   MANAGER.SYS
#O322   #J'5     $STDLIST  READY   MANAGER.SYS
#O330   #J5      $STDLIST  READY   MANAGER.SYS
#O336   #J6      $STDLIST  READY   MANAGER.SYS
#I11    #J11     $STDIN    READY   MANAGER.SYS
#I2     #J2      $STDIN    OPEN    FTP.SYS
> s saved
-### #FILE   #JOB     FNAME     STATE   OWNER
  -1 #O266   #J'185   OFFLINE   READY   MANAGER.SYS
  -2 #O271   #J'188   OFFLINE   READY   MANAGER.SYS
  -3 #O301   #J'192   OFFLINE   READY   MANAGER.SYS
  -4 #O130   #J'64    $STDLIST  READY   FTP.SYS
  -5 #O304   #J'1     $STDLIST  READY   MANAGER.SYS
  -6 #O311   #S'7     OFFLINE   READY   MANAGER.SYS
  -7 #O314   #J'9     $STDLIST  READY   MANAGER.SYS
  -8 #O305   #J'2     $STDLIST  READY   FTP.SYS
  -9 #O317   #J'1     $STDLIST  READY   MANAGER.SYS
 -10 #O323   #J'5     OFFLINE   READY   MANAGER.SYS
 -11 #O318   #J'2     $STDLIST  READY   FTP.SYS
 -12 #O325   #J1      $STDLIST  READY   MANAGER.SYS
 -13 #O331   #J5      OFFLINE   READY   MANAGER.SYS
 -14 #O337   #J6      OFFLINE   READY   MANAGER.SYS
 -15 #O326   #J2      $STDLIST  OPEN    FTP.SYS
 -16 #O265   #J'185   $STDLIST  READY   MANAGER.SYS
 -17 #O270   #J'188   $STDLIST  READY   MANAGER.SYS
 -18 #O300   #J'192   $STDLIST  READY   MANAGER.SYS
 -19 #O302   #J'199   $STDLIST  READY   MANAGER.SYS
 -20 #O303   #J'200   $STDLIST  READY   MANAGER.SYS
 -21 #O322   #J'5     $STDLIST  READY   MANAGER.SYS
 -22 #O330   #J5      $STDLIST  READY   MANAGER.SYS
 -23 #O336   #J6      $STDLIST  READY   MANAGER.SYS
 -24 #I11    #J11     $STDIN    READY   MANAGER.SYS
 -25 #I2     #J2      $STDIN    OPEN    FTP.SYS
> text -2
[#O271]> p*
   Purged #O271
> exit
:
```

Figure 4.3 - Accessing SAVEd Spoolfile List

```
:spook

SPOOK [1.12] - LPS Toolbox A.01a                    (Copyright (c) Lund)
> s @.sys;ready
#FILE    #JOB     FNAME     STATE   OWNER
#O266    #J'185   OFFLINE   READY   MANAGER.SYS
#O301    #J'192   OFFLINE   READY   MANAGER.SYS
#O130    #J'64    $STDLIST  READY   FTP.SYS
#O304    #J'1     $STDLIST  READY   MANAGER.SYS
#O311    #S'7     OFFLINE   READY   MANAGER.SYS
#O314    #J'9     $STDLIST  READY   MANAGER.SYS
#O305    #J'2     $STDLIST  READY   FTP.SYS
#O317    #J'1     $STDLIST  READY   MANAGER.SYS
#O323    #J'5     OFFLINE   READY   MANAGER.SYS
#O318    #J'2     $STDLIST  READY   FTP.SYS
#O325    #J1      $STDLIST  READY   MANAGER.SYS
#O331    #J5      OFFLINE   READY   MANAGER.SYS
#O337    #J6      OFFLINE   READY   MANAGER.SYS
#O265    #J'185   $STDLIST  READY   MANAGER.SYS
#O270    #J'188   $STDLIST  READY   MANAGER.SYS
#O300    #J'192   $STDLIST  READY   MANAGER.SYS
#O302    #J'199   $STDLIST  READY   MANAGER.SYS
#O303    #J'200   $STDLIST  READY   MANAGER.SYS
#O322    #J'5     $STDLIST  READY   MANAGER.SYS
#O330    #J5      $STDLIST  READY   MANAGER.SYS
#O336    #J6      $STDLIST  READY   MANAGER.SYS
> text last
[#O336]> text first
[#O266]> text next
[#O301]> exit
:
```

Figure 4.4 - TEXT Command Modifiers

```
:spook

SPOOK [1.12] - LPS Toolbox A.01a                    (Copyright (c) Lund)
> s @.sys;ready
#FILE    #JOB     FNAME     STATE   OWNER
#O266    #J'185   OFFLINE   READY   MANAGER.SYS
#O301    #J'192   OFFLINE   READY   MANAGER.SYS
#O130    #J'64    $STDLIST  READY   FTP.SYS
#O304    #J'1     $STDLIST  READY   MANAGER.SYS
#O311    #S'7     OFFLINE   READY   MANAGER.SYS
#O314    #J'9     $STDLIST  READY   MANAGER.SYS
#O305    #J'2     $STDLIST  READY   FTP.SYS
#O317    #J'1     $STDLIST  READY   MANAGER.SYS
#O323    #J'5     OFFLINE   READY   MANAGER.SYS
#O318    #J'2     $STDLIST  READY   FTP.SYS
#O325    #J1      $STDLIST  READY   MANAGER.SYS
#O331    #J5      OFFLINE   READY   MANAGER.SYS
#O337    #J6      OFFLINE   READY   MANAGER.SYS
#O265    #J'185   $STDLIST  READY   MANAGER.SYS
#O270    #J'188   $STDLIST  READY   MANAGER.SYS
#O300    #J'192   $STDLIST  READY   MANAGER.SYS
#O302    #J'199   $STDLIST  READY   MANAGER.SYS
#O303    #J'200   $STDLIST  READY   MANAGER.SYS
#O322    #J'5     $STDLIST  READY   MANAGER.SYS
#O330    #J5      $STDLIST  READY   MANAGER.SYS
#O336    #J6      $STDLIST  READY   MANAGER.SYS
```

```
> s @.sys;@o
#FILE    #JOB     FNAME     STATE   DEV/CL    PR COP RFN OWNER
#O266    #J'185   OFFLINE   READY   dev/cl     8   1      MANAGER.SYS
#O301    #J'192   OFFLINE   READY   dev/cl     8   1      MANAGER.SYS
#O130    #J'64    $STDLIST  READY   dev/cl     8   1      FTP.SYS
#O304    #J'1     $STDLIST  READY   dev/cl     8   1      MANAGER.SYS
#O311    #S'7     OFFLINE   READY   dev/cl     8   1      MANAGER.SYS
#O314    #J'9     $STDLIST  READY   dev/cl     8   1      MANAGER.SYS
#O305    #J'2     $STDLIST  READY   dev/cl     8   1      FTP.SYS
#O317    #J'1     $STDLIST  READY   dev/cl     8   1      MANAGER.SYS
#O323    #J'5     OFFLINE   READY   dev/cl     8   1      MANAGER.SYS
#O318    #J'2     $STDLIST  READY   dev/cl     8   1      FTP.SYS
#O325    #J1      $STDLIST  READY   dev/cl     8   1      MANAGER.SYS
#O331    #J5      OFFLINE   READY   dev/cl     8   1      MANAGER.SYS
#O337    #J6      OFFLINE   READY   dev/cl     8   1      MANAGER.SYS
#O326    #J2      $STDLIST  OPEN    dev/cl     8   1      FTP.SYS
#O265    #J'185   $STDLIST  READY   dev/cl     1   1      MANAGER.SYS
#O270    #J'188   $STDLIST  READY   dev/cl     1   1      MANAGER.SYS
#O300    #J'192   $STDLIST  READY   dev/cl     1   1      MANAGER.SYS
#O302    #J'199   $STDLIST  READY   dev/cl     1   1      MANAGER.SYS
#O303    #J'200   $STDLIST  READY   dev/cl     1   1      MANAGER.SYS
#O322    #J'5     $STDLIST  READY   dev/cl     1   1      MANAGER.SYS
#O330    #J5      $STDLIST  READY   dev/cl     1   1      MANAGER.SYS
#O336    #J6      $STDLIST  READY   dev/cl     1   1      MANAGER.SYS
#FILE    #LDEV      LABEL  ADDR          SECTORS        LINES       TIME
#O266    %2     $00000001  $0037dd00     5312          10057        8:28 12/ 6/95
#O301    %1     $00000001  $00453900     5328          10083        8:51 12/ 8/95
#O130    %1     $00000001  $0003f300       16             78        9:17 12/11/95
#O304    %1     $00000001  $00015900       16             51        9:18 12/11/95
#O311    %2     $00000001  $000a4700      544           1442       11: 0 12/11/95
#O314    %1     $00000001  $004b5d00      384            743       11:37 12/11/95
#O305    %2     $00000001  $0005ee00       16             81       14:41 12/11/95
#O317    %1     $00000001  $000a8000       16             51       14:42 12/11/95
#O323    %1     $00000001  $0004dd00     6160          11881        8:50 12/12/95
#O318    %1     $00000001  $001db800       16             85       12: 8 12/12/95
#O325    %1     $00000001  $004f8300       16             51        9:56 12/13/95
#O331    %1     $00000001  $0004c800       16             39        8: 7 12/14/95
#O337    %2     $00000001  $0037b300       16             41        8:45 12/15/95
#O326    %1     $00000001  $00046b00      256              0
#O265    %2     $00000001  $00037500     4512          10236        8:49 12/ 6/95
#O270    %2     $00000001  $00393000     4464          10116        8:28 12/ 7/95
#O300    %2     $00000001  $00387900       32            183        8:51 12/ 8/95
#O302    %1     $00000001  $001cc200       16             57        3: 1 12/ 9/95
#O303    %1     $00000001  $000b4000       16             57        3: 1 12/10/95
#O322    %1     $00000001  $001cf200       32            186        8:51 12/12/95
#O330    %2     $00000001  $000f0f00       32            168        8: 8 12/14/95
#O336    %2     $00000001  $000a8000       32            168        8:45 12/15/95
> p -i
   Purged #O266
> exit
:
```

Figure 4.5 - Long Output (@O) Format

CASPER Error Messages

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

Message	***Alter what?***
Cause	User input needs Alter option.
Action	Valid Alter options are: PRI, DEFER, COPIES, DEV, UNDEFER
Message	***Both numbers in range must be relative or both must be positive numbers.***
Cause	Mixing relative spoolfile numbers and real spoolfile numbers is not allowed.
Action	Do not mix range types when specifying spoolfile numbers.
Message	***Didn't find semicolon separating #O numbers from options.***
Cause	Bad user input for Alter command.
Action	Review Alter command syntax.
Message	***Failed to alter #Onnnn***
Cause	CASPER's "ALTSPOOLFILE" command failed.
Action	ALTSPOOLFILE must be ALLOWed for user running CASPER. Review ALLOW command in MPE/iX Command Manual
Message	***Invalid line range***
Cause	Generally: not standard range given.
Action	Review range syntax for command in question.
Message	***No prior spoolfile number remembered.***
Cause	Relative spoolfile list may be empty.
Action	Use Show SAVEd command to review relative spoolfile list.
Message	***No spoolfile ids remembered.***
Cause	Relative spoolfile list may be empty.
Action	Use Show SAVEd command to review relative spoolfile list.
Message	***Only ## spoolfile ids in saved list.***
Cause	Relative spoolfile number too large.
Action	Use Show SAVEd command to review relative spoolfile list.
Message	***Relative spoolfile number ## no longer valid.***
Cause	Entered relative spoolfile number not in list.
Action	Use Show SAVEd command to review relative spoolfile list.
Message	***Unknown option.***
Cause	Bad option for Alter command.
Action	Review Alter command syntax.
Message	***Unknown SET/RESET option: <option>, ignored***
Cause	An invalid RE[SET] option was used
Action	Review the valid RE[SET] options for your selection.

Chapter 5

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 associated 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 updated 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.

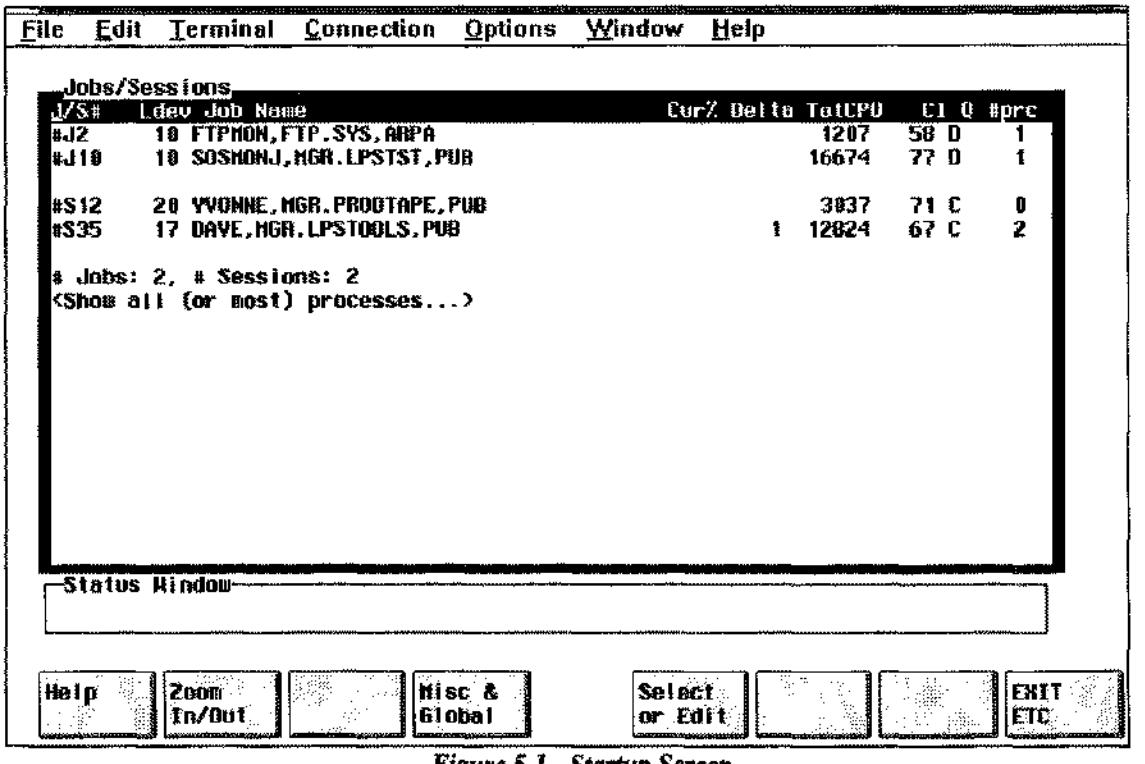


Figure 5.1 - Startup Screen

Figure 5.1 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 then 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.

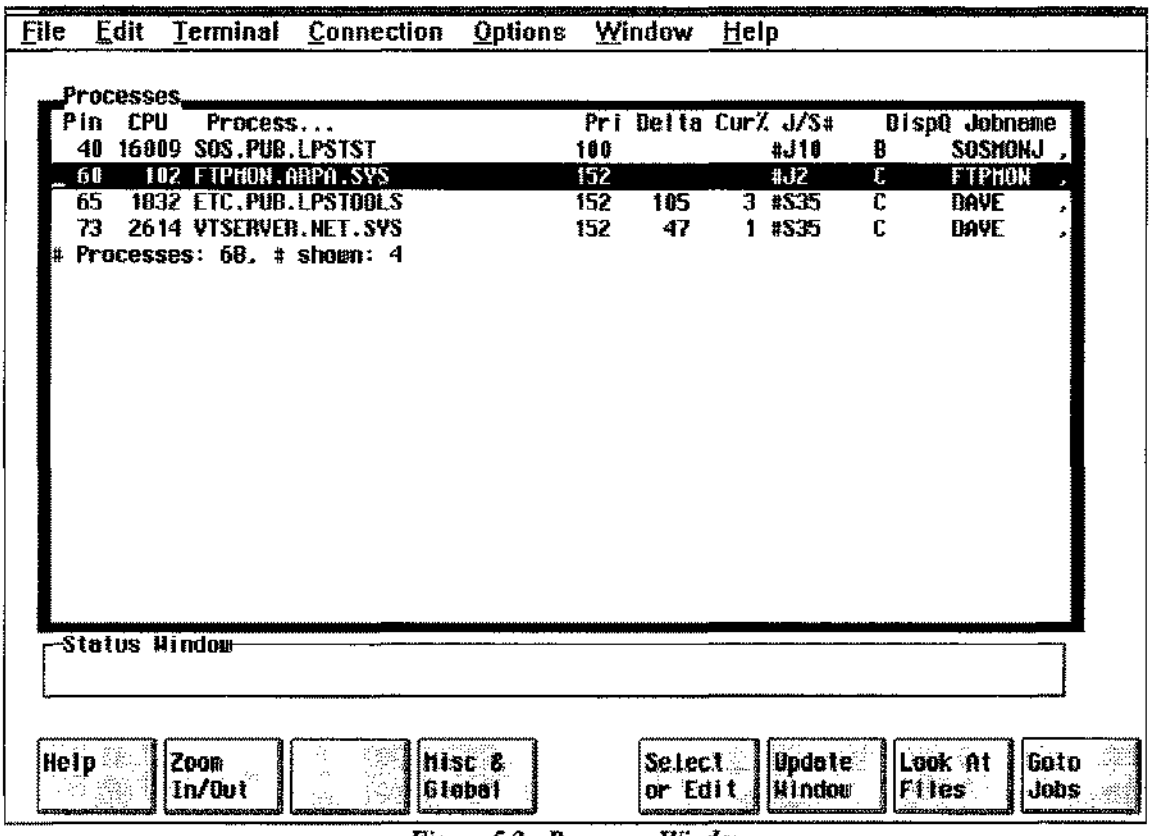


Figure 5.2 - Processes Window

Figure 5.2 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.

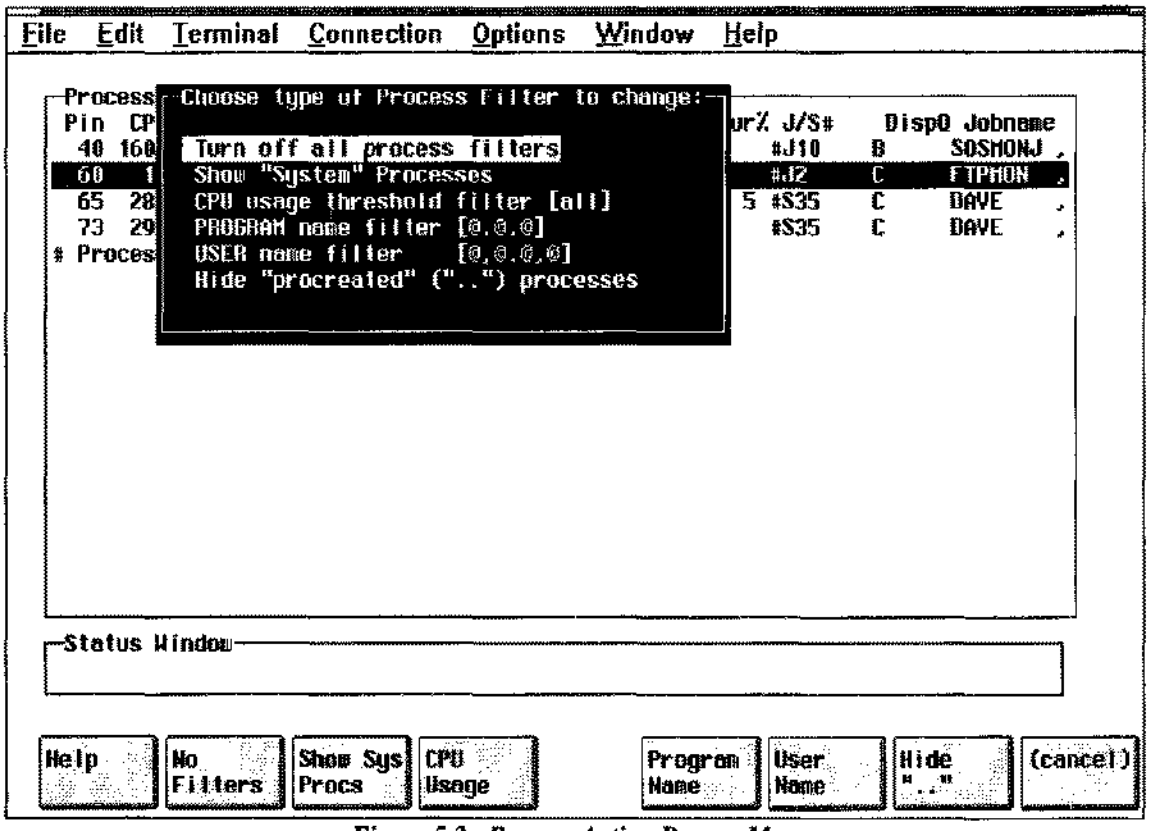


Figure 5.3 - Process Action Pop-up Menu

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.

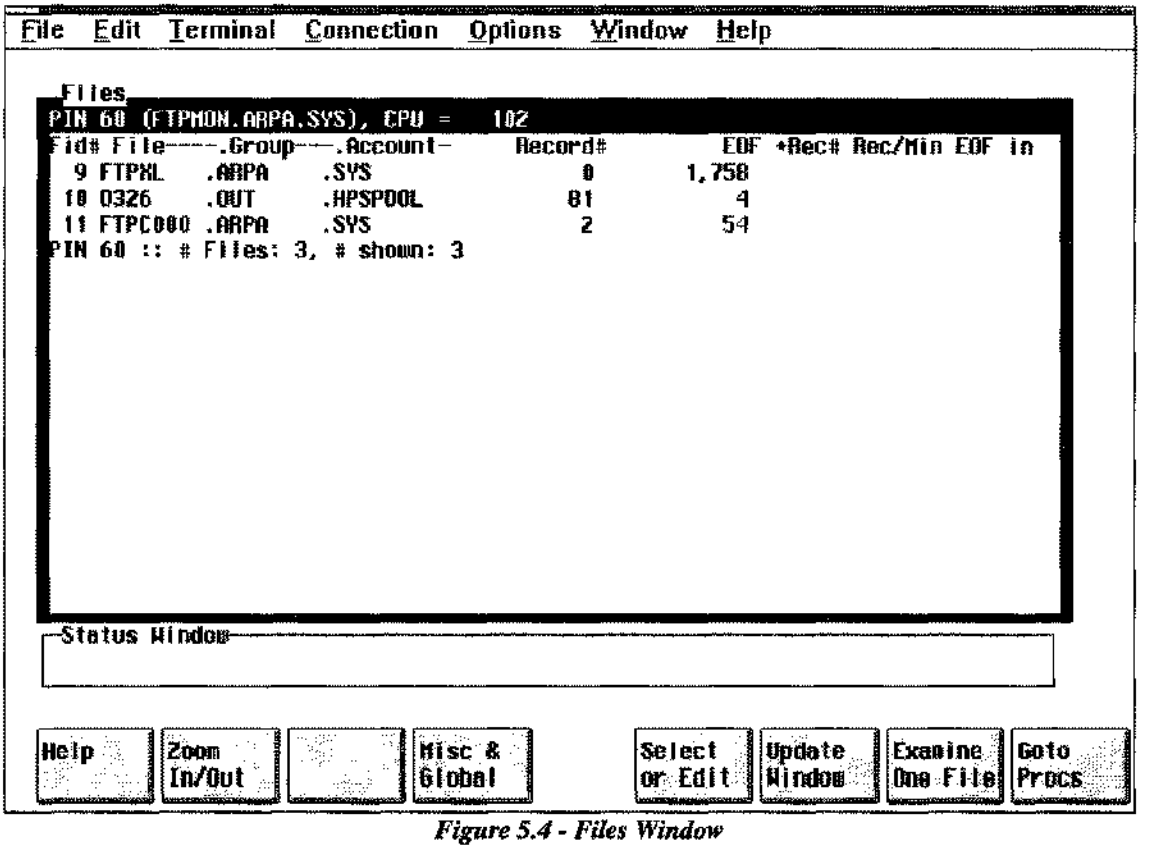


Figure 5.4 - Files Window

Figure 5.4 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.

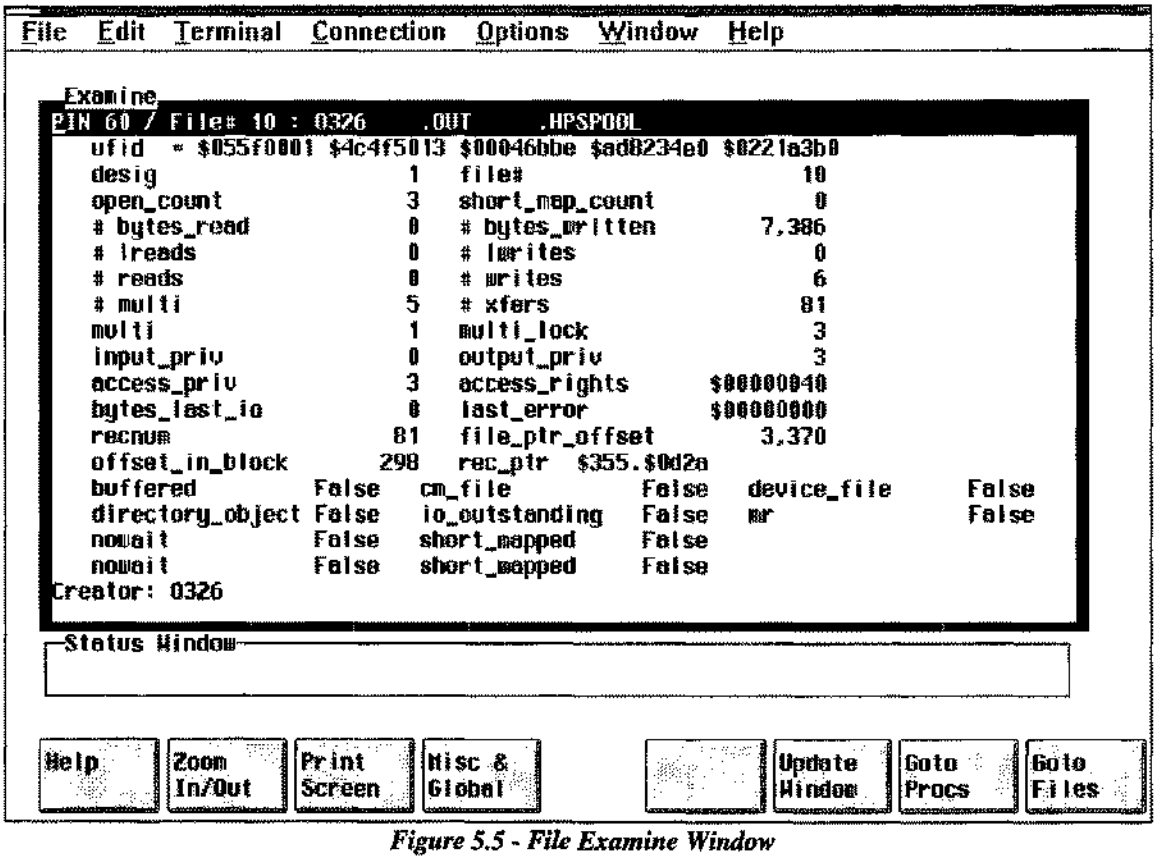


Figure 5.5 - File Examine Window

Figure 5.5 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 5.5* 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.

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 Jobs/Session window.
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.

Function Key	Option/Window*	Description
Select or Edit	Jobs/Sessions	Displays a Jobs Action window where jobs-related information may be specified for the information view. Note that 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 that 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.

Chapter 6

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 given 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.

```
:grant
LOCKWORD: GRANT.PUB.LPSTOOLS?


GRANT [2.0] - LPS Toolbox [A.01a]          (c) 1995 Lund Performance Solutions

For Help, :RUN GRANT.PUB.LPSTOOLS,HELP

granted.

END OF PROGRAM
:
```

Figure 6.1 - Running GRANT

GRANT Error Messages

Message	***GRANT cannot run on this version of MPE, PCB size=xxx.***
Cause	Because GRANT works directly with MPE data-structures, there are checks built into it to prevent incorrect operation due to operating system changes.
Action	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.

Chapter 7

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, then 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 chapter in the ***System Managers Toolbox*** section 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.

Command Code	Description
COUNT	Displays percentage of file in memory
Exit	Terminates the program
FETCH	Loads file into memory
FREEZE	Freezes file into memory
GUFD	Displays the Global Unique File Description
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)
UFID	Displays the Unique File ID
UNFREEZE	Synonym for Thaw

Command Definitions

Listed below is a detailed description of each KLONDIKE command. Syntax is provided in some cases.

COUNT <filename>

Counts the number of pages of a file that are currently in memory and then displays the results on the screen. 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 <filename> [# <Pages | Bytes>]
[<WAIT | NOWAIT> | All]

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.

The WAIT option tells KLONDIKE to wait until all of the fetched pages are in memory before continuing.

The NOWAIT option (default) tells KLONDIKE to not 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 <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 issued.

GUFD <filename>

This command displays the virtual address of the Global Unique File Description in hexadecimal format. This is a 4-byte address.

OPEN <filename>

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

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. The WAIT option (default) tells KLONDIKE to wait until all of the posted pages have been written to disk before continuing. The NOWAIT option tells KLONDIKE to not wait for all the disk writes to finish. Specifying MAKEROC marks a file's pages as ROC.

SET | REset

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

PARTial When set, this option allows partial file operations. The default is RESET PARTial.

TIMES When set, this option causes KLONDIKE to report on CPU usage after each command KLONDIKE performs. It is not affected by the QUIET command.

QUIET When set, this option is used to suppress most normal output from KLONDIKE. This may be desirable when using KLONDIKE in a command file, job, or UDC.

VERBOSE When set, this option causes most of KLONDIKE's commands to display information on: Frozen Count, Coming In Count, Going Out Count, Dirty Page Count, References Page Count, ROC Count. For example: SET VERBOSE

THAW [<filename>] | 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.

UFID <filename>

This command displays the virtual address of the Unique File ID in hexadecimal format. This is a 20-byte address.

KLONDIKE Examples

Following are examples of the KLONDIKE tool:

Figure 7.1 shows how the COUNT command is used and what the output looks like:

```
:klondike
KLONDIKE [2.0] - LPS Toolbox [A.01a]        (c) 1995 Lund Performance Solutions

For Help at the KLONDIKE prompt enter   ?

KLONDIKE: count qedit.pub.robelle

qedit.pub.robelle @ $21.0 ... opened ok.
File: 681 pages; InMem: 369 (54% of file); 181 Referenced

KLONDIKE: exit
:
```

Figure 7.1 - COUNT Command

Figure 7.2 shows the effect of the **Verbose** and **Times** options when used with FETCH:

```
:klondike
KLONDIKE [2.0] - LPS Toolbox [A.01a]        (c) 1995 Lund Performance Solutions

For Help at the KLONDIKE prompt enter   ?

KLONDIKE: fetch catalog.pub.sys

catalog.pub.sys @ $179.100 ... opened ok.
forcing WAIT option
.
File: 274 pages; InMem: 274 (100% of file); 49 Referenced

KLONDIKE: set verbose
ok

KLONDIKE: set times
ok
CPU = 3, elapsed = 4 milliseconds.

KLONDIKE: fetch catalog.pub.sys

catalog.pub.sys @ $179.100 ... opened ok.
forcing WAIT option
.
File: 274 pages; InMem: 274 (100% of file); 0 Frozen; 0 Coming In;
   0 Going Out; 0 Dirty; 274 Referenced; 0 ROC; 0 Resident
CPU = 145, elapsed = 150 milliseconds.

KLONDIKE: exit
:
```

Figure 7.2 - Verbose and Times Options

Figure 7.3 shows how to freeze a source file into memory:

```
:klondike
KLONDIKE [2.0] - LPS Toolbox [A.01a]        (c) 1995 Lund Performance Solutions

For Help at the KLONDIKE prompt enter   ?

KLONDIKE: freeze myfile.data

myfile.data @ $471.100 ... opened ok.
.
File: 274 pages; InMem: 274 (100% of file); 274 Frozen; 274 Referenced

KLONDIKE: exit
:
```

Figure 7.3 - Freezing a Source File into Memory

Figure 7.4 uses the COUNT 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:

```
:klondike
KLONDIKE [2.0] - LPS Toolbox [A.01a]        (c) 1995 Lund Performance Solutions

For Help at the KLONDIKE prompt enter   ?

KLONDIKE: count myfile.data

myfile.data @ $471.100 ... opened ok.
File: 274 pages; InMem: 274 (100% of file); 274 Frozen; 274 Referenced

KLONDIKE: thaw myfile.data

myfile.data @ $471.100 ... opened ok.
.
File: 274 pages; InMem: 274 (100% of file); 274 Referenced

KLONDIKE: exit
:
```

Figure 7.4 - 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.

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

Chapter 8

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 CM 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, then 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 finds an 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 jobname 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 determined 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.

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

END prevents further commands from being read. End-of-file is interpreted as an END.

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.

Exit

Exit terminates KNOCKOUT.

HELP

HELP invokes KNOCKOUT Help.

IDLE jobname, user.account = seconds
[WARN | NOWARN]

Up to 40 IDLE commands may be issued in the KNOCKOUT job. **Jobname**, **user**, and **account** specifications may be literal or MPE wildcard based (e.g.: **@, 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. For example:

IDLE @, FIELD.SUPPORT = 10 NOWARN
IDLE MARY, @.ACCTNG = 0

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.

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 and 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.

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 8.1 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 8.1 - Script Example

Figure 8.2 works just like the prior example, but users are not warned. Instead, they are simply aborted. When JOHN,MANAGER.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 8.2 - Script Example

In *Figure 8.3*, 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 8.3 - Script Example

Figure 8.4 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. FRI, DEC 15, 1995,  2:55 PM.


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

Figure 8.4 - KNOCKOUT Warning Message

Figure 8.5 is a KNOCKOUT Job Stream:

```
:JOB KNOCKOUT,MANAGER.SYS,PUB.
PRIORITY = DS; INPRI = 8; TIME = UNLIMITED SECONDS.
JOB NUMBER = #J245.
MON, DEC 17, 1996, 8:36 AM.
HP3000  RELEASE: C.50.02   USER VERSION:  C.50.02
MPE/iX HP31900 B.79.06  Copyright Hewlett-Packard 1987.

:Run knockout.pub.lpstools

KNOCKOUT [1.1] - LPSTOOLS [H.28.14]                          XYZ Company [A0010]
(c) 1995 Lund Performance Solutions                          Albany, Oregon

For Help at the KNOCKOUT prompt enter   ?

       exclude ldevs console <<<--User or Job input
Input = EXCLUDE LDEVS CONSOLE <<<--KOCKOUT's response
       loop 61
Input = LOOP 61
       set warn
Input = SET WARN
       set global 0
Input = SET GLOBAL 0
       set logabort
Input = SET LOGABORT
       set logwarn
Input = SET LOGWARN
       idle joe,@.@=15
Input = IDLE JOE,@.@=15
       report

-----------------------------------------------
KOCKOUT controls:
       IDLE patterns:
             JOE,@.@ = 15 WARN ;
       SET GLOBAL 0 seconds (disabled)
       SET WARN    LOGWARN    LOGABORT
       LOOP 61
       EXCLUDE LDEVs CONSOLE
-----------------------------------------------
  end
```

Figure 8.5 - KNOCKOUT Job Stream

KNOCKOUT Error Messages

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

Message	***Bad acctname pattern.***
Cause	User entered a pattern for a acctname which KNOCKOUT does not understand.
Action	Review acctname pattern, it should follow conventions defined by Hewlett-Packard's JOB command.
Message	***Bad jobname pattern.***
Cause	User entered a pattern for a jobname which KNOCKOUT does not understand.
Action	Review jobname pattern, it should follow conventions defined by Hewlett-Packard's JOB command.
Message	***Bad username pattern.***
Cause	User entered a pattern for a username which KNOCKOUT does not understand.
Action	Review username pattern, it should follow conventions defined by Hewlett-Packard's JOB command.
Message	***The maximum number of IDLE patterns "xxxx" has already been defined.***
Cause	User has defined more IDLE patterns than KNOCKOUT supports.
Action	Rethink, consolidate IDLE patterns into fewer IDLE commands.

Chapter 9

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 hyphen (-) 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:

```
/LPSTOOLS/PUB: magnet "-f@.c -c 'main' 'define'"
MAGNET [2.5] - LPS Toolbox [A.01c]      (c) 1995 Lund Performance Solutions

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

Scanned 7 files in 0.895 seconds
```

Figure 9.1 - MAGNET Output

If a "-l" option is specified, then the entire matching line is displayed as well as the name of the file 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>' [<TEXTSTRING>]
```

Figure 9.2 - Text String Definition

MAGNET 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'"

```
<fileset>
       = <file set descriptor>
         [ [ <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. Or, an indirect file.}

<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>
          |    "CCTL"    <onoroff>
          |    "RIO"     <onoroff>
          |    "MSG"     <onoroff>
          |    "CIR"     <onoroff>
          |    "REC"     <relop> <numeric value>
          |    "TEMP"
          |    "ASCII"
          |    "BINARY"
          |    "FIXED"
          |    "VARIABLE"
          |    "UNDEFINED"
<onoroff>
          = "=" {"ON" | "OFF" }

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

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

Figure 9.3 - MAGNET Extended Fileset Syntax

Note: All literals are case-insensitive.

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 (e.g., **-noascii**). Complete descriptions are provided in the next section.

Option Name	Description
-a	All words must be present
-aligned	Aligns file-group-account names into columns
-ascii7	Allow for 7 or 8-bit characters
-b	Keep output as permanent file
-binary	Use for binary file specifications
-c	Case-insensitive search
-cctl	Use carriage control when the -p option is used
-d	Use for binary file specifications
-dashes	Print line of dashes between matches
-e	Editor/3000 line number format
-f fileset	Fileset descriptor
-g	Prefetch file via KLONDIKE
-h	Print page header
-help	Displays the help file
-l	Print line that contains match
-lockword	Specifies lockword for opening files
-m	Output fileset in LISTF,6 format
-maxlines #	Limits search output for -L option
-maxrecords #	Specify max records to print when matched
-msg	Read MSG files
-msgcopy	Use copy mode when opening a MSG file
-msgwait	Wait after reading an empty MSG file
-n	Print line numbers
-o	Output sent to this file
-olddates	Preserve access dates
-origin	Displays the origin of the current option string
-p device	Alternate output device
-paging	Pages interactive output on the screen
parm=1	Suppress paging
parm=2	Displays input method for search specification
-pascal	Shorthand for -w -s_ option combination
-prefetch	Synonym for -g
-printable	Change non-printable characters into dots
-q	Quiet output-related messages
-qedit	Scans QEDIT files
-s specials	Special characters
-spl	Shorthand for -w -s' option combination
-splash	Shorthand for -w -s_' option combination
-t	Search text
-tellop	Sends file matching messages to the console
-timestamp	Prints output request date and time information
-u	Don't enhance matching values
-w	Whole words
-y	Summarizes search results
-[	Use for matches that must start in the first column

Option Name	Description
-(	Left margin of files being searched
-)	Right margin of files being searched
-72	Limits searches to the first 72-bytes in a record

Note: Single letter options are not case-sensitive

Options Definitions

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

-a

This option is used to tell MAGNET that all of the strings being searched for must be present to be considered a match. For example, **:magnet "-.f@.@ -a 'first' 'second'"** means that both of the words "first" and "second" must be present to be considered a match.

-aligned

Tells MAGNET to align file-group-account names into three columns of eight characters each. The format is as follows, where each dash (-) represents a space:

file----.group---.account-

The default is **-NOaligned**.

-ascii7

This option specifies that any ASCII characters outside of the range 32 through 126 should be displayed as dots (.). The **-NOascii** option will allow all characters in the range 32 through 255 to be displayed without change. The default is **-NOascii7**.

-b

Will keep the output fileset as a permanent file rather than as a temporary file. This command only affects the file given in the **-o** option. For example, **:magnet "-f@ -b -o myfile 'first'"** would cause "myfile" to be stored as a permanent file. It would also list all files containing the string "first."

-binary

This specifies that all files must be searched. MAGNET normally defaults to searching only ASCII and QEDIT files. *Note*: This option is equivalent to **-d**. If you want to target QEDIT files specifically, refer to the **-qedit** option. The default is **-NOd** (i.e., ASCII and QEDIT only).

-c

Tells MAGNET to ignore the case of alphabetic characters while searching. The default is case-sensitive searching. For example, **:magnet "-f@ -c copyright"** would include files containing "copyright," "Copyright," or "COPYRIGHT."

-cctl

This option tells MAGNET to use carriage control when writing to **"-p (device)"** (by default, usually a printer or the STDLIST). The default is **-cctl** if you are using the **-p** option. Otherwise, MAGNET assumes **-NOcctl**.

-d

Forces MAGNET to search through binary files as well as ASCII. This is useful for locating text within a program file. For example, **:magnet "-f@,code=nmprg -d -c 'copyright'"** would search only those files with a filecode of nmprg, and include only those files containing the string "copyright," regardless of case.

-dashes

This tells MAGNET to print a line of dashes after each matched file. This option is particularly nice when used with a **-l** (list) option.

-e

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

For example, **:magnet "-f@.source,credate>960201 -c -e -l 'select'"** would search all files in the source group that were created after "960201" for the string "select" in any case (upper, lower, or mixed) and list the output with Editor/3000 compatible line numbers (the actual line numbers are listed on the far left preceded by a pound (#) sign).

-f fileset

Specifies a set of files to search. The fileset syntax is a superset of the fileset syntax used by the :LISTF command. See the *Figure 9.3* entitled "*MAGNET Extended Fileset Syntax*" for a complete illustration.

Multiple **-f** options are allowed. The default is "**-f@.**" For example,

:magnet "-f@.source-@q.include-bnf.source -c select"

specifies a fileset that contains all files in the SOURCE group except BNF.SOURCE or any file ending in "Q." Then, this fileset is searched for the string, "select," regardless of case. Another example would be,

:magnet "-f@.@,(eof<limit) -c 'file'"

which specifies all files whose EOF is less than their LIMIT that contain the string, "file," regardless of case.

-g

This option tells MAGNET to programmatically use the KLONDIKE tool in ***System Managers Toolbox*** to prefetch files into memory before searching through them. This can save significant CPU time when searching through large files (see the KLONDIKE chapter for a discussion of the FETCH command).

For example, **:magnet "-f@ -g -c copyright"** would load all files in the logon group into memory before searching them for the string, "copyright," regardless of case.

-h

This tells MAGNET to print a page header on each page. The default is to not print a page header.

For example, **:magnet "-f@ -c -h -l select"** will list all lines with the string, "select," in any case from the logon group and print a header of "s e l e c t" (vertically spaced) on each page.

-help

This option is used to print the entire help file to the screen.

-l (the letter "L")

Tells MAGNET to list all occurrences of the search string. The default is to not list lines. Instead, files that contain one or more occurrences of the search string(s) are flagged with an asterisk.

Note: If more than one string is to be searched and the **-l** option is in effect, searching is considerably slower.

For example, **:magnet "-f@.SOURCE -l 'Lund Performance Solutions"** searches all source files for the string, "Lund Performance Solutions," and lists the lines that contain this string.

-lockword

This option specifies the lockword for MAGNET to use when opening files to search. MAGNET will not open any file with a different lockword, but will open each file without a lockword.

If **-NOlockword** is set, and a file has a lockword, then MPE will prompt the user for the lockword before it will open the file. If **-neverlockword** is set, then MAGNET will ignore any subsequent **-[NO]lockword** command and open each file.

For MPE/iX versions 5.0 and newer, the default is **-lockword fakelock**. For MPE/iX versions 4.5 and older, the default is **-NOlockword**.

-m

This tells MAGNET to store the fileset in the LISTF,6 format rather than MAGNET's own internal format (which closely resembles a LISTF,2). This command only affects the output that is stored in the file given by the **-o** option.

For example, **:magnet "-f@ -m -b -omyfile 'first'"** would cause "myfile" to be stored as a permanent file, in the LISTF,6 format, and would include all files found with the word, "first."

-maxlines #

This tells MAGNET to stop scanning a file after "#" matches are found. This option is only meaningful when used in conjunction with the **-l** option.

A value of 0 (or **-NOmaxlines**) tells MAGNET to conduct an unlimited search. Specifying a value of "10" tells MAGNET to stop searching once ten matches have been found. The default is **-NOmaxlines**.

-maxrecords #

This option tells MAGNET to stop printing matching lines after "#" lines are printed. This option is only meaningful when used in conjunction with the **-l** option.

-msg

Tells MAGNET to read MSG files. These files are normally ignored. However, if **-msgcopy** is also set (which it is by default) then the MSG files will be read in the **copy** mode and will not disappear. The default is **-NOmsg.**

Note: Prior to the MPE/iX 5.0 version, records do not necessarily appear in chronological order when reading a MSG file in the copy mode.

-msgcopy

This tells MAGNET to use the **copy** mode when opening a MSG file. If **-msg** and **-NOmsgcopy** are set, then the records that are read from a message file will disappear.

-msgwait

This tells MAGNET to wait after it has opened an empty MSG file. The **-msgwait** command only has an effect if **-msg** and **-NOmsgcopy** are also set.

The default for **-msgwait** is **-NOmsgwait**.

-n

This option tells MAGNET to print line numbers for each listed line. The default is no line numbers are shown.

For example, **:magnet "-f@ -l -n version"** would print the lines and line numbers for all records containing the word, "version," in lowercase.

-never lockword

This makes MAGNET ignore any subsequent **-[NO]lockword** commands.

-never olddates

This makes MAGNET ignore any subsequent **-[NO]olddates** commands.

-never prefetch

This makes MAGNET ignore any subsequent **-g** (**-prefetch**) commands.

-o <filename>

This option is used to have MAGNET create a file that contains the resultant fileset (as defined by the user **-f** option). The format for the information in the file can either be LISTF,6 (**-m** option) or MAGNET's own internal format. If the latter is used, then it can be used as input to a subsequent MAGNET search via the **-f** option (**-fmyfile**). *Note*: See the example for **-m**. You may also wish to review portions of the WILDCARD documentation which describes the internal storage format of filesets.

For example, **:magnet "-f@ -c -o test1 'first'"** would build a temporary file called "**test1**," which would contain all files found. To permanently save the file, use the **save** command or MAGNET's **-b** option.

-olddates

This tells MAGNET to try to restore the old access date for any file it touches. When **-NOolddates** is set, MAGNET will not try to reset the access date. If **-neverolddate** is set, MAGNET will ignore any subsequent **-[NO]olddates.**

-origin

This tells MAGNET to display the origin of the current (and subsequent) option string(s) being analyzed, and to display a copy of the string, as well. The default is **-NOorigin.**

-p device

This option tells MAGNET the name of the device (e.g., "LP" or "113") on which the output is to be printed. When the **-p** option is given without a device, then LP is used. The default is to print output 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 (e.g., 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

This option is used to paginate interactive output. See also PARM=1 for an alternate method to paginate output. The default is "**-paging for interactive searches**," and "**-NOpaging for batch searches.**"

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 an underscore (_) to the list of characters that make up words. This is useful if you are searching for a pascal-style variable, but it can be used for any search string containing an underscore. This option has the same effect as the "**-w -s_**" option combination. The default is **-NOpascal**.

-prefetch

The **-prefetch** command is the same as **-g**.

-printable

This tells MAGNET to change non-printable characters into dots (.). This option is only meaningful when used in conjunction with the **-l** option.

-q

This 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 redirecting to a file and manipulating with an editor. The default is **not quiet**. For example, **:magnet "-f@ -b -otest2 -q -m 'procedure'"** would create a permanent file called **test2** containing a list of all files found (using LISTF,6 format) which contain the string, "procedure," in lowercase.

-qedit

This tells MAGNET to treat (and read) QEDIT files as though they were ASCII files. The **-NOqedit** option tells MAGNET to skip reading QEDIT files. The default is **-qedit**.

-s specials

This specifies a string of characters to be considered as non-terminators when a WHOLE-WORD search is done (see **-w** option). The first example in the *Example* section, which follows, illustrates how this is accomplished. The default is (empty).

-spl

Add an apostrophe (') to the list of characters that make up words. This is useful if you are searching for SPL-style variables, but it can be used for any search string containing an apostrophe. This option has the same effect as the "**-w -s**" option combination. The default is **-NOspl**.

-splash

Add an apostrophe (') and an underscore (_) to the list of characters that make up words. This is useful if you are searching for a SPLash!-style variable, but it can be used for any search string containing an apostrophe and underscore. The default is **-NOsplash**.

-t

This option designates a text string as a target of the search, but it is not required. For example, **:magnet "-f@ -t copyright"** is functionally identical to **:magnet "-f@ copyright"** in that both statements cause MAGNET to search for the word, "copyright."

-tellop

This option tells MAGNET to send a message to the operator console announcing each matching file found. The default is **-NOtellop**.

-timestamp

This tells MAGNET to print a timestamp (date and time) at the conclusion of its search.

-u

This option is used to display text matches without the enhancements that MAGNET normally uses to highlight the specified string. This option is useful when used in conjunction with the **-l** option if you don't want the specified string contained in the line to be highlighted.

For example, **:magnet "-f@ -l -u 'version'"** would search all files in the logon group and list all lines found that include the string, "version." No portion of the line will be enhanced by MAGNET.

-w

This 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. The first example in the section illustrates how this is accomplished. The default matches any string.

-y

Using this option produces a short summary of the search results. The default is **-NOy**. *Note*: If **-p** is used with **-y**, the summary is sent to both the terminal and to the **-p** device.

-[

This option tells MAGNET to only check for a match against strings that start in column #1.

For example, **:magnet "-f@.source -c -['procedure'"** would cause MAGNET to locate all occurrences of the string, "procedure," that start in column #1.

-(<number>

This command is used to change MAGNET's default left margin. Normally MAGNET's left margin starts at column #1, however this command can be used to modify that setting.

For example, **:magnet "-f@ -c -(10 'first'"** causes MAGNET to change its left margin to column 10 prior to searching for the string, "first," in columns 10-n, where "**n**" is the width of the file.

-) <number>

This command is used to change MAGNET's default right margin. Normally MAGNET's right margin is set to the record width of the file being processed.

For example, **:magnet "-f@ -c -)60 'first'"** would cause MAGNET to change its right margin to column 60 prior to searching for the string, "first," in columns 1 through 60 only.

-72

Shorthand for "**-)72**", which is the value specified for the "-)" option that limits searching to the first 72 characters of each record. This option is useful when your search string contains numbers and you want to avoid "matches" for record numbers in numbered files.

MAGNET Examples

In the first example, we want to find the string, "foo_fum," which contains an underscore character as part of the word. To do this, the **-s** (special character) and the **-w** (whole word) options are used. The **-s** option is used because MAGNET normally treats the underscore as a word separator. In this example, we want the underscore to be part of the string. The **-w** option is used to explicitly define the string as "foo_fum." Thus, MAGNET would bypass a word like "foo_fum_fie" when the **-w** option is used. Without the **-w** option, a word like "foo_fum_fie" would be included in MAGNET output. Thus, your command line would look like the following:

:magnet "-f@.source -s_ -w -c foo"

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

:magnet "-f@.source-@q.source-bnf.source -l foo"

The above command line shows how to tell MAGNET to search all files in the SOURCE group except the file BNF.SOURCE, as well as any file name ending in "Q." List each line that contains the string, "foo."

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 9.4*.

```
:magnet "-f@.help.lpstools-standard.help.lpstools -c -l standard"
MAGNET [2.2] - LPS Toolbox [A.01a]          (c) 1995 Lund Performance Solutions


CAPTURE.HELP.LPSTOOLS
disk 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
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 20 files (12426 lines, 14 hits) in 1.376 seconds
:
```

Figure 9.4 - MAGNET Output on a Text String Search

MAGNET Error Messages

Message	***Invalid file set.***
Cause	Improper use of fileset specification.
Action	Review "specifying filesets" in the MAGNET chapter.

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

Chapter 10

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 Appendix B.) 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

Program capabilities required include IA, BA, DS and PH. User SM capability is required to run MODA.

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

Command Codes	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
NewGroup	Edits attributes for a new group
NewUser	Edits attributes for a new user
SET/REset	Enables and disables options
User	Displays attributes for a user

Command Definitions

Following is a detailed description of each MODA command.

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

This command replicates or clones the account structure of an existing account into a new account. The new account will be created with the same attributes as the old account.

For every user and group within the old account, a duplicate will be created within the new account. The groups and users created within the new account will have the same attributes as the old account. *Note*: UDC settings and Private Volume information are not replicated between accounts nor are the files copied.

The COPYACCOUNT and CLONEACCOUNT commands are very similar. The COPYACCOUNT command requires that the new account already exists, while the CLONEACCOUNT command requires that the new account must not currently exist.

If the ACAP = ALL option is used, then the new account will have all capabilities rather than the set from the old account.

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

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

The NOPASS keyword will cause the new account 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. CLONEACCOUNTQ and CLONEQ have the QUIET keyword assumed.

Using the CLONE commands requires SM capability.

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

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

The CREATE keyword instructs the COPYACCOUNT command to create within the new account any group or user found within the old account that does not currently exist.

The COPYACCOUNT and CLONEACCOUNT commands are very similar. The COPYACCOUNT command requires that the new account already exists, while the CLONEACCOUNT command requires that the new account must not currently exist.

If the ACAP = ALL option is used, then the new account will have all capabilities, rather than the set from the old account.

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

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

The NOPASS keyword will cause the passwords for the new account and all of its groups and users to remain unchanged.

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

Using the COPYACCOUNT command requires SM capability.

Exit

The Exit command terminates MODA.

Group <name> [<subset>]

The GROUP command displays the attributes for the specified group. These attributes may be edited as required. 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.**"

Using the GROUP command requires SM capability.

HELP

The HELP command invokes the MODA Help facility.

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 template account, then a default set of attributes is chosen for editing. If a template account is entered, then MODA will fetch the attributes for that account and display them for editing. The template account 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.

1. New account without a template; standard capabilities.
   NEWACCOUNT FOO

   This command displays text to be edited that looks like the following:

   :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=

2. New account without a template; all capabilities.
   NA FOO,CAP=ALL

   This command displays text to be edited that looks like the following:

   :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=

3. New account, FOO, looks like the LPSTOOLS account.
   NEWACCOUNT LPSTOOLS

   This will result in the following note:

   LPSTOOLS exists, used as a template

   Text to be edited looks like this:

   :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=

Changing the question mark (?) to read "FOO" (not done here) would complete the exercise.

NewGroup [template group] [, CAP = ALL]

The NEWGROUP command (minimum abbreviation NG) is used to edit the attributes for a new group. If NEWGROUP is entered without a template group, then a default set of attributes is chosen for editing. If a template group is entered, then MODA will fetch the attributes for that group and display them for editing. The template group 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 template group exists.

See NEWACCOUNT for examples similar to NEWGROUP.

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 template user, then a default set of attributes is chosen for editing. If a template user is entered, then MODA will fetch the attributes for that user and display them for editing. The template user 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 template user exists.

See NEWACCOUNT for examples similar to NEWUSER.

User <name> [<subset>] [, CAP=ALL]

The USER command displays the attributes for the specified user. These attributes may be edited as required. 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.

Command Code	Description
^A	Goto end of line (append)
^B	Activate Insert mode
^D	Delete the character at the cursor position
^E	Erase contents from cursor position to end-of-line
^G	Oops, undo changes
^H	Backspace (non-destructive) cursor left
^O	Activate Overwrite mode
^Y	Abort changes
<Space bar>	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 Appendix F for a complete description of the MODIFY editor.

MODA Examples

Following is an example of the MODA tool **cloneaccount** command.

```
:moda
MODA [2.0] - LPS Toolbox [A.01a]            (c) 1995 Lund Performance Solutions

For Help at the MODA prompt enter   ?

MODA: cloneaccount lpstools,lpstback
Cloning from account LPSTOOLS to account LPSTBACK
Created account LPSTBACK
   cloned new group CFG.LPSTBACK
   cloned new group CM.LPSTBACK
   cloned new group CMD.LPSTBACK
   cloned new group DATA.LPSTBACK
   cloned new group EXTERNAL.LPSTBACK
   cloned new group HELP.LPSTBACK
   cloned new group JOB.LPSTBACK
   cloned new group O.LPSTBACK
   cloned new group PUB.LPSTBACK
   cloned new group PUBSYS.LPSTBACK
   cloned new group RL.LPSTBACK
   cloned new group SOURCE.LPSTBACK
   cloned new group USL.LPSTBACK
   cloned new group XL.LPSTBACK
   cloned user  MANAGER.LPSTBACK
   cloned user  MGR.LPSTBACK
Cloned ok

MODA: exit
:
```

Figure 10.1 - Clone Account

MODA Error Messages

Message	***Expected valid account name.***
Cause	User entered an account-name that does not conform with the convention established by Hewlett-Packard for account names.
Action	Review NEWACCT section in the MPE/iX Command Reference Manual for a complete description of the format for account names.
Message	***Expected valid group name.***
Cause	User entered a group-name that does not conform with the convention established by Hewlett-Packard for group names.
Action	Review NEWGROUP section in the MPE/iX Command Reference Manual for a complete description of the format for group names.
Message	***Expected valid user name.***
Cause	User entered a user-name that does not conform with the convention established by Hewlett-Packard for user names.
Action	Review NEWUSER section in the MPE/iX Command Reference Manual for a complete description of the format for user names.
Message	***You must have AM or SM capability to run this program.***
Cause	User who tried to run MODA did not have adequate capabilities to run MODA.
Action	Log into an account that has proper capabilities or run the GRANT from the ***System Managers Toolbox*** to temporarily give yourself the necessary capabilities.

Chapter 11

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

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 advantage realized by adding more 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 2,048 bytes. So, for a system with 64 MB of physical memory there would be 32,768 physical pages of memory available for the Memory Manager. 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.

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. Presently, MPE/iX machines can address 2^{16} virtual spaces, for a total virtual address of 48 bits. Systems which employ 48-bit virtual addresses are known as "Level One" systems. A "Level Two" system would be able to address 2^{32} virtual spaces, for a total virtual address of 64 bits. 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 isn't 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 matched 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 0-799 (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, each number characterizes the type of data in the page. See the files PAGES@.DATA.LPSTOOLS for definitions of all 800 object classes.

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 memory 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 data. This includes:

Class Name	Class Number			
	Pre-3.0	**3.0-3.1**	**4.0**	**4.5**
CM_DATA	006	006	006	006
CM_USER_DATA	327	327	552	552

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 five object classes:

Class Name	Class Number			
	Pre-3.0	**3.0-3.1**	**4.0**	**4.5**
TURBO_ROOT	482	482	694	694
TURBO_DATA_SET	483	483	695	695
TURBO_DATA_BASE_ACCESS	484	484	696	696
TURBO_ILR_LOG	486	486	698	698
TURBO_DBRECOV_RESTART	487	487	699	699

The object classes associated with TurboIMAGE control blocks are considered to be part of the SYSTEM_CLASS.

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.

Class Name	Class Number			
	Pre-3.0	**3.0-3.1**	**4.0**	**4.5**
NM_STACK	002	002	002	002
CM_STACK	003	003	003	003
NM_CODE	004	004	004	004
CM_CODE	005	005	005	005
NM_HEAP	083	083	083	083
CM_USER_CODE	325	325	550	550
NM_PROGRAM	413	413	663	663
CM_PROGRAM	414	414	664	664

Note: Pages belonging to the stack or heap of system processes are counted as part of the USER_CLASS. PAGES cannot determine the ownership status of stack/heap pages held 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.

Command Code	Description
Exit	Terminates PAGES
FIND <what>	Searches for specified virtual address
FRozen #	Searches for frozen count
HELP	Invokes PAGES help
HPDIR	Display HPDIR information
Objclass #	Searches for specified Object Class
Processes	Report PIN and memory PAGE information
RAMUSage	Report affect of adding memory
SCan	Scans through memory (long report)
SET/REset	Enables and disables options
Status	Reports memory status (short report)

Command Definitions

This section discusses each of the PAGES commands in detail. At the end of this section is information on each of the options that can be used with the [RE]SET command.

FIND [<virtual address>]
[ROC] [IMI] [PRESent] [ABSent] [DIRTy] [REFerenced] [UNUSED]
[FROZen [#1 [/#2]]]
[OBJclass #1 [/#2]]
[PID #1 [/#2]]
[PPAGE #1 [/#2]]

The FIND command looks at every entry in the PDIR, searching for pages that match user specifications.

Specifying more than one option usually results in a page having to meet all of the options. Exceptions to the rule include: ABSent, IMI, PRESent, and ROC. If a page meets any of those four specified options AND all other options, then it is displayed.

<virtual address> = **<spaceid>.<offset>**
<spaceid>.@
offset
ALL

If a virtual address was specified, only those pages that match that address will be reported.

Note: Sometimes, several physical pages appear to be associated with the same virtual address. PAGES cannot distinguish between these to determine which, if any, is currently "active."

```
:pages
PAGES [2.22] - LPS Toolbox [A.01a]          (c) 1995 Lund Performance Solutions

For Help at the PAGES prompt enter   ?

SERIES 917LX
MPE/iX 5.0 (or later)
#CPUS: 1
Memory size: 56 MB (58,720,256 bytes; 14,336 logical pages)

PAGES: find $c0000000
Looking for virtual address: $a.$c0000000
Found logical page $a21, virtual page = $0000000a.$c0000000
   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 = TRANSIENT_DATA
Found 1 4KB pages

PAGES: exit
:
```

Figure 11.1 - FIND Command

The FIND command displays the following header line:

Phys$ VirtSpace.VrtOffset RIVDB $PID MN Fr Sta Ob# Object Class Name

FIND Column Headers

Header	Definition
Phys$	Physical page number
VirtSpace	Virtual space identifier (upper 32 bits of virtual address)
VrtOffset	Virtual offset (lower 32 bits of a virtual address)
R	Referenced Bit
I	(unknown)
V	(unknown)
D	Dirty Bit
B	Data Breakpoint bit. This bit is on when a page has a data breakpoint on it. Data breakpoints are set up with DEBUG's DATAB command
$PID	Protection ID; a 15-bit value representing the "lockword" for a page
M	Memory-Resident
N	(unknown)
Fr	Frozen counter
Sta	Page State. The values for this field are PRESENT, IMI, ROC, and ABSENT
Ob#	Object Class Number
Object Class Name	Every "object" (a range of virtual addresses) created by MPE/iX has a number associated with it called the "object class". This number can be used to help determine how many pages of disk 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) The ALL option for the FIND command tells PAGES to look at all physical pages, not just those associated with a particular virtual address or virtual space.

```
:pages
PAGES [2.22] - LPS Toolbox [A.01a]          (c) 1995 Lund Performance Solutions

For Help at the PAGES prompt enter   ?

SERIES 917LX
MPE/iX 5.0 (or later)
#CPUS: 1
Memory size: 56 MB (58,720,256 bytes; 14,336 logical pages)

PAGES: find all

Page$ VirtSpace.VrtOffset RD MN Froz Status    PIN Object_Class_Name
----- ---------.--------- -- -- ---- -------   --- -----------------
    1 $00000000.$00001000           1 PRESENT     0 TRANSIENT_DATA
    2 $00000000.$00002000           1 PRESENT     0 TRANSIENT_DATA
    3 $00000000.$00003000           1 PRESENT     0 TRANSIENT_DATA
    4 $00000000.$00004000           1 PRESENT     0 TRANSIENT_DATA
    5 $00000000.$00005000           1 PRESENT     0 TRANSIENT_DATA
    6 $00000000.$00006000           1 PRESENT     0 TRANSIENT_DATA
    7 $00000000.$00007000           1 PRESENT     0 TRANSIENT_DATA
    8 $00000000.$00008000           1 PRESENT     0 TRANSIENT_DATA
    9 $00000000.$00009000           1 PRESENT     0 TRANSIENT_DATA
    a $00000000.$0000a000           1 PRESENT     0 TRANSIENT_DATA
    b $00000000.$0000b000           1 PRESENT     0 TRANSIENT_DATA
    c $00000000.$0000c000           1 PRESENT     0 TRANSIENT_DATA
    d $00000000.$0000d000           1 PRESENT     0 TRANSIENT_DATA
    e $00000000.$0000e000           1 PRESENT     0 TRANSIENT_DATA
    f $00000000.$0000f000           1 PRESENT     0 TRANSIENT_DATA
   10 $00000000.$00010000           1 PRESENT     0 TRANSIENT_DATA
   11 $00000000.$00011000           1 PRESENT     0 TRANSIENT_DATA
   12 $00000000.$00012000           1 PRESENT     0 TRANSIENT_DATA
   13 $00000000.$00013000           1 PRESENT     0 TRANSIENT_DATA
   14 $00000000.$00014000           1 PRESENT     0 TRANSIENT_DATA
Hit <return> to continue, / to stop:
```

Figure 11.2 - FIND ALL Option

SPACE ID The SPACE option tells PAGES to look only for physical pages that are used to hold virtual pages belonging to the specified space ID. For example: FIND SPACE $3a6.

ABSent Restricts the FIND command to just those physical pages that are in the "Absent" state.

DIRTy Restricts the FIND command to just those physical pages that have been written to and not yet posted to disk.

IMI Restricts the FIND command to just those physical pages that are in the "In Motion In" state.

PRESent Restricts the FIND command to just those physical pages that are in the "PRESent" state.

REFerenced Restricts the FIND command to just those physical pages that have been referenced (read or write) "recently." The memory manager periodically resets the referenced bits.

ROC Restricts the FIND command to just those physical pages that are in the "Recoverable Overlay Candidate State."

UNUSED	Restricts the FIND command to just those physical pages that have not been used.
FROZen [#1 [/#2]]	Restricts the FIND command to just those physical pages whose frozen count is in the range specified. *Note*: If /#2 is omitted, #2 is set to #1, which will search for just that frozen count. If #1 is also not given, #1/#2 defaults to 1/255.
OBJclass #1 [/#2]	Restricts the FIND command to just those physical pages whose object class is in the range specified. *Note*: If /#2 is omitted, #2 is set to #1, which will search for just that object class.
PID #1 [/#2]	Restricts the FIND command to just those physical pages whose protection ID (PID) is in the range specified. *Note*: If /#2 is omitted, #2 is set to #1, which will search for just that PID.
PPAGE #1 [/#2]	Restricts the FIND command to just those physical pages in the range #1 to #2. *Note*: If /#2 is omitted, #2 is set to #1, which will "find" at most one physical page. *Note*: If the ABSent option is not seen, then no pages in the "absent" state will be shown.

FRozen

The FRozen 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.

Objclass

The Objclass 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 [min#pages] [<SORT | NOSORT>]

The PROCESSES command scans through the memory, trying to determine what process caused each page to be brought into memory. It then summarizes the number of pages by process.

min#pages	Reports processes that have brought at least that many pages into memory.
SORT	Reports the processes in order of descending number of pages brought into memory. *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 process. This usually means that the process either "page-faulted" on the page, or did a "prefetch" on the page. In the case of page shared by multiple processes, only the most recent page-fault (or prefetch) is know to us.

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 [% of memory]

The SCAN command provides a summary of memory utilization based on object classes. The report generated by this command may be greater than 250 lines in length.

```
:pages
PAGES [2.22] - LPS Toolbox [A.01a]          (c) 1995 Lund Performance Solutions

For Help at the PAGES prompt enter   ?

SERIES 917LX
MPE/iX 5.0 (or later)
#CPUS: 1
Memory size: 56 MB (58,720,256 bytes; 14,336 logical pages)

PAGES: scan
Memory size: 14,336 logical pages (56 MB)
      3,896 Dirty,     9,925 Referenced,     3,686 Frozen.        0 Unused.
  (   1,595 Dirty,     6,707 Referenced of the unfrozen pages)

   State of in-use pages:  14,238 Present, 0 IMI, 27 ROC, 70 Absent.

Frequency of "freeze" counts:
   freeze count    # pages
          0         10,649
          1          3,198
          2             94
          3            390
          4              2
          5              1
         31              1

Frequency of page type (from IDIR):

    PageType      Count         PageType      Count
    --------   ---------        --------   ---------
      DATA_R       2,224          GATE_0           3
     DATA_RW       6,453          GATE_1           2
        CODE       4,849          GATE_2           1
    CODE_RWX         803          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                2,720  |   2 NM_STACK                        370
  8 NM_SYS_LIB                    2,329  |   9 CM_SYS_LIB                      743
 44 Label Table                     334  |  63 PORT_AND_POOL                   269
 77 USER_NM_PROGRAM                 186  | 275 NM_SUBSYS_LIB                   233
290 XM_CB_POOL                      153  | 661 NM_SL                           155
663 NM_PROGRAM                    3,608  | 665 ORD_FIX                       1,047
671 NM_KSAM_FIX                     185  |
```

```
Memory usage by "type" of Object Class:

       Class      #LogicalPages   #MB  % total
   -------------- -------------   ---  -------
   SYSTEM_CODE           3,305    12    23.1%
   SYSTEM_DATA           5,876    19    35.4%
   USER_CODE             3,825    14    26.7%
   USER_DATA               238     0     1.7%
   USER_STACK              528     2     3.7%
   USER_FILE             1,363     5     9.5%

   Totals:              14,335    55   100.0%

   "User" pages are  41.5% of memory (24 MB out of 56 MB)

SID counts:   (#instances each SpaceID found in memory;
               when > 14 pages)

   $SID   #Pages | $SID   #Pages | $SID   #Pages | $SID   #Pages |
     $0   1,020 |    $a   4,832 |    $b     903 |   $11     160 |
    $15      17 |   $18      16 |   $19      52 |   $21     364 |
    $2d      19 |   $38     743 |   $3d      29 |   $45      14 |
    $81      33 |   $b1      33 |   $e0      60 |   $e1      18 |
    $e4      33 |   $e8     174 |   $f4      55 |  $179      53 |
   $17d      39 |  $199      16 |  $1a1      14 |  $1a4     233 |
   $1b5      15 |  $1c1      83 |  $1cd      75 |  $1e0     145 |
   $211      14 |  $241      98 |  $285      57 |  $288      20 |
   $295      23 |  $298      25 |  $2a8      19 |  $2a9      32 |
   $2e0     308 |  $2ec      79 |  $30d      28 |  $339      29 |
   $345      32 |  $349      32 |  $389      25 |  $431     105 |
   $471     274 |  $480      19 |  $484      32 |  $48d     138 |
   $499     126 |  $4c0     105 |  $4c5      27 |  $4cd      71 |
   $509     138 |  $52d      78 |  $551      28 |  $598      25 |
   $5b8      16 |  $5c4      25 |  $5d0      47 |  $5d1      47 |
   $5ec      15 |  $5ed      14 |  $5f0      25 |  $5f8      21 |
   $5fc      65 |  $62d      30 |  $660      82 |  $6d0     103 |
   $738     185 |  $739     154 |  $748      68 |  $764      68 |
   $76d     130 |  $770     122 |  $771     602 |  $7b4      40 |
   $7d0      28 |  $7d9      36 |  $7f1      67 |  $801     105 |
   $819      16 |  $82d     104 |  $865      40 |  $8ac      80 |

Largest SID ever seen: $8b0

PAGES: exit
:
```

Figure 11.3 - SCAN Command

SET | REset

Use the SET or RESET commands to specify the following options. Enable or disable these options as needed for the task at hand.

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.

LAUNCH This option directs the Find command to display process launch information. It is only valid for systems running under MPE 5.0.

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.

A complete list of SET/RESET options may be found by entering "?" at the PAGES prompt (i.e., **PAGES: ?**)

Status

The STATUS command produces the short form memory usage report displayed below. Use the :COPYLP command to obtain a listing of STATUS output.

```
:pages
PAGES [2.22] - LPS Toolbox [A.01a]          (c) 1995 Lund Performance Solutions

For Help at the PAGES prompt enter   ?

SERIES 917LX
MPE/iX 5.0 (or later)
#CPUS: 1
Memory size: 56 MB (58,720,256 bytes; 14,336 logical pages)

PAGES: status
Memory size: 14,336 logical pages (56 MB)
      3,896 Dirty,    9,893 Referenced,    3,690 Frozen.        0 Unused.
  (   1,591 Dirty,    6,671 Referenced of the unfrozen pages)

   State of in-use pages:  14,218 Present, 0 IMI, 47 ROC, 70 Absent.

Memory usage by "type" of Object Class:

       Class      #LogicalPages  #MB  % total
   ------------- -------------  ---  -------
   SYSTEM_CODE           3,291   12    23.0%
   SYSTEM_DATA           5,080   19    35.4%
   USER_CODE             3,827   14    26.7%
   USER_DATA               238    0     1.7%
   USER_STACK              528    2     3.7%
   USER_FILE             1,371    5     9.6%

   Totals:              14,335   55   100.0%

   "User" pages are  41.6% of memory (24 MB out of 56 MB)

PAGES: exit
:
```

Figure 11.4 - 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.

PAGES Examples

The following illustrates a PAGES operation showing the amount of memory used by the object CM_STACK (objclass number 3).

```
:pages
PAGES [2.22] - LPS Toolbox [A.01a]          (c) 1995 Lund Performance Solutions

For Help at the PAGES prompt enter   ?

SERIES 917LX
MPE/iX 5.0 (or later)
#CPUS: 1
Memory size: 56 MB (58,720,256 bytes; 14,336 logical pages)

PAGES: objclass 3
Logical page:   $ae, Virtual: $000007d9.$41616000
Logical page:  $212, Virtual: $00000295.$41616000
Logical page:  $517, Virtual: $0000003c.$41616000
Logical page:  $798, Virtual: $000005d0.$41617000
Logical page:  $830, Virtual: $000007d9.$41615000
Logical page:  $848, Virtual: $0000000b.$40011000
Logical page:  $850, Virtual: $000007b4.$41615000
Logical page:  $865, Virtual: $000005d0.$41618000
Logical page:  $a9e, Virtual: $000002ec.$41615000
Logical page: $10b2, Virtual: $000005f0.$41616000
Logical page: $113f, Virtual: $000005f0.$41615000
Logical page: $16c4, Virtual: $00000318.$41616000
Logical page: $16d3, Virtual: $0000003c.$41618000
Logical page: $173B, Virtual: $000005d0.$41615000
Logical page: $173b, Virtual: $000001c1.$41617000
Logical page: $1827, Virtual: $000005f5.$41615000
Logical page: $190a, Virtual: $00000295.$41615000
Logical page: $1e0b, Virtual: $00000480.$41617000
Logical page: $1f9f, Virtual: $0000017d.$41616000
Logical page: $1fc7, Virtual: $0000017d.$41615000
Logical page: $2035, Virtual: $000001a1.$41616000
Logical page: $2415, Virtual: $0000035d.$41615000
Logical page: $24d0, Virtual: $000005d1.$41615000
Logical page: $25c1, Virtual: $000002a8.$41616000
Logical page: $25d8, Virtual: $00000318.$41615000
Logical page: $277a, Virtual: $00000865.$41616000
Logical page: $2b70, Virtual: $0000021c.$41616000
Logical page: $2b71, Virtual: $0000021c.$41615000
Logical page: $2bb5, Virtual: $000005d0.$41616000
Logical page: $2c0d, Virtual: $000002ec.$41616000
Logical page: $2cda, Virtual: $000001c1.$41615000
Logical page: $2e6e, Virtual: $00000045.$41615000
Logical page: $2e7a, Virtual: $000001a1.$41615000
Logical page: $2e8f, Virtual: $000000e1.$41615000
Logical page: $2e95, Virtual: $000001dc.$41615000
Logical page: $2fda, Virtual: $0000003c.$41617000
Logical page: $2ff3, Virtual: $000005d1.$41616000
Logical page: $30ce, Virtual: $00000480.$41616000
Logical page: $31b1, Virtual: $00000089.$41615000
Logical page: $31b8, Virtual: $00000040.$41615000
Logical page: $31ef, Virtual: $00000191.$41615000
Logical page: $3201, Virtual: $00000084.$41615000
Logical page: $3224, Virtual: $000001a0.$41615000
Logical page: $3254, Virtual: $00000085.$41615000
Logical page: $3256, Virtual: $00000865.$41615000
Logical page: $3259, Virtual: $00000088.$41615000
Logical page: $326d, Virtual: $00000349.$41616000
...
Logical page: $33cf, Virtual: $00000220.$41615000
Logical page: $33d3, Virtual: $000000e4.$41615000
```

```
Logical page: $33d4, Virtual: $0000016d.$41615000
Logical page: $33d5, Virtual: $00000190.$41615000
Logical page: $33da, Virtual: $00000178.$41615000
Logical page: $33dc, Virtual: $0000019d.$41615000
Logical page: $33e0, Virtual: $000001c8.$41615000
Logical page: $33e2, Virtual: $000001c0.$41615000
Logical page: $33e4, Virtual: $000001c4.$41615000
Logical page: $33eb, Virtual: $0000038d.$41615000
Logical page: $33ed, Virtual: $000002ac.$41615000
Logical page: $33ee, Virtual: $00000324.$41615000
Logical page: $33f9, Virtual: $00000335.$41615000
Logical page: $33fd, Virtual: $00000315.$41615000
Logical page: $33ff, Virtual: $00000330.$41615000
Logical page: $3406, Virtual: $000003b0.$41615000
Logical page: $3409, Virtual: $0000035c.$41615000
Logical page: $343d, Virtual: $00000480.$41615000
Logical page: $34c0, Virtual: $000002d0.$41615000
Logical page: $357b, Virtual: $00000361.$41615000
Logical page: $3599, Virtual: $00000288.$41616000
Logical page: $35df, Virtual: $000007b4.$41616000
Logical page: $371c, Virtual: $000005d1.$41617000
Logical page: $3742, Virtual: $000001c1.$41616000

PAGES: exit
:
```

Figure 11.5 - Object Memory

The following shows various applications of the FIND command.

```
:pages
PAGES [2.22] - LPS Toolbox [A.01a]          (c) 1995 Lund Performance Solutions

For Help at the PAGES prompt enter   ?

SERIES 917LX
MPE/iX 5.0 (or later)
#CPUS: 1
Memory size: 56 MB (58,720,256 bytes; 14,336 logical pages)

PAGES: find roc

Page$ VirtSpace.VrtOffset RD MN Froz Status    PIN Object_Class_Name
----- ---------.--------- -- -- ---- -------  ---- ----------------
   a6 $00000015.$002a4000          0 ROC        79 XM_LOG
  6f5 $0000000a.$0067d000          0 ROC        74 NM_SYS_LIB
  79b $0000000a.$005a9000          0 ROC        74 NM_SYS_LIB
  fe4 $0000000a.$00d8d000          0 ROC        76 NM_SYS_LIB
 1031 $000001cd.$007a8000          0 ROC        76 NM_SL
 1068 $000001cd.$007ac000          0 ROC        76 NM_SL
 11f1 $00000015.$0029f000          0 ROC        79 XM_LOG
 12de $00000038.$01380000          0 ROC        76 CM_SYS_LIB
 130a $0000000a.$00422000          0 ROC        74 NM_SYS_LIB
 13c7 $000001cd.$007ad000          0 ROC        76 NM_SL
 1468 $000001cd.$007b2000          0 ROC        76 NM_SL
 166e $00000038.$01550000          0 ROC        76 CM_SYS_LIB
 167f $00000015.$002ae000          0 ROC        49 XM_LOG
 16f2 $00000038.$01569000          0 ROC        76 CM_SYS_LIB
 1700 $00000038.$0137e000          0 ROC        76 CM_SYS_LIB
 1724 $00000038.$0155d000          0 ROC        76 CM_SYS_LIB
 1772 $00000038.$01568000          0 ROC        76 CM_SYS_LIB
 1841 $0000000a.$005aa000          0 ROC        74 NM_SYS_LIB
 188b $0000000a.$00b13000          0 ROC        74 NM_SYS_LIB
 188d $0000000a.$00b0c000          0 ROC        74 NM_SYS_LIB
```

```
Page$ VirtSpace.VrtOffset RD MN Froz Status   PIN Object_Class_Name
----- ---------.--------- -- -- ---- ------   --- -----------------
Found 20 4KB pages


PAGES: find dirty

Page$ VirtSpace.VrtOffset RD MN Froz Status   PIN Object_Class_Name
----- ---------.--------- -- -- ---- ------   --- -----------------
   52 $000000e8.$00147000 RD        0 PRESENT  69 Label Table
   69 $000000e8.$004fc000 RD        0 PRESENT  64 Label Table
   6c $0000000a.$d5e98000 RD        0 PRESENT  74 CM_USER_DATA
   6e $000005ed.$00007000 RD        0 PRESENT  40 ORD_FIX
   72 $00000551.$0000d000 RD M      1 PRESENT   - COUNTER_ARRAY
   74 $0000000a.$d3438000 RD        0 PRESENT  42 LDR_SYS_PLABEL
   79 $0000000a.$d5f49000 RD        0 PRESENT   3 PIBX
   80 $0000017d.$40206000 RD        0 PRESENT  43 LDR_PFL table
   85 $0000017d.$40207000 RD        0 PRESENT  43 LDR_PFL table
   86 $0000000b.$83659000 RD        0 PRESENT  70 XM_CB_POOL
   88 $0000000b.$85ffb000 RD        0 PRESENT  43 FS_PLFD table
   8d $000000e8.$000a7000  D        0 PRESENT  78 Label Table
   8f $00000865.$40203000 RD        0 PRESENT  57 LDR_PFL table
   9b $0000000b.$86040000 RD        0 PRESENT  70 UNSPECIFIED_PORT
   a9 $00000000.$000a9000 RD M      1 PRESENT   0 TRANSIENT_DATA
   b7 $00000011.$00238000 RD        0 PRESENT  65 Label Table
   b8 $00000011.$00435000 RD        0 PRESENT  73 Label Table
   b9 $0000000b.$80000000 RD M      1 PRESENT   0 TRANSIENT_DATA
   ba $0000000b.$80001000 RD M      1 PRESENT   0 TRANSIENT_DATA
   bc $0000000b.$80003000 RD M      1 PRESENT   0 TRANSIENT_DATA

Page$ VirtSpace.VrtOffset RD MN Froz Status   PIN Object_Class_Name
----- ---------.--------- -- -- ---- ------   --- -----------------
Found 20 4KB pages


PAGES: find unused

Page$ VirtSpace.VrtOffset RD MN Froz Status   PIN Object_Class_Name
----- ---------.--------- -- -- ---- ------   --- -----------------
    0 $0000188d.$00003635       65535 PRESENT   0 EXT_B_TREE table
 1248 $0000000a.$d6181000 RD       1 ABSENT   65 LDR_UNSAT table
 1316 $0000000a.$d6180000 RD       1 ABSENT   65 LDR_UNSAT table
 13ec $00000318.$422d0000 RD       1 ABSENT   49 NM_HEAP
 1635 $00000318.$423c4000 RD       1 ABSENT   49 NM_HEAP
 1639 $00000318.$423c3000 RD       1 ABSENT   49 NM_HEAP
 1644 $00000318.$423c2000 RD       1 ABSENT   49 NM_HEAP
 164c $00000318.$423bf000 RD       1 ABSENT   49 NM_HEAP
 1660 $00000318.$423c1000 RD       1 ABSENT   49 NM_HEAP
 185f $0000000a.$d6184000 RD       1 ABSENT   65 LDR_UNSAT table
 1f44 $0000000a.$d6186000 RD       1 ABSENT   65 LDR_UNSAT table
 2c00 $0000000a.$d6182000 RD       1 ABSENT   65 LDR_UNSAT table
 317e $0000000a.$d6187000 RD       1 ABSENT   65 LDR_UNSAT table
 3425 $0000000a.$d6183000 RD       1 ABSENT   65 LDR_UNSAT table
 35ee $00000318.$4183e000 RD       1 ABSENT   49 NM_STACK
 36a9 $0000000a.$d6185000 RD       1 ABSENT   65 LDR_UNSAT table
 36b2 $00000318.$4183f000 RD       1 ABSENT   49 NM_STACK
Found 17 4KB pages

PAGES: exit
:
```

Figure 11.6 - FIND Command

PAGES Error Messages

Message	***Bad frozen count value***
Cause	Either no number or an illegal number was entered.
Action	Enter a number in the range 0 to 127 (decimal).
Message	***Bad Object Class number***
Cause	Either no number or an illegal number was entered.
Action	Enter a number in the range 0 to 500 (decimal).
Message	***Expected an address***
Cause	Invalid **virtual address** given to FIND command.
Action	Enter valid **virtual address**, see FIND command.
Message	***Expected offset within space***
Cause	Bad **virtual address** offset specification.
Action	FIND **spaceid** expects "**.<offset>**" or "**.@**"
Message	***Expected space ID***
Cause	Bad **virtual address** space id specification.
Action	FIND SPACE expects **spaceid**.
Message	***Failed to open PAGES.DATA.LPSTOOLS data file.***
Cause	PAGES could not open companion file.
Action	Check to ensure that PAGES.DATA.LPSTOOLS is available.
Message	***Frozen count must be in range 0..127***
Cause	A number not in range 0 to 127 was entered as a FROZEN count.
Action	Choose a number in range 0 to 127 (decimal).
Message	***Unexpected I/O error reading PAGES.DATA, unable to load Class data.***
Cause	FILE SYSTEM I/O error during the loading of PAGES.DATA.LPSTOOLS.
Action	Ensure the integrity of the file PAGES.DATA.LPSTOOLS
Message	***Unknown class type***
Cause	During the loading of PAGES.DATA.LPSTOOLS, an unknown class type was discovered.
Action	Correct the offending entry to the file PAGES.DATA.LPSTOOLS.

Chapter 12

The REDWOOD Tool

REDWOOD identifies frequently accessed files by scanning the system log files and extracting FILE CLOSE information. Additionally, REDWOOD will report any I/O errors that have been recorded in the system log files.

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 on 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 instructions 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:

```
SYSGEN version D.00.00 : catalog version D.00.00      WED, APR 3, 1991, 3:47
Copyright 1987 Hewlett-Packard Co. All Rights Reserved.

       ** 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
```

Figure 12.1 - SYSGEN Program Screen

3. Next, type "**log**" at the :SYSGEN prompt to display the following configuration commands.

```
**LOG configuration commands**

show (sh)           slog (sl)           ulog (ul)

clear (cl)(c) exit (ex)(e)        help (he)(h)        hold (ho)
oclose (oc)         redo
```

Figure 12.2 - 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.

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

The following list provides a summary description of REDWOOD 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 REDWOOD requires in order to distinguish one command from another. However, the commands themselves are not case-sensitive.

Command Codes	Description
CReate	Creates a "summary log" file
EXClude	Sets/resets exclusion options
Exit	Terminates REDWOOD
HELP	Invokes REDWOOD help
List	Sorts & reports a "summary log" file
LP	Directs LIST output to a printer
SCAN	Same as CReate, except that no summary file is created
SORTSCR	Scans for SORT intrinsic SORTSCR files
SET/REset	Enables or disables REDWOOD options
TERMinal	Directs LIST output to terminal

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 specified log file(s) are read sequentially and all type 5 (CM FCLOSE) and 105 (NM FCLOSE) records for disk files (subtype 0) are extracted. These records are sorted by 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 seven-digit number, are gathered into a single record for each **group/account**. Similarly, FSEDIT work files of the form "**Fnnnnnnn**" are gathered into a single record for each **group/account**. This is controlled by the [RE]SET EDITOR and [RE]SET FSEDIT commands.

After scanning all specified log files, REDWOOD's temporary sort file is then sequentially read 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/iX), 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 FCLOSEs. If this indicator is set to TRUE, then 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.

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 will be used to hold the summary records for each FCLOSEed file.

Depending on the number of log file(s) and their sizes, it may take REDWOOD quite a while to process all of the information. So, during the log file scanning phase, REDWOOD 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 [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
```

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 disk files.

SMALL # 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 indicate zero (0) blocks were transferred.

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 issued 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 totaled, 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 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

List Header The LIST command produces a report with a header like the following:

```
                                                J       Maximum          Average
File.Group.Acct  Ldn #Record  #Fcloses  Cum   S  R/B  sectors   sectors
---------------  --- -------  --------  ---   -  ---  -------   -------
```

Figure 12.3 - LIST Command Report Header

Each of these fields is described next.

Field	Description
File.Group.Acct	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 by 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 by default.
LDN	The LDN column reports the approximate ldev for each file. *Note*: The actual ldev is not given. Instead, the volume-set index is shown. (This is because MPE/iX puts the index, and not the ldev, into the file-close record.) For disks 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. 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.
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.

LIST Special Characters Some columns in the LIST output have special characters to indicate various things. This section documents each of these characters.

Character	Description
*	An asterisk (*) after LDEV column after the ldev 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	An "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 to device = LP. All reports are sent to this file until a TERMINAL or EXIT command is 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 standard ***LPS-Tools*** 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

The SET and RESET commands are used to specify the following options.

ARITRAP	Instructs REDWOOD to pay attention to any internal arithmetic faults that might occur. RESET ARITRAP tells REDWOOD to ignore faults.
CPU	Instructs REDWOOD to report its CPU usage at the end of handling a command.
mergeDOMAINS	Instructs 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	Instructs REDWOOD to merge all EDIT/3000 temporary files into one file.
FIRSThour = #	Instructs REDWOOD to ignore any file-close records that occurred before the specified hour of the day. The hour value must be in the range of 0 through 23.
FSEDIT	Instructs REDWOOD to merge all FSEDIT temporary files into one file.
GMulti	Instructs REDWOOD to attempt to compensate for the MPE bug in reporting the number of blocks transferred for files that were opened with GMULTI (e.g., message files). When compensation occurs, it is flagged as such in the summary report. RESET GMULTI disables the compensation.

LASThour = #	Instructs REDWOOD to ignore any file-close records that occurred after the specified hour of the day. The hour value must be in the range of 0 through 23.
LOGERRors	Instructs REDWOOD to report I/O errors in the logfile. RESET LOGERRORS tells REDWOOD not to report I/O errors.
NONZERO	This is a synonym for EXCLUDE NONZERO. RESET NONZERO disables excluding non-zero file-close records.
TEMPONLY	Instructs REDWOOD to exclude file-close records for permanent disk files. SET TEMPONLY is a synonym for EXCLUDE PERM. RESET TEMPONLY disables the exclusion.
ZERO	This 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.

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.

```
:redwood

REDWOOD [1.0] - LPSTOOLS [H.28.12]                          XYZ COMPANY [A0010]
(C) 1995 Lund Performance Solutions                         Albany, Oregon

For Help at the REDWOOD prompt enter  ?

Max sort records (based on  ;PARM=) are 40000
```

Figure 12.4 - Invoking a REDWOOD Session

The first step almost always involves creating the Summary Log. This is demonstrated in the next section.

```
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

LOG028        LOG0209       LOG0210       LOG0211

Enter first log file number (CR=exit) : 210

Enter last log file number (CR = 210): 210

Enter name of new summary file (CR = SUMLOG): TESTLOG
Initializing sort...Reading log files...
LOG0210 opened ok (eof = 916)
       148 desirable fclose records found
```

Figure 12.5 - 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 file 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 12.6 - 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 01/22/91 from MPEXL log file 210
Log files dated: 01/20/91..02/22/91
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 12.7 - 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 1.0 from Lund Performance Solutions  FRI, FEB 22, 1991, 2:38 PM

Summary created 02/22/91 from MPEXL log file 210
Log files dated: 02/20/91..02/22/91
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 12.8 - Output Report

REDWOOD Error Messages

Message	***Illegal file name***
Cause	An illegal file descriptor was entered.
Action	Review HP documentation concerning valid filenames.
Message	***Illegal file title:***
Cause	An invalid filename was given.
Action	Review HP's documentation about file name descriptions.
Message	***Oops...log files MUST be called LOG####[.group[.acct]]***
Cause	Log files can be moved into groups other than PUB.SYS.
Action	Log file names should not be altered; rename to original name given by MPE XL.
Message	***Sorry, that is a damaged summary file...you should probably purge it.***
Cause	The summary file is damaged and cannot be used.
Action	Use REDWOOD to reconstruct the summary file from individual log files.
Message	***That file has an improper file code.***
Cause	REDWOOD summary files use a file code of 1001.
Action	Use LISTF to locate summary files.
Message	***Unknown SET/RESET option:***
Cause	An unknown REDWOOD set/reset option was given.
Action	Valid choices for REDWOOD's set/reset options are: ARITRAP, CPU, DOMAINS, MERGEDOMAINS, EDITOR, FIRSTHOUR, FSEDIT, GMULTI, LASTHOUR, LOGERRORS, NONZERO, SORTSCR, TEMPONLY, ZERO

Chapter 13

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. Options include: the initial number of extents, max extents, 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 either 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.

If you are using NS/3000, the inputfile and outputfile file names are allowed to have nodenames (e.g., "**rep fun:fozzie, localfun**")

File equations can be used for either the input or output files (or both).

Options Summary

The following list provides a summary description of 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.

Option	Description
CODE	Specify filecode for output file.
CRUNCH	Close output file with crunch disposition.
DBSTORE	Updates timestamp & store flag for database copies.
DELAY	Pause during copy operation.
DEVICE	Specify outfile device.
DISABLESTAR	Tells REP not to print "*" in quiet mode.
DOTS (*9*)*	Display progress "dots" on screen during copy.
EXTENTS	Initial extents for output file.
FILECODE	Specify filecode for output file.
KEYFILE	New keyfile for KSAM file copies.
LDEV	Specifies disk drive for output file.
LOCAL (*14*)	Use logon group & account names for output file.
MAXBLOCKS	Specify maximum number of blocks for output file.
MAXEXTENTS	Specify maximum number of extents for output file.
NOKSAM	Copy KSAM file as non-KSAM file.
NULL	Does not perform copy, creates output file.
PURGE (*15*)	Overwrite (purge) output file if it exists.
QUIET (*13*)	Suppress some messages.
ROOTONLY (*12*)	Only copy rootfile of database.
TEMP	Create output file in temporary domain.
TIMES (*11*)	Display CPU usage.
XLCRUNCH	Close output file with xlcrunch disposition.
YES (*15*)	Overwrite (purge) output file if it exists.

*Numbers within the parentheses and in italic typeface are Parm bit values that you can set to select an option.

Options Definitions

Following is a detailed description of each of the REP options.

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

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, some users may want to update 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 [= #]

During large copy operations, use this option to 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 = devicename

Specifies the devicename where the output file should be sent.

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 REP to display progress dots (.) as it proceeds through a copy operation. The default setting is to not print progress dots.

EXTENTS = #

Specifies the initial number of extents to allocate for the output file.

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 = filepart

Specifies the name of the new key file for KSAM output files. The filepart may not have a group or account. This is restricted to CM KSAM files only.

LDEV = ldev#

Specifies the disk drive where the output file should be sent.

LOCAL

Output file is the same file-part name, but in the logon group & account.

MAXBLOCKS=#

Use this option to specify the maximum number of blocks for the output file.

MAXEXTENTS=#

Use this option 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

Using 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 okay to purge any existing file with the same name as the output file. NOPURGE tells REP that it must ask for permission prior to purging.

[NO]QUIET

QUIET Tells REP to suppress some output messages.

[NO]ROOTONLY

ROOTONLY tells REP not to 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 (default).

[NO]XLCRUNCH

XLCRUNCH tells REP to close the output file with the new "xlcrunch" disposition. This tells the file system to discard any unused disk space past the EOF but does not set the file limit down to the file eof. Thus, an xlcrunched file is expandable at some later point in time.

YES

The YES option is a synonym for PURGE. YES tells REP that it is okay to purge old files with the same name as the output file. By default, REP prompts before purging an existing TO file:

Old TO file:<filename> exists... Purge old TO file? (default - N).

REP Examples

Following are examples of the REP tool.

```
:Run rep.pub;info="olddb newdb dbstore"

REP [1.15] - LPSTOOLS 9H.28.12]                          XYZ COMPANY   [100110]
(c) 1995 Lund Performance Solutions                        Albany, Oregon

From OLDDB, to NEWDB
The FROM file is an IMAGE database...will copy entire db.
       (Will copy 10 sets)
         # of records copies = 11
       (copied root file)
       from:  OLDDB01
       to:    NEWDB01.TEST.LPSTOOLS (41 records, 1 chunk)
       from:  OLDDB02
       to:    NEWDB02.TEST.LPSTOOLS (12 records, 1 chunk)
       from:  OLDDB03
       to:    NEWDB03.TEST.LPSTOOLS (12 records, 1 chunk)
       from:  OLDDB04
       to:    NEWDB04.TEST.LPSTOOLS (101 records, 1 chunk)
       from:  OLDDB05
       to:    NEWDB05.TEST.LPSTOOLS (101 records, 1 chunk)
       from:  OLDDB06
       to:    NEWDB06.TEST.LPSTOOLS (101 records, 1 chunk)
       from:  OLDDB07
       to:    NEWDB07.TEST.LPSTOOLS (101 records, 1 chunk)
       from:  OLDDB08
       to:    NEWDB08.TEST.LPSTOOLS (211 records, 1 chunk)
       from:  OLDDB09
       to:    NEWDB09.TEST.LPSTOOLS (101 records, 1 chunk)
       from:  OLDDB10
       to:    NEWDB10.TEST.LPSTOOLS (400 records, 5 chunks)
```

Figure 13.1 - Database Copy

```
:run rep.pub;info="testfile newfile filecode+5555"

REP [1.15] - LPSTOOLS [H.28.12]                          XYZ COMPANY   [A00110]
(c) 1995 Lund Performance Solutions                      Albany, Oregon

From TESTFILE, to NEWFILE
(filecode := 5555)
       # of records copied = 1
```

Figure 13.2 - Specify Different Filecode

```
:run rep.pub;info="rep.source -xlcrunch"

REP [1.15] - LPSTOOLS [H.28.12]                          XYZ COMPANY   [A00110]
(c) 1995 Lund Performance Solutions                      Albany, Oregon

To := REP.TEST.LPSTOOLS
From REP.SOURCE, to REP.TEST.LPSTOOLS
       # of records copies = 4040

END OF PROGRAM
```

Figure 13.3 - XLCRUNCH Copy

REP Error Messages

Message	***Copying from $NULL is illegal!***
Cause	Bad source (i.e., FROM) filename.
Action	The FROM file must be a file that exists in the permanent or temporary domain.
Message	***Identical FROM file and TO file names are not allowed!***
Cause	Bad choice of name for either the FROM or TO file.
Action	Change either the FROM or the TO filenames to a different name.
Message	***IMAGE root files and datasets may not have lockwords.***
Cause	Lockwords were discovered on either the Image rootfile or dataset.
Action	REP does not support copying Image files with Lockwords.
Message	***TO file may not be a system file (except $NULL or $NEWPASS).***
Cause	User specified a destination filename that was neither $NULL nor NEWPASS.
Action	Enter a valid destination filename.

Chapter 14

The SHOT Tool

SHOT offers a comprehensive, flexible method for monitoring process activity on any MPE/iX machine. Various methods for reporting this information are available. SHOT can also be used to alter the state of a process.

Operation

Basically, SHOT provides information about a session, job, or a procedures process tree. This information can be organized in several ways based on user commands. The process tree is organized from top to bottom as parent above children. Indentation on the display is used to indicate children versus suspended processes. Across the top of each process tree that SHOT displays are many columns of information ranging from PINs to SwitchLevel. All of this information is configurable. See SHOT's ADM command and Process-Display section for details.

Viewing System Activity

One of SHOT's most useful features is its ability to locate processes which seem to be over-utilizing the CPU resources and other CPU-related activities. SHOT has several commands that allow you to view the CPU activity in which you are most interested. For example, the JOBS ONLY command will only display CPU information for the jobs currently executing. The DELTA command (default) displays all of the activity that has "passed" through the CPU on a periodic basis (which is user-definable). The DELTA display is a single-line-per-process display that summarizes which processes have been using the CPU. SHOT's ALL command will display all of the process information available at that time. This could easily run several pages, since all system process information is displayed. *Note*: SHOT's default is to display almost all user process activity.

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 Break, Resume or Kill commands. OP, SM or PM is required to use the PRI 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 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 States are:

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	
i	initiating	Process being born
u	unborn	Process just started being created

The **Process Identification Number** is denoted by the **PIN** column header. This is a 16-bit number which is reused when a process terminates. The PID is the Process ID, an extended form of the PIN. 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.

The **Job/Session Number** is denoted by the **Job#** column header. This column reports the job/session number that a process belongs to. By default, this column is suppressed. It can be requested by specifying SET MOST or SET ALL.

Total Process CPU Usage is denoted by the **CPU** column header. This 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 suffix 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 suffixes are: "s" for seconds, "m" for minutes, "h" for hours and "d" for days.

Process Name is denoted by the **Process Name** column header. This 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 invoked with "Start!". These processes are referred to as being "procreated." These processes do not have standard names (e.g., EDITOR.PUB.SYS) because there is no associated program file. SHOT tries to determine the name of the original procedure that initiated the process and, if successful, it displays the first 32 characters (or so) of that procedure name. If unsuccessful, the procedure address is shown in hexadecimal as a space and offset

(e.g., $a.4b2d90). *Note*: SHOT is unable to determine the names of procedures started from SL.PUB.SYS. It indicates these as "(CM Procedure)."

Process Priority is denoted by the **Pri** column header. This column shows the current priority of a process. The priority is a value in the range 0 through 255, with 0 being the highest priority. The MPE XL command, :SHOWQ, reports the base and limit priority values for the C, D, and E scheduling queues.

Process Queue is denoted by the **Q** column. This reports what scheduling queue a process is in. The possible queues are:

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

It is possible to put a process into four queues called BM, CM, DM, and EM. 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.

CPU Usage (Absolute) is denoted by the **Delta** column header. This column shows the amount of CPU time 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, and "d" for days.

CPU Usage By Percent is denoted by the **%** column header. This column shows the approximate percentage of available CPU time that a process has used since a process display was last 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.

Execution Mode is denoted by the **Ic** column. This 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.

Code	Description
Cn	Process is a Compatibility Mode program, currently in Native Mode
Cc	Process is a Compatibility Mode program, currently in CM
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 CM
On	Process is a Compatibility Mode program (OCT'ed), currently in NM
Pn	Process is a POSIX Native Mode program, currently in NM
Pc	Process is a POSIX Native Mode program, currently in CM

Note: Processes that are "procreated" native mode procedures will nave an "n" instead of "N" in the Initial column.

Switch Level is denoted by the **S** column header. This column shows the current nesting count of "switch" calls for both switch-to-NM and switch-to-CM activities. This is not a cumulative value, but instead reflects the number of switch markers you would see if a stacktrace for the process were 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.

INT PRI denotes the initial priority of a process.

TOT % displays the total percentage of CPU time the process has consumed since it began executing.

Process Type is denoted by the **Ptype** column header. This 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:

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.**"

Wait State is denoted by the **Wait State** column header. This column reports on why a process is waiting. On a single CPU machine, every process (except SHOT itself) should either be waiting for something to happen (e.g., a page to be read from disk) or should be "READY" to run. On a multiple CPU machine, several processes (in addition to SHOT) might be "EXECUTING" at the same time.

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:

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 summary description of SHOT 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 SHOT requires in order to distinguish one command from another. However, the commands themselves are not case-sensitive.

Command Code	Description
ADM	Formats process output display
All	Shows all processes
Break pin#	Suspends a process (requires SM capability)
CPU pin#	Prints the CPU usage for the specified pin
DEBUG	Invokes the HP DEBUG program
Delta	Shows processes using CPU between displays
DETERMINEPROCNAMES	Tells SHOT to regenerate SHOTPIDS.DATA
ERASEPROCNAMES	Tells SHOT to erase SHOTPIDS.DATA
Exit	Terminates SHOT
HELP	Invokes SHOT help
HELP NEWS	Display information on recent enhancements
HIGHLIGHT	Defines the display lines SHOT highlights
JOBINFO	Displays information about the next job and session
Job [Only]	Shows processes in a job-oriented display
JS pin#	Shows information about the specified pin
KILL pin#	Terminates the process
Pause #secs	Tells SHOT to pause for specified number of seconds
% #	Displays processes using this % of CPU per display
PIN pin#	Displays information about specified pin
Priority pin#	Changes priority of a process
PROG	Lists all program files currently in use.
Resume	Activates suspended process (SM capability)
Sessions	Lists all jobs and sessions
SET/REset	Includes options for selecting report items
SIRS	Reports on locked SIRS
Suspend	Suspends SHOT until a parent or child process awakens it
TABLES	Displays information on tables
TP pin#	Displays accumulated CPU time for a process tree
TRace pin#	Prints stack trace for specified pin
Tree [pin#]	Shows a nested Tree-oriented display of processes
USER	Shorthand for SET PTYPE=USER SON MAIN

Command Definitions

This section discusses each of the SHOT commands in detail. In several cases, we have provided syntax examples.

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. Example:

ADM
Active ADM field list:
ADM (STATE, PIN, CPU, NAME, QUEUE, DELTA, PERCENT, WAIT)

If ADM is followed by a simple list of fields, then the old list is cleared and the new list is used. 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 "**ctrl-Y**" or erase the entire list (**ctrl-X** followed by **ctrl-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. For example, if you enter "**ADM?**" the following will result

Field	Length	Format	Dup ok?
BLANK	1	Left	Yes
BLOCKEDR	4	Right	
CFAULTS	5	Right (blank 0s)	
CLS	n.a.		
COMMA	1	Left	Yes
CMSTACK	1	Left	
CMSZ	1	Left	
CPU	6	Right	
CR	1	<cr>	Yes
CRITICAL	1	Left	
DELTA	5	Right (blank 0s)	
DELFAULTS	5	Right (blank 0s)	
DELDFAULTS			
DELCFAULTS			
DFAULTS	5	Right (blank 0s)	
EXECMODE	2	Center	

Field	Length	Format	Dup ok?
FORKED			
INTPRI	6	Left	
JSNUM	6	Left	Yes
NAME	28	Left	
NEWPAGE	n.a.		
PERCENT	3	Right (blank 0s)	
PFT			
PID	7	Left	Yes
PIN	3	Right	Yes
PRI	3	Right	
PTYPE	4	Left	
QUEUE	1	Center	
STATE	1	Left	
SWDEPTH	1	Right (blank 0s)	
SYSCODE	1	Left	
TOTFAULTS	5	Right (blank 0s)	
TOTPERCENT	4	Right (blank 0s)	
USERACCT	17	Left	
WAIT	64	Left	
WAITPORT	6	Right (blank 0s)	

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. *Note*: Some of the SET commands do implicit ADM commands. For example, SET SWDEPTH adds the SWDEPTH field to the ADM field list.

All < All | Users >

Displays information on all active processes. The information displayed will reflect what the user has selected through the SET/RESET commands.

Break <pin#>

Suspends the process specified by **pin#**. The **pin#** can be specified in decimal (#32) or in hexadecimal ($20). Because of the advanced nature of this command, BREAKing is only allowed for SM or PM users.

CPU pin#

Reports the CPU usage for the specified process.

DEBUG

Invokes the system debugger (DEBUG). There are no parameters for this command. See the *System Debug Reference Manual* for details on DEBUG.

Delta

Display information about processes that have used the CPU since the last display.

DETERMINEPROCNAMES

DETERMINEPROCNAMES tells SHOT to erase the SHOTPIDS.DATA file and determine anew the names for "procreated" processes (e.g., "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. This command is not generally useful to users.

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. HIGHLIGHT 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.

For example, to highlight users of QUERY.PUB.SYS, type:

HIGHLIGHT PROGRAM QUERY.PUB.SYS

To highlight all users in the SALES account, type:

HIGHLIGHT USERS @.SALES

JOBINFO [NEXTJOB #] [NEXTSESSION #]

The JOBINFO command with no options tells SHOT to display job/session global information.

The NEXTJOB option allows SM users to reset the counter that MPE uses to assign the next job's number.

The NEXTSESSION option allows SM users to reset the counter that MPE uses to assign the next session's number.

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 jobs.

JS pin#

The JS command reports the Job/Session associated with a specified pin.

KILL pin#

Terminates the process.

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 "**Ctrl+Y**."

% #

Tells SHOT to only display processes that have used at least the specified percentage of the CPU since the last display.

PIN pin#

Displays information about the specified process.

Priority pin# <CS | DS | ES | BS | AS>

This sets the specified pin to the desired priority. *Note*: A process set to BS or AS priority could possibly take over the system. This is for SM or PM users only.

For example: "**PRI 0 CS**" sets SHOT priority to CS and "**P 45 ES**" sets pin 45 to priority ES

PROG

Lists all program files currently in use, along with the initial mode (either native or compatibility mode) and the capabilities. Output for this command is shown later in this chapter.

Resume pin#

Resumes a process suspended by the BREAK command. *Note*: BREAK and RESUME processes can cause system problems. RESUME is for SM or PM users only.

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 sessions.

SET | REset

The SET and RESET commands are used to specify the following options.

DELAY #seconds 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 resulting display is governed by the last Delta, ALL, or **%** command seen. *Note*: The minimum delay is 2 seconds. For example, **DELAY 0** disables the timed out reads (Default is 0):

SET DELAY 10

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.

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.

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.

DFAULTS: The **Dflts** column shows the total number of data page faults for each process. A "data page fault" was an attempt to access data on a virtual page, but the page was not in memory.

DELFAULTS: The **DelFl** column shows the number of code and data page faults for each process since the last display. ("Del" is short for "Delta.")

TOTFAULTS: The **TotFl** column shows the total number of code and data page faults for each process.

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.

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.

JSnum When JSNUM is on, the process display will include job and session numbers.

MOST Setting MOST is equivalent to doing: SET hexpins, jobstep, jsnum, showptype, swdepth, unknown.

ONE <char> For interactive runs, the ONE option (on by default) tells SHOT to do single-character reads at the prompt. The NOONE option makes the prompt do a bigger read, so an entire command line can be entered at once. *Note*: RESET ONE and SET NOONE are equivalent.

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's standard signal, message, and interrupt ports.

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 XL.

PTYPE The PTYPE option tells SHOT what "kind" of processes should be shown in the process display. Every process has a process type, which is either **detach, system, task, ucop, main**, or **user**:

PTYPE [=] [+ | -]<ALL | DETach | NONE | SYStem | TASK | UCOp | USErs>

SET 132/SET 80 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.

BIRTHS
DEATHS Normally, when SHOT notices that a process has terminated or that a new process has been born, it will not volunteer that information. SET showBIRTHS tells SHOT to report on new process activity. SET showDEATHS tells SHOT to do the same for terminated processes.

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.

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 labeled "S." A blank denotes a switch depth of "0." *Note*: This does not show the cumulative number of switches documented since the process started.

TOTPERCENT When TOTPERCENT is on, the process display will show the total CPU usage percentage since each process was first seen by SHOT.

UNKNOWN When UNKNOWN is on, SHOT will report processes that are in the unknown state if they use CPU time. This is a relatively infrequent combination, but MPE/iX has a small timing window that causes this to occur now and then.

SIRS

The SIRS command reports what SIRS (System Internal Resources) are locked (if any), by whom, and the list of waiting processes.

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 displays USM Table information.

TP pin#

The TP (Tree Process time) command prints the accumulated CPU time for the entire process tree starting at the specified pin.

Trace pin#

The TRACE command tells SHOT to print a stack trace for the specified pin.

Tree pin# [depth#]

Displays a "tree" of process information, with the specified process as the root. Using a **pin#** of "1" will display the entire system's process tree. The **depth#** option, if specified, tells SHOT to limit the "depth" of the tree to the value specified. For example:

TREE 1	the tree of all processes
TREE 1, 2	pin 1 and only its direct children (no grand-children)
TREE 0	process tree starting at SHOT

USER

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.

SHOT Examples

Here are some example of the SHOT tool.

```
SHOT [Delta]:
SHOT @ FRI, DEC 15, 1995,  3:25 PM

? Pin  Cpu    Process Name                Pri Q Delta  %  Ic Wait State
- ---- ------ ------------------------- --- - ----- --- -- --------------------
    3  13571 pm_cleanup                  100 B   113     nn JUNK
    5   2833 port_facility_process       152 C    18     nn pfp_port1
    7  35125 port_facility_process       152 C     7     nn pfp_port3
   28  74736 NMCONSOL.PUB.SYS            149 B    15     Nn MsgPort#-161
   63   1540 VTSERVER.NET.SYS            152 C   124     Nn TERMINAL_READ_WAIT
   67   3890 QEDIT.PUB.ROBELLE           152 C   840   2 Nn TERMINAL_READ_WAIT
   70   1573 VTSERVER.NET.SYS            152 C     8     Nn TERMINAL_READ_WAIT
   73   1596 SHOT.PUB.LPSTOOLS           189 C    81     Nn (executing)

# PINs: 73
(# processes recently started: 3; # since bootup: 415)
CPU status: busy 11%, idle 89% (3657 CPU out of 31215)
# Page faults: 6 Code, 134 Data.  (Overall faults/second: 17)

(Warning: MI on)

SHOT [Delta]:
```

Figure 14.1- 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 starting with pin#1 (PROGEN) in ascending order to the highest pin.

```
SHOT [Delta]: all
SHOT @ FRI, DEC 15, 1995,  3:25 PM

? Pin  Cpu    Process Name                Pri Q Delta  %  Ic Wait State
- --- ------ --------------------------- --- - ------ --- -- --------------------
    1  18155 PROGEN.PUB.SYS               13 A           Nn PROGEN_GLOBAL_PORT
    2   5133 LOAD.PUB.SYS                142 B           On JUNK
    3  13663 pm_cleanup                  100 B     92    nn JUNK
    4    263 port_facility_process        13 A           nn PFP_PORT0
    5   2833 port_facility_process       152 C           nn pfp_port1
    6    476 port_facility_process       152 C           nn pfp_port2
    7  35129 port_facility_process       152 C      4    nn pfp_port3
    8    138 port_purger_process          13 C           nn DELAYED_PURGER_PORT
    9    272 fsipc_post_timeout           13 C           nn BIPC_TIMER_PORT
   10    141 xm_checkpoint_server         30 A           nn XM_CHECK_PT_PORT
   11    638 xm_static_checkpoint_serve   30 C           nn XM
   12     20 xm_static_checkpoint_serve   30 C           nn XM
   13     20 xm_static_checkpoint_serve   30 C           nn XM
   14     20 xm_static_checkpoint_serve   30 C           nn XM
   15     93 spsnet_process_init         100 B           nn SPUNET_PORT
   16     32 repeater_process_init        13 C           nn REPEATER_PORT
   17     50 io_mgr_process               13 A           nn MsgPort#-111
   18     87 io_mgr_process               13 A           nn MsgPort#-115
   19    230 io_mgr_process               13 A           nn MsgPort#-119
   20   2790 avr_process_init            100 B           nn AVR_PORT
   21     59 mms_process_init            100 B           nn MMS_PORT
   22   2775 cm: NMMON                   149 B           cn cm_MESSAGE
   23    633 cm: MESSENGER               120 B           cn IO, TIMER, cm_MESSAG
   24   3615 NMFILE.PUB.SYS              149 B           Cn cm_MESSAGE
   25    917 NMLOGMON.PUB.SYS            149 B           Nn MsgPort#-151
   26     32 NMLOGICS.PUB.SYS            148 B           Nn NMS_LOG_GLOBAL_PORT
   27    292 NMTRCMON.PUB.SYS            149 B           Cn cm_MESSAGE
   28  74746 NMCONSOL.PUB.SYS            149 B     10    Nn MsgPort#-161
   29   2315 io_mgr_process              13 A           nn MsgPort#-166
   30     59 io_mgr_process              13 A           nn MsgPort#-170
   31    111 io_mgr_process              13 A           nn MsgPort#-174
   32    108 io_mgr_process              13 A           nn MsgPort#-179
   33     52 io_mgr_process              13 A           nn MsgPort#-183
   34     95 io_mgr_process              13 A           nn MsgPort#-187
   35     99 io_mgr_process              13 A           nn MsgPort#-203
   36     51 io_mgr_process              13 A           nn MsgPort#-219
   37   1406 PSMON.PRED.SYS              152 C           Pn TERMINAL_READ_WAIT
   38    287 LOG.PUB.SYS                  50 B           Nn SYSLOG
   39     73 SYSMAIN.PUB.SYS              49 B           Nn SYSMAIN_PORT
   40  14817 SOS.PUB.LPSTST              100 B           Nn TIMER
   41    139 SPOOLMOM.PUB.SYS            100 B           Nn SPOOLER_MOM_PORT
   42   5516 MEMLOGP.DIAG.SYS            152 C           Nn TERMINAL_READ_WAIT
   43   2529 SESSION.PUB.SYS             100 B           Nn SESSIONMAIN_PORT
   44  22893 JOB.PUB.SYS                 100 B           Nn 2 ports@$21c.4163896
                waiting on: JOBMAIN_PORT, JOB_QUEUE_PORT
   45    327 JSMAIN.PUB.SYS               152 B           Nn MsgPort#-32771
   46   1431 CI.PUB.SYS                   152 C           Nn SON, FATHER
   47     94 OUTSPOOL.PUB.SYS             100 B           Nn OTHER_IO_WAIT
   50   2255 DIAGMON.DIAG.SYS             13 A           Nn TERMINAL_READ_WAIT
   51     49 ICMPSERV.NET.SYS             149 C           Nn MEMORY_WAIT
   52    389 JSMAIN.PUB.SYS               152 B           Nn MsgPort#-32776
   53   2714 NETCP.NET.SYS                149 B           Nn DATA_COMM_WAIT
   54    119 TCPSIP.NET.SYS               149 B           Nn MsgPort#-279
   55    294 SOCKREG.NET.SYS              149 B           Cn TERMINAL_READ_WAIT
   56     31 PT2PNSTN.NET.SYS             149 B           Nn SigPort#-32780
   57   3506 DSDAD.NET.SYS                149 B           Nn TERMINAL_READ_WAIT
   58    716 CI.PUB.SYS                   202 D           Nn SON, FATHER
```

```
   59     346 JSMAIN.PUB.SYS              152 B            Nn MsgPort#-32781
   60     102 FTPMON.ARPA.SYS             152 C            Nn OTHER_WAIT
   61     691 HTTPD.BIN.WWW               202 D            Pn OTHER_WAIT
   62     631 JSMAIN.PUB.SYS              152 B            Nn MsgPort#-32785
   63    1540 VTSERVER.NET.SYS            152 C            Nn TERMINAL_READ_WAIT
   64      21 JSMAIN.PUB.SYS              152 B            Nn MsgPort#-32789
   65    7230 CI.PUB.SYS                  152 C            Nn SON, FATHER
   66    2169 HTTPBACK.BIN.WWW            202 D            Pn cm_MESSAGE
   67    3890 QEDIT.PUB.ROBELLE           152 C            Nn TERMINAL_READ_WAIT
   70    1580 VTSERVER.NET.SYS            152 C      7     Nn TERMINAL_READ_WAIT
   71    1413 CI.PUB.SYS                  152 C            Nn TERMINAL_READ_WAIT
   72     527 JSMAIN.PUB.SYS              152 B            Nn MsgPort#-32788
   73    1749 SHOT.PUB.LPSTOOLS           152 C    153     Nn (executing)
   74     339 JSMAIN.PUB.SYS              152 B            Nn MsgPort#-32798
   75     344 JSMAIN.PUB.SYS              152 B            Nn MsgPort#-32804
   77     472 CI.PUB.SYS                  202 D            Nn SON, FATHER

# PINs: 72
(# processes recently started: -44; # since bootup: 371)
CPU status: busy 1%, idle 99% (266 CPU out of 26409)
# Page faults: 1 Code, 2 Data.  (Overall faults/second: 10)

(Warning: MI on)

SHOT [All]:
```

Figure 14.2 - ALL Command

This is an example that shows the use of restriction. In this case, the SHOT display is for jobs only.

```
SHOT [All]: jobs only
SHOT @ FRI, DEC 15, 1995,  3:27 PM

? Pin  Cpu    Process Name               Pri Q Delta  %  Ic Wait State
- --- ------ -------------------------- --- - ------ --- -- --------------------
#J2   FTPMON,FTP.SYS, ARPA, total CPU = 0 seconds, ldev 10, jsmain 52.
   Step: RUN ftpmon
    58    716 CI.PUB.SYS                202 D            Nn SON, FATHER
     60    102 FTPMON.ARPA.SYS           152 C            Nn OTHER_WAIT

#J10  SOSMONJ,MGR.LPSTST, PUB, total CPU = 15 seconds, ldev 10, jsmain 59.
   Step: RUN sos.pub.lpstst
    77    472 CI.PUB.SYS                202 D            Nn SON, FATHER
     40  14817 SOS.PUB.LPSTST           100 B            Nn TIMER


# PINs: 72 (# processes since bootup: 371)
CPU status: busy 1%, idle 99% (275 CPU out of 16010)
# Page faults: 1 Code, 8 Data.  (Overall faults/second: 5)

(Warning: MI on)

SHOT [Jobs]:
```

Figure 14.3 - Restricted SHOT Display

This is a typical SHOT "sessions only" display.

```
SHOT [Jobs]: sessions only
SHOT @ FRI, DEC 15, 1995,  3:27 PM

? Pin  Cpu    Process Name               Pri Q Delta  %  Ic Wait State
- --- ------ -------------------------- --- - ------ --- -- --------------------
#S12  YVONNE,MGR.PRODTAPE, PUB, total CPU = 2 seconds, ldev 20, jsmain 45.
   Step: PRINTO 343
    71   1413 CI.PUB.SYS                152 C            Nn TERMINAL_READ_WAIT
```

```
#S32  MICHAEL,MGR.HENSLEY, PUB, total CPU = 43 seconds, ldev 17, jsmain 74.
   Step: SHOT
    65   7230 CI.PUB.SYS                  152 C           Nn SON, FATHER
     73   3197 SHOT.PUB.LPSTOOLS          172 C   200   1 Nn (executing)
    70   1703 VTSERVER.NET.SYS            152 C    22     Nn TERMINAL_READ_WAIT

#S31  MICHAEL,MGR.LPSTOOLS, PUB, total CPU = 10 seconds, ldev 18, jsmain 75.
   Step: QEDIT
    46   1431 CI.PUB.SYS                  152 C           Nn SON, FATHER
    67   3890 QEDIT.PUB.ROBELLE           152 C           Nn TERMINAL_READ_WAIT
    63   1540 VTSERVER.NET.SYS            152 C           Nn TERMINAL_READ_WAIT


# PINs: 72 (# processes since bootup: 371)
CPU status: busy 1%, idle 99% (230 CPU out of 12659)
# Page faults: none.  (Overall faults/second: 4)

(Warning: MI on)

SHOT [Jobs]:
```

Figure 14.4 - Sessions-only SHOT Display

In this example, 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.0014d70c enable_int+$2c
NM* 0) SP=41846c28 RP=a.0027aa5c notify_dispatcher.block_current_process+$2f0
NM  1) SP=41846c28 RP=a.0027cea8 notify_dispatcher+$264
NM  2) SP=41846ba8 RP=a.003525e0 ipc_impede+$274
NM  3) SP=41846aa8 RP=a.00352358 ?ipc_impede+$8
         export stub: a.0178f30c sk_block+$1b4
NM  4) SP=41846968 RP=a.0178fbb4 sk_block_for_completion+$fc
NM  5) SP=41846828 RP=a.017b4f24 sk_accept.wait_for_passive_open_request+$8c
NM  6) SP=418466e8 RP=a.017b520c sk_accept+$1f8
NM  7) SP=41846668 RP=a.0176f750 IPCRECVCN+$2d4
NM  8) SP=41846568 RP=a.0176f468 ?IPCRECVCN+$8
         export stub: 364.00007f9c
NM  9) SP=418462e8 RP=364.00000000
     (end of NM stack)

SHOT [Jobs]:
```

Figure 14.5 - 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 @ FRI, DEC 15, 1995,  3:29 PM

? Pin  Cpu    Process Name                Pri Q Delta  %  Ic Wait State
- ---- ------ --------------------------- --- - ----- --- -- --------------------
     1  18155 PROGEN.PUB.SYS               13 A           Nn PROGEN_GLOBAL_PORT
     2   5133 LOAD.PUB.SYS                142 B           On JUNK
     3  13663 pm_cleanup                   100 B           nn JUNK
     4    266 port_facility_process         13 A     3     nn PFP_PORT0
     5   2833 port_facility_process        152 C           nn pfp_port1
     6    476 port_facility_process        152 C           nn pfp_port2
     7  35167 port_facility_process        152 C     9     nn pfp_port3
     8    138 port_purger_process           13 C           nn DELAYED_PURGER_PORT
     9    273 fsipc_post_timeout            13 C           nn BIPC_TIMER_PORT
    10    141 xm_checkpoint_server          30 A           nn XM_CHECK_PT_PORT
    11    638 xm_static_checkpoint_se       30 C           nn XM
```

```
    12      20 xm_static_checkpoint_se     30 C            nn XM
    13      20 xm_static_checkpoint_se     30 C            nn XM
    14      20 xm_static_checkpoint_se     30 C            nn XM
    15      93 spsnet_process_init        100 B            nn SPUNET_PORT
    16      32 repeater_process_init       13 C            nn REPEATER_PORT
    17      50 io_mgr_process              13 A            nn MsgPort#-111
    18      87 io_mgr_process              13 A            nn MsgPort#-115
    19     230 io_mgr_process              13 A            nn MsgPort#-119
    20    2790 avr_process_init           100 B            nn AVR_PORT
    21      59 mms_process_init           100 B            nn MMS_PORT
    22    2775 cm: NMMON                  149 B            cn cm_MESSAGE
    24    3615 NMFILE.PUB.SYS             149 B            Cn cm_MESSAGE
    25     917 NMLOGMON.PUB.SYS           149 B            Nn MsgPort#-151
    26      32 NMLOGICS.PUB.SYS           148 B            Nn NMS_LOG_GLOBAL_PORT
    27     292 NMTRCMON.PUB.SYS           149 B            Cn cm_MESSAGE
    28   74827 NMCONSOL.PUB.SYS           149 B    19      Nn MsgPort#-161
    53    2714 NETCP.NET.SYS              149 B            Nn DATA_COMM_WAIT
    51      49 ICMPSERV.NET.SYS           149 C            Nn MEMORY_WAIT
    55     294 SOCKREG.NET.SYS            149 B            Cn TERMINAL_READ_WAIT
    56      31 PT2PNSTN.NET.SYS           149 B            Nn SigPort#-32780
    54     119 TCPSIP.NET.SYS             149 B            Nn MsgPort#-279
    57    3506 DSDAD.NET.SYS              149 B            Nn TERMINAL_READ_WAIT
    23     633 cm:MESSENGER               120 B           cn IO, TIMER, cm_MESSAG
    29    2315 io_mgr_process             13 A            nn MsgPort#-166
    30      59 io_mgr_process             13 A            nn MsgPort#-170
    31     111 io_mgr_process             13 A            nn MsgPort#-174
    32     108 io_mgr_process             13 A            nn MsgPort#-179
    33      52 io_mgr_process             13 A            nn MsgPort#-183
    34      95 io_mgr_process             13 A            nn MsgPort#-187
    35      99 io_mgr_process             13 A            nn MsgPort#-203
    36      51 io_mgr_process             13 A            nn MsgPort#-219
    38     287 LOG.PUB.SYS                50 B            Nn SYSLOG
    39      73 SYSMAIN.PUB.SYS            49 B            Nn SYSMAIN_PORT
    41     139 SPOOLMOM.PUB.SYS          100 B            Nn SPOOLER_MOM_PORT
    47      94 OUTSPOOL.PUB.SYS          100 B            Nn OTHER_IO_WAIT
    43    2529 SESSION.PUB.SYS           100 B            Nn SESSIONMAIN_PORT
    45     327 JSMAIN.PUB.SYS            152 B            Nn MsgPort#-32771
    71    1413 CI.PUB.SYS                152 C            Nn TERMINAL_READ_WAIT
    74     339 JSMAIN.PUB.SYS            152 B            Nn MsgPort#-32798
    65    7230 CI.PUB.SYS                152 C            Nn SON, FATHER
    73    5879 SHOT.PUB.LPSTOOLS         152 C  2378    5 Nn (executing)
    70    1764 VTSERVER.NET.SYS          152 C    46      Nn TERMINAL_READ_WAIT
    75     344 JSMAIN.PUB.SYS            152 B            Nn MsgPort#-32804
    46    1431 CI.PUB.SYS                152 C            Nn SON, FATHER
    67    3890 QEDIT.PUB.ROBELLE         152 C            Nn TERMINAL_READ_WAIT
    63    1540 VTSERVER.NET.SYS          152 C            Nn TERMINAL_READ_WAIT
    64      21 JSMAIN.PUB.SYS            152 B            Nn MsgPort#-32789
    44   22893 JOB.PUB.SYS               100 B           Nn 2 ports@$21c.4163896
                  waiting on: JOBMAIN_PORT, JOB_QUEUE_PORT
    52     389 JSMAIN.PUB.SYS            152 B            Nn MsgPort#-32776
    58     716 CI.PUB.SYS                202 D            Nn SON, FATHER
    60     102 FTPMON.ARPA.SYS           152 C            Nn OTHER_WAIT
    59     346 JSMAIN.PUB.SYS            152 B            Nn MsgPort#-32781
    77     472 CI.PUB.SYS                202 D            Nn SON, FATHER
    40   14817 SOS.PUB.LPSTST            100 B            Nn TIMER
    62     631 JSMAIN.PUB.SYS            152 B            Nn MsgPort#-32785
    72     527 JSMAIN.PUB.SYS            152 B            Nn MsgPort#-32788
    50    2255 DIAGMON.DIAG.SYS           13 A            Nn TERMINAL_READ_WAIT
    42    5516 MEMLOGP.DIAG.SYS          152 C            Nn TERMINAL_READ_WAIT
    37    1408 PSMON.PRED.SYS            152 C            Pn TERMINAL_READ_WAIT
    66    2174 HTTPBACK.BIN.WWW          202 D            Pn cm_MESSAGE
    61     691 HTTPD.BIN.WWW             202 D            Pn OTHER_WAIT
SHOT [Jobs]:
```

Figure 14.6 - TREE Command

SHOT's default display can easily be modified using the ADM command 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 @ FRI, DEC 15, 1995,  3:30 PM

Pin Process Name                Pri Q Ic S Wait State
--- --------------------------- --- - -- - ------------------------------------
  7 port_facility_process       152 C nn   pfp_port3
 28 NMCONSOL.PUB.SYS            149 B Nn   MsgPort#-161
 70 VTSERVER.NET.SYS            152 C Nn   TERMINAL_READ_WAIT
 73 SHOT.PUB.LPSTOOLS           154 C Nn   (executing)

# PINs: 72 (# processes since bootup: 371)
CPU status: busy 0%, idle 100% (148 CPU out of 18918)
# Page faults: none.  (Overall faults/second: 2)

(Warning: MI on)

SHOT [Delta]:
```

Figure 14.7 - ADM Command

The SWITCHDEPTH column (headed with the letter S) indicates how many times a process has switched between NM and CM.

```
SHOT [All]:

SHOT @ FRI, DEC 15, 1995,  3:31 PM

Pin Process Name                Pri Q Ic S Wait State
--- --------------------------- --- - -- - ------------------------------------
  2 LOAD.PUB.SYS                142 B On 1 JUNK
 23 cm: MESSENGER               120 B cn 1 IO, TIMER, cm_MESSAGE
 46 CI.PUB.SYS                  152 C Nn 2 SON, FATHER
 47 OUTSPOOL.PUB.SYS            100 B Nn 2 OTHER_IO_WAIT
 58 CI.PUB.SYS                  202 D Nn 2 SON, FATHER
 67 QEDIT.PUB.ROBELLE           152 C Nn 2 TERMINAL_READ_WAIT
 71 CI.PUB.SYS                  152 C Nn 2 TERMINAL_READ_WAIT
 77 CI.PUB.SYS                  202 D Nn 2 SON, FATHER

# PINs: 72 (# processes since bootup: 371)
CPU status: busy 2%, idle 98% (118 CPU out of 5274)
# Page faults: none.  (Overall faults/second: 2)

(Warning: MI on)

SHOT [All]:
```

Figure 14.8 - SWITCHDEPTH Column

Note: Program optimization on an MPE/iX machine often involves migrating from CM to NM. SHOT's SWDEPTH option can be used to determine how many times your application is switching between CM and NM. Excessive switching can result in significant performance penalties.

This is an example that shows the use of the PROG command.

```
SHOT [All]: prog

Building list of programs...sorting...

#Procs   Program Name                    Capabilities
------   ----------------------------- -------------
     5 : CI.PUB.SYS                     NM:
     1 : DIAGMON.DIAG.SYS               NM:
     1 : DSDAD.NET.SYS                  NM:
     1 : FTPMON.ARPA.SYS                NM:
     1 : HTTPBACK.BIN.WWW               NM:
     1 : HTTPD.BIN.WWW                  NM:
     1 : ICMPSERV.NET.SYS               NM:
     1 : JOB.PUB.SYS                    NM:
     8 : JSMAIN.PUB.SYS                 NM:
     1 : LOAD.PUB.SYS                   OCT: PrivSeg
     1 : LOG.PUB.SYS                    NM:
     1 : MEMLOGP.DIAG.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
     1 : OUTSPOOL.PUB.SYS               NM:
     1 : PROGEN.PUB.SYS                 (could not open, error 52)
     1 : PSMON.PRED.SYS                 NM:
     1 : PT2PNSTN.NET.SYS               NM:
     1 : QEDIT.PUB.ROBELLE              NM:
     1 : SESSION.PUB.SYS                NM:
     1 : SHOT.PUB.LPSTOOLS              NM:
     1 : SOCKREG.NET.SYS                CM:  PM ds mr ph
     1 : SOS.PUB.LPSTST                 NM:
     1 : SPOOLMOM.PUB.SYS               NM:
     1 : SYSMAIN.PUB.SYS                NM:
     1 : TCPSIP.NET.SYS                 NM:
     2 : VTSERVER.NET.SYS               NM:

SHOT [All]:
```

Figure 14.9 - PROG Command

SHOT Error Messages

Message	***DEBUG requires SM or PM capability.***
Cause	Using the DEBUG command from within SHOT requires that you have PM.
Action	Use GRANT from the ***System Managers Toolbox*** to give yourself PM, or logon as a user that has PM.
Message	***You must have SM capability to do this.***
Cause	The SHOT commands KILL, BREAK, RESUME require that a user has SM.
Action	Use GRANT from the ***System Managers Toolbox***, or logon as a user that has SM.
Message	***You must have SM or OP capability to do this.***
Cause	The SHOT command PRIORITY requires that a user have SM or OP.
Action	Use GRANT from the ***System Managers Toolbox***, or logon as a user that has either SM or OP.
Message	***You must have SM or PM capability to do this.***
Cause	The SHOT command TRACE requires that a user have SM or PM.
Action	Use GRANT from the ***System Managers Toolbox***, or logon as a user that has either SM or PM.

Chapter 15

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 report of files verified for many kinds of tape formats. TINDEX can verify NMSTORE, CMSTORE, DBLOAD, (classic) MEMDUMP, HPUX Core tapes, SPOOK, (classic) DUX, HPPA INSTALL, and (spectrum) MEMDUMP tapes. TINDEX also has limited support for UNIX TAR tapes, AIX Boot Tapes, DISCUTIL tape recognition, CPIO support, and RS6000 Backup 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 so 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 (e.g., **/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

Detailed information about each file is reported to a spoolfile 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=mylist	Change the file designator

Usage

TINDEX can be run from either the supplied UDC or via a fully-qualified statement.

- UDC
  :TINDEX [tapename [option1 option2 ...]]
- 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.

Option	Description
132 (*256*)*	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
CHECKSUM	Displays checksum data
COMPARE (*64*)	Compares modification date
CONTENTS	Dumps tape directory to disk file
CORETAR	Tells TINDEX to treat the tape as core TAR format
CPIOSUMMARY	Displays the CPIO Summary
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
DOCDOT	Displays information about the progress indicator
EOF	Displays EOF
EXPLAIN	Displays extra information about long creator names
EXTENTS	Displays extents
FCODE	Displays filecode
FGA	Displays filename.group.account
FILENUM	Displays filenumber
FLAGCREATOR	Displays account creator information
FORCEMULTI	Forces multiple reel
FULLQUICK	Displays directory only, one filename per line
HEADER	Displays header at the top of each page
HELP	Displays on-line help
HFSONLY	Displays HFS file information only
LABELLED (*8*)	Labeled tape
LAND132	132 character landscape output
LAND176	176 character landscape output
LASTEXTENT	Displays last extent information
LIMIT	Displays file limit
LOCKWORD	Displays lockwords
LONG (*1*)	More detailed output
MATRIX	Displays file security matrix
MINIMUM	Minimum output
MODIFIED	Displays modification date
NEWDISK (*4*)	Outputs to disk instead of printer
NOCORE	Opposite effect of the CORETAR option

Option	Description
NOTHING	Only displays filename.group.account
ONLINE	Puts tape drive on-line
PAGESIZE	Changes output page size
PORT132 (*256*)	132 character portrait output
PV	Synonym for BUILDPV
QUICK (*2*)	Directory listing only
REARM	Periodically rearms control-y
RECSZT	Displays record size
RESTOREQUICK	Produces a RESTORE compatible listing
SECTORS	Displays number of sectors
SECURITY	Displays file security matrix
SHORTCORE	Tells TINDEX to treat the tape as a short core format
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 (*16*)	Use fcontrol - forward skip file
SKIPHFS (*16*)	Skips HFS file information
TAPECONT	Dumps tape directory to disk
TAR	Forces TINDEX to treat the tape as a TAR format
TARSUMMARY	Displays TAR tape summary
TRYNM	Tells TINDEX to run STORE to read the tape
TRYXC	Tells TINDEX to try to process 7980XC compressed tapes
TYPE	Displays file type information
USERLABELS	Displays user label information
VERIFY (*32*)	Verifies file label and file data

*Numbers within the parentheses and in italic typeface are Parm bit values that you can set to select an option.

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 ([ ]).

[NO]ACCESSED

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. The default is ACCESSED.

[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. If your STORE tape has files in alphabetical 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. The default is NOALPHASORT.

[NO]BLKSZ

BLKSZ tells TINDEX that you want to see the block size for every file. NOBLKSZ tells TINDEX to suppress showing the block size. The default is BLKSZ.

Note: BLKSZ may not be visible on "narrow" output.

[NO]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
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. The default is NOBUILDPV.

[NO]CHECKSUM

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. This option tries to skip those fields in a file label that might change without the underlying file's data changing (e.g., access date, ldev). It also 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).

Note: Due to a flaw in the implementation of "TRANSPORT" mode STORE on MPE/iX, the checksums generated may vary from STORE to STORE, even if the files have not been modified. The default is NOCHECKSUM.

[NO]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 equal sign (=) 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:

Character	Description
<	older than disk file
>	newer than disk file
=	same date as disk file
blank	disk file version does not exist
*	disk file exists, could not compare
?	error occurred fetching disk file info.

The default is NOCOMPARE.

CONTents < <eol> | tapecontname >

This option causes TINDEX to save a copy of the undecoded directory on disk in a file whose name follows the CONT option. The following example will create a text file called "DIRCOPY":

RUN TINDEX.PUB.LPSTOOLS;INFO="fulldump CONT dircopy"

This file cannot be read without special tools.

[NO]CORETAR

This option tells TINDEX to treat core tapes as if they are TAR format tapes.

[NO]CPIOSUMMARY

This option tells TINDEX to only display a summary of tape information for CPIO tapes. *Note*: CPIO is a UNIX copy Tool.

[NO]CREATED

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. The default is CREATED.

[NO]CREATOR

CREATOR tells TINDEX to report the creator for each file (if available). NOCREATOR suppresses this column. The default is CREATOR.

[NO]DATES

DATES is a synonym for ACCESSED, CREATED, MODIFIED. NODATES is a synonym for NOACCESSED, NOCREATED, NOMODIFIED. The default is DATES.

[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

DEVICE tells TINDEX that you want to see the device name for every file. NODEVICE tells TINDEX to suppress showing the device name. The default is DEVICE.

Note: Device names of "DISC" are automatically replaced by 8 blanks. Also, DEVICE may not be visible on "narrow" output.

[NO]DOCDOT

This option describes the progress indicator in terms of units where units are files, sectors, or whatever measurement is being used to indicate progress. The progress indicator may vary depending on the tape format you use.

[NO]EOF

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. The default is EOF. *Note*: EOF may not be visible on "narrow" output.

[NO]EXPLAIN

EXPLAIN tells TINDEX to display the text that is generated when it encounters long creator names. NOEXPLAIN tells TINDEX to suppress the text that is normally displayed when it encounters long creator names. The default is EXPLAIN.

[NO]EXTENTS

EXTENTS tells TINDEX that you want to see the number of extents for every file (if available). NOEXTENTS tells TINDEX to suppress showing the number extents. The default is EXTENTS.

Note: EXTENTS may not be visible on "narrow" output.

[NO]FCODE

FCODE tells TINDEX that you want to see the file code for every file. NOFCODE tells TINDEX to suppress showing the block size. The default is FCODE.

Note: Lowercase 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

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**." The default is FGA.

Note: It might not be very useful to suppress FGA!

[NO]FILENUM

FILENUM tells TINDEX to report the file number of each file it finds. NOFILENUM suppresses this column. The default is FILENUM.

[NO]FLAGCREATOR

FLAGCREATOR tells TINDEX to display account creator information. The default is FLAGCREATOR.

[NO]FORCEMULTI

FORCEMULTI tells TINDEX to force a multiple reel operation. The default is NOFORCEMULTI.

[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, but 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.

[NO]HEADER

HEADER tells TINDEX to produce a short header at the top of every page of output. NOHEADER disables the page headers. The default is HEADER.

HELP

The HELP options displays the TINDEX Help facility.

[NO]HFSONLY

This option tells TINDEX to only display HFS file information.

[NO]LABelled

The LABELLED option tells TINDEX to make a special effort to read a labeled STORE tape. Normally, to get a directory listing of a labeled 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 unlabelled tape for Read & Write access, which allows a labeled tape to be "sneaked" past MPE. Don't worry, TINDEX will NOT write to the tape!

If a labeled tape has a lockword, you will be asked to supply it, regardless of your capabilities.

[NO]LAND132

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

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

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. The default is LASTEXTENT.

Note: LASTEXTENT may not be visible on "narrow" output.

[NO]LIMIT

LIMIT tells TINDEX that you want to see the file limit for every file. NOLIMIT tells TINDEX to suppress showing the file limit. The default is LIMIT.

Note: LIMIT may not be visible on "narrow" output.

[NO]LOCKword

LOCKWORD tells TINDEX to report the lockword for each file (if available, and if appropriate for your capabilities). NOLOCKWORD suppresses this column. The default is LOCKWORD.

LONG

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). For 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

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. SECURITY is a synonym for MATRIX. The default is 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. The default is NOMINIMUM.

[NO]MODIFIed

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. The default is MODIFIED.

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

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. Thus you can use NOTHING and then turn on selected options. NONOTHING turns on all REPORT options.

[NO]ONLINE [=] ldev#

ONLINE tells TINDEX that you want to 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 [=]

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 is 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., try specifying "PAGE=58" or "PAGE=57."

[NO]PORT132

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

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 **Ctrl+Y** trap. This option should be unnecessary, but can be useful when running on a version of MPE XL that has problems "losing" **Ctrl+Y**.

[NO]RECSZT

RECSZT tells TINDEX that you want to see the record size of every file. NORECSZT tells TINDEX to suppress this information. The default is RECSZT. *Note*: RECSIZE is a synonym for RECSZT. Also, 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.

[NO]SECTORS

SECTORS tells TINDEX that you want to see the number of sectors each file occupies. NOSECTORS tells TINDEX to suppress this information. The default is SECTORS.

Note: SECTORS may not be visible on "narrow" output.

[NO]SECURITY

SECURITY is a synonym for MATRIX. See the MATRIX command for more information.

[NO]SHORTCORE

This option tells TINDEX to use short form output for core tapes.

[NO]SHOWNEW

SHOWNEW* tells TINDEX that you only want to see those files on tape that are newer than diskfiles 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 have the same date as disk files with the same name.

**Note*: 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 these options (SHOWNEW, SHOWOLD or 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.

SKIPHFS

The SKIPHFS option tells TINDEX not to display HFS file information.

TAPECONT diskname

The TAPECONT option causes TINDEX to save a copy of the undecoded 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 read without special tools.

[NO]TAR

Using this option causes TINDEX to treat the tape as a TAR format tape.

[NO]TARSUMMARY

This option is used to display a summary for TAR tapes.

[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. The default is NOTRYNM.

[NO]TRYXC

The TRYXC 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. The default is NOTRYXC.

[NO]TYPE

TYPE tells TINDEX that you want to see the file type information for every file. NOTYPE tells TINDEX to suppress the file type information. The default is TYPE. "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

USERLABELS tells TINDEX that you want to see the user label information for every file. NOUSERLABELS tells TINDEX to suppress showing user label information. The default is USERLABELS.

Note: USERLABELS may not be visible on "narrow" output.

[NO]VERify

The VERIFY option tells TINDEX to check that every bit on the tape is readable. The default is VERIFY. 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:

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

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*


Please mount reel # 1 for TAPE LPSTAPE.

(Printer file has 132 characters per line)
Will VERIFY by reading every tape record.


NM STORE:
   Created              : FRI, DEC 15, 1995,  3:35 PM
   Options: recovery; show; fastsearch; copyacd; No_Compression

There are 90 files on your tape set.
Looking for start of first user file...
Starting to read file labels now...

Tape created: 12/15/95 @  3:35 PM

Will print one dot (.) per 10 files: .........
***END OF TAPE SET***
```

```
Largest file: #78 = (5,168 sectors (1 MBs)

Total # of sectors on tape:  42,496 (10 MBs); total # of files: 90

VERIFY --> no errors
:showout

DEV/CL    DFID        JOBNUM   FNAME     STATE FRM SPACE RANK PRI #C
LP        #O347       #S32     LPSLP     READY          48    D 8   1
17        #O17        #S32     $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:  128 SECTORS
OUTFENCE = 8
OUTFENCE = 13 FOR LDEV 6
:printo 347

 NM STORE:
    Created              : FRI, DEC 15, 1995,  3:35 PM
    Options: recovery; show; fastsearch; copyacd; No_Compression

 There are 90 files on your tape set.
 TINDEX (2.2) - LPS Toolbox (A.01a)        (c) 1995 Lund Performance Solutions


 NM STORE Tape: LPSTAPE, Ldev 7 (DDS), Created: 12/15/95 @  3:35 PM;
                                       Reel # 1  Page 1

 FILE     .GROUP    .ACCOUNT  CREATED  MODIFIED ACCESSED CREATOR  LOCKWORD
 ------------------------- -------- -------- -------- -------- --------
 LPSINST .JOB      +PRODTAPE 26 Jun95 26 Jun95 10 Oct95 MGR

 INSTOS   .PUBSYS  +PRODTAPE 26 Jun95 26 Jun95 30 Nov95 MGR
 INSTMI   .PUBSYS  +PRODTAPE 26 Jun95 26 Jun95 30 Nov95 MGR

 HOLIDAYS.PUB      +PRODTAPE 26 Jun95 26 Jun95 06 Oct95 MGR
 ITEMLIST.PUB      +PRODTAPE 26 Jun95 26 Jun95 06 Oct95 MGR
 SOS      .PUB     +PRODTAPE 26 Jun95 26 Jun95 14 Nov95 MGR
 SOSKIP   .PUB     +PRODTAPE 26 Jun95 26 Jun95 06 Oct95 MGR
 LPSCFG   .PUB     +PRODTAPE 15 Dec95 15 Dec95 15 Dec95 MGR
 LPSCHECK.PUB      +PRODTAPE 26 Jun95 26 Jun95 06 Oct95 MGR
 LPSEXTND.PUB      +PRODTAPE 26 Jun95 26 Jun95 06 Oct95 MGR
 LOGHELP .PUB      +PRODTAPE 26 Jun95 26 Jun95 06 Oct95 MGR
 PFGITEMS.PUB      +PRODTAPE 26 Jun95 26 Jun95 06 Oct95 MGR
 RCITEMS .PUB      +PRODTAPE 26 Jun95 26 Jun95 06 Oct95 MGR
 REDITEMS.PUB      +PRODTAPE 26 Jun95 26 Jun95 06 Oct95 MGR
 REPRTDEF.PUB      +PRODTAPE 26 Jun95 26 Jun95 06 Oct95 MGR
 SL       .PUB     +PRODTAPE 26 Jun95 26 Jun95 06 Oct95 MGR
 SL94000A.PUB      +PRODTAPE 26 Jun95 26 Jun95 30 Nov95 MGR
 SOSADVIC.PUB      +PRODTAPE 26 Jun95 26 Jun95 06 Oct95 MGR
 SOSFULL .PUB      +PRODTAPE 26 Jun95 26 Jun95 06 Oct95 MGR
 SOSGRAPH.PUB      +PRODTAPE 26 Jun95 26 Jun95 06 Oct95 MGR
 SOSHELP .PUB      +PRODTAPE 26 Jun95 26 Jun95 06 Oct95 MGR
 SOSJOB   .PUB     +PRODTAPE 26 Jun95 26 Jun95 06 Oct95 MGR
 SOSLOGX .PUB      +PRODTAPE 26 Jun95 26 Jun95 06 Oct95 MGR
 SOSLOGXJ.PUB      +PRODTAPE 26 Jun95 26 Jun95 06 Oct95 MGR
 SOSMONJ .PUB      +PRODTAPE 26 Jun95 26 Jun95 02 Nov95 MGR
 SOSPRANJ.PUB      +PRODTAPE 26 Jun95 26 Jun95 06 Oct95 MGR
 RPTCARDJ.PUB      +PRODTAPE 22 Sep95 22 Sep95 09 Oct95 MGR
 PERFCOLJ.PUB      +PRODTAPE 16 Aug95 16 Aug95 14 Nov95 MGR
 SOSPRANL.PUB      +PRODTAPE 26 Jun95 26 Jun95 07 Nov95 MGR
 SOSPRDMP.PUB      +PRODTAPE 26 Jun95 26 Jun95 06 Oct95 MGR
```

```
SOSRCOM .PUB      +PRODTAPE 26 Jun95 26 Jun95 06 Oct95 MGR
SOSSNOOP.PUB      +PRODTAPE 26 Jun95 26 Jun95 06 Oct95 MGR
XL      .PUB      +PRODTAPE 26 Jun95 26 Jun95 06 Oct95 MGR

ISFONT  .CHART    +PRODTAPE 26 Jun95 26 Jun95 30 Nov95 MGR
ISROOT  .CHART    +PRODTAPE 26 Jun95 26 Jun95 30 Nov95 MGR
SOSCHART.CHART    +PRODTAPE 26 Jun95 26 Jun95 30 Nov95 MGR

CPUUDEX1.GRAPH    +PRODTAPE 26 Jun95 26 Jun95 06 Oct95 MGR
CPUUTIL1.GRAPH    +PRODTAPE 26 Jun95 26 Jun95 06 Oct95 MGR
CPUUTIL2.GRAPH    +PRODTAPE 26 Jun95 26 Jun95 06 Oct95 MGR
CPUUTIL3.GRAPH    +PRODTAPE 26 Jun95 26 Jun95 06 Oct95 MGR
CPUUTIL4.GRAPH    +PRODTAPE 26 Jun95 26 Jun95 06 Oct95 MGR
DISCIO1 .GRAPH    +PRODTAPE 26 Jun95 26 Jun95 06 Oct95 MGR
DISCIO2 .GRAPH    +PRODTAPE 26 Jun95 26 Jun95 06 Oct95 MGR
Total # of sectors on page:  22,272 (  5 MBs)

TINDEX [2.2] - LPS Toolbox [A.01a]        (c) 1995 Lund Performance Solutions

NM STORE Tape: LPSTAPE, Ldev 7 (DDS), Created: 12/15/95 @  3:35 PM;
                                    Reel # 1  Page 2

FILE    .GROUP    .ACCOUNT  CREATED  MODIFIED ACCESSED CREATOR  LOCKWORD
------------------------- -------- -------- -------- -------- --------
GRAPHCAT.GRAPH    +PRODTAPE 26 Jun95 26 Jun95 06 Oct95 MGR
GRAPHDEV.GRAPH    +PRODTAPE 26 Jun95 26 Jun95 06 Oct95 MGR
MEMUTIL1.GRAPH    +PRODTAPE 26 Jun95 26 Jun95 06 Oct95 MGR
MODESWI1.GRAPH    +PRODTAPE 26 Jun95 26 Jun95 06 Oct95 MGR
RESPTIM1.GRAPH    +PRODTAPE 26 Jun95 26 Jun95 06 Oct95 MGR
TRANMGR1.GRAPH    +PRODTAPE 26 Jun95 26 Jun95 06 Oct95 MGR

ANALRPT .SAMPLE   +PRODTAPE 26 Jun95 26 Jun95 06 Oct95 MGR
CPUUTIL .SAMPLE   +PRODTAPE 26 Jun95 26 Jun95 06 Oct95 MGR
DISCFREE.SAMPLE   +PRODTAPE 26 Jun95 26 Jun95 06 Oct95 MGR
DISCINFO.SAMPLE   +PRODTAPE 26 Jun95 26 Jun95 06 Oct95 MGR
DISCUTIL.SAMPLE   +PRODTAPE 26 Jun95 26 Jun95 06 Oct95 MGR
DISCVOL .SAMPLE   +PRODTAPE 26 Jun95 26 Jun95 06 Oct95 MGR
GLOBAL  .SAMPLE   +PRODTAPE 26 Jun95 26 Jun95 06 Oct95 MGR
MEMUTIL .SAMPLE   +PRODTAPE 26 Jun95 26 Jun95 06 Oct95 MGR
MODESWIT.SAMPLE   +PRODTAPE 26 Jun95 26 Jun95 06 Oct95 MGR
RESPHIST.SAMPLE   +PRODTAPE 26 Jun95 26 Jun95 06 Oct95 MGR
RESPTIME.SAMPLE   +PRODTAPE 26 Jun95 26 Jun95 06 Oct95 MGR
STOPS   .SAMPLE   +PRODTAPE 26 Jun95 26 Jun95 06 Oct95 MGR
WORKDETL.SAMPLE   +PRODTAPE 26 Jun95 26 Jun95 06 Oct95 MGR
WORKLOAD.SAMPLE   +PRODTAPE 26 Jun95 26 Jun95 06 Oct95 MGR

DBLOADJ .UTIL     +PRODTAPE 26 Jun95 26 Jun95 30 Nov95 MGR
DBLOADNG.UTIL     +PRODTAPE 26 Jun95 26 Jun95 30 Nov95 MGR
FILERPT .UTIL     +PRODTAPE 26 Jun95 26 Jun95 30 Nov95 MGR
FILERPTD.UTIL     +PRODTAPE 26 Jun95 26 Jun95 30 Nov95 MGR
FILERPTJ.UTIL     +PRODTAPE 26 Jun95 26 Jun95 30 Nov95 MGR
FILUSERS.UTIL     +PRODTAPE 07 Dec95 07 Dec95 07 Dec95 MGR
                                  Creator = MGR.HENSLEY
LZW     .UTIL     +PRODTAPE 26 Jun95 26 Jun95 30 Nov95 MGR
ONLINE  .UTIL     +PRODTAPE 27 Nov95 27 Nov95 30 Nov95 MANAGER
                                  Creator = MANAGER.SYS
PCLINK2 .UTIL     +PRODTAPE 26 Jun95 26 Jun95 30 Nov95 MGR
PSCREEN .UTIL     +PRODTAPE 26 Jun95 26 Jun95 30 Nov95 MGR
QUAD    .UTIL     +PRODTAPE 26 Jun95 26 Jun95 30 Nov95 MGR
RAMUSAGE.UTIL     +PRODTAPE 27 Nov95 27 Nov95 30 Nov95 MGR
SL      .UTIL     +PRODTAPE 26 Jun95 26 Jun95 30 Nov95 MGR
SYSLOG  .UTIL     +PRODTAPE 26 Jun95 26 Jun95 30 Nov95 MGR
```

```
DEFRAGX .DEFRAGX +PRODTAPE 10 Oct95 10 Oct95 08 Nov95 MGR
DEFRAGXH.DEFRAGX +PRODTAPE 26 Jun95 26 Jun95 10 Oct95 MGR
HISTORY .DEFRAGX +PRODTAPE 26 Jun95 26 Jun95 10 Oct95 MGR
INITIAL .DEFRAGX +PRODTAPE 08 Sep95 08 Sep95 10 Oct95 MGR
JDEFRAGX.DEFRAGX +PRODTAPE 07 Nov95 07 Nov95 07 Nov95 MGR

QXL     .QXL     +PRODTAPE 07 Jul95 07 Jul95 16 Nov95 MANAGER
QXLCVT  .QXL     +PRODTAPE 26 Jun95 26 Jun95 06 Oct95 MGR
QXMON   .QXL     +PRODTAPE 07 Jul95 07 Jul95 06 Oct95 MANAGER
QXMONJOB.QXL     +PRODTAPE 26 Jun95 26 Jun95 06 Oct95 MGR
QXLOG   .QXL     +PRODTAPE 26 Jun95 26 Jun95 06 Oct95 MGR
QXLOGJOB.QXL     +PRODTAPE 26 Jun95 26 Jun95 06 Oct95 MGR
JCONVQXL.QXL     +PRODTAPE 26 Jun95 26 Jun95 06 Oct95 MGR

Total # of sectors on page:  20,208 (  4 MBs)

TINDEX [2.2] - LPS Toolbox [A.01a]        (c) 1995 Lund Performance Solutions

NM STORE Tape: LPSTAPE, Ldev 7 (DDS), Created: 12/15/95 @  3:35 PM;
                                    Reel # 1  Page 3

FILE    .GROUP   .ACCOUNT  CREATED  MODIFIED ACCESSED CREATOR  LOCKWORD
--------.--------.--------.--------.--------.--------.--------.--------
JQXLA10A.JOB     +PRODTAPE 26 Jun95 26 Jun95 06 Oct95 MANAGER


***END OF TAPE SET***

Total # of sectors on page:      16 (  0 MBs)

Note: 2 files have creators in different accounts.

Largest file: #78 = (5,168 sectors (1 MBs)

Total # of sectors on tape:  42,496 (10 MBs); total # of files: 90

Storage summary...
Group   .Account  Files   Sectors      File# Page  Reel
--------.--------.-------.----------.-------.----.--------
JOB     .PRODTAPE       1         48       1    1        1
PUBSYS  .PRODTAPE       2        352       3    1        1
PUB     .PRODTAPE      30     16,656      33    1        1
CHART   .PRODTAPE       3      4,912      36    1        1
GRAPH   .PRODTAPE      13        464      49    2        1
SAMPLE  .PRODTAPE      14        368      63    2        1
UTIL    .PRODTAPE      14     11,328      77    2        1
DEFRAGX .PRODTAPE       5      5,328      82    2        1
QXL     .PRODTAPE       7      3,024      89    3        1
JOB     .PRODTAPE       1         16      90    3        1
@       .PRODTAPE      90     42,496      90    3        1


VERIFY --> no errors

:
```

Figure 15.1- 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 example, 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"


Please mount reel # 1 for TAPE LPSTAPE.

(Printer file has 132 characters per line)
Will VERIFY by reading every tape record.
Will compare modify dates of files


NM STORE:
   Created              : FRI, DEC 15, 1995,  3:35 PM
   Options: recovery; show; fastsearch; copyacd; No_Compression

There are 90 files on your tape set.
Looking for start of first user file...
Starting to read file labels now...

Tape created: 12/15/95 @  3:35 PM

Will print one dot (.) per 10 files: .........
***END OF TAPE SET***

Largest file: #78 = (5,168 sectors (1 MBs)

Total # of sectors on tape:  42,496 (10 MBs); total # of files: 90

VERIFY --> no errors
:showout

DEV/CL   DFID       JOBNUM  FNAME    STATE FRM SPACE RANK PRI #C
LP       #0348      #S32    LPSLP    READY          48    D 8   1
17       #017       #S32    $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:  176 SECTORS
OUTFENCE = 8
OUTFENCE = 13 FOR LDEV 6
:printo 348

 NM STORE:
    Created              : FRI, DEC 15, 1995,  3:35 PM
    Options: recovery; show; fastsearch; copyacd; No_Compression

 There are 90 files on your tape set.
 TINDEX [2.2] - LPS Toolbox [A.01a]        (c) 1995 Lund Performance Solutions


 NM STORE Tape: LPSTAPE, Ldev 7 (DDS), Created: 12/15/95 @  3:35 PM;
                                        Reel # 1  Page 1

 FILE    .GROUP    .ACCOUNT  CREATED  MODIFIED ACCESSED CREATOR  LOCKWORD
 ------------------------- --------- --------- --------- -------- --------
 LPSINST .JOB      +PRODTAPE=26 Jun95 26 Jun95 10 Oct95 MGR

 INSTOS  .PUBSYS   +PRODTAPE=26 Jun95 26 Jun95 30 Nov95 MGR
 INSTMI  .PUBSYS   +PRODTAPE=26 Jun95 26 Jun95 30 Nov95 MGR
```

```
HOLIDAYS.PUB      +PRODTAPE=26 Jun95 26 Jun95 06 Oct95 MGR
ITEMLIST.PUB      +PRODTAPE=26 Jun95 26 Jun95 06 Oct95 MGR
SOS      .PUB     +PRODTAPE=26 Jun95 26 Jun95 14 Nov95 MGR
SOSKIP   .PUB     +PRODTAPE=26 Jun95 26 Jun95 06 Oct95 MGR
LPSCFG   .PUB     +PRODTAPE=15 Dec95 15 Dec95 15 Dec95 MGR
LPSCHECK.PUB      +PRODTAPE=26 Jun95 26 Jun95 06 Oct95 MGR
LPSEXTND.PUB      +PRODTAPE=26 Jun95 26 Jun95 06 Oct95 MGR
LOGHELP .PUB      +PRODTAPE=26 Jun95 26 Jun95 06 Oct95 MGR
PFGITEMS.PUB      +PRODTAPE=26 Jun95 26 Jun95 06 Oct95 MGR
RCITEMS .PUB      +PRODTAPE=26 Jun95 26 Jun95 06 Oct95 MGR
REDITEMS.PUB      +PRODTAPE=26 Jun95 26 Jun95 06 Oct95 MGR
REPRTDEF.PUB      +PRODTAPE=26 Jun95 26 Jun95 06 Oct95 MGR
SL       .PUB     +PRODTAPE=26 Jun95 26 Jun95 06 Oct95 MGR
SL94000A.PUB      +PRODTAPE=26 Jun95 26 Jun95 30 Nov95 MGR
SOSADVIC.PUB      +PRODTAPE=26 Jun95 26 Jun95 06 Oct95 MGR
SOSFULL .PUB      +PRODTAPE=26 Jun95 26 Jun95 06 Oct95 MGR
SOSGRAPH.PUB      +PRODTAPE=26 Jun95 26 Jun95 06 Oct95 MGR
SOSHELP .PUB      +PRODTAPE=26 Jun95 26 Jun95 06 Oct95 MGR
SOSJOB  .PUB      +PRODTAPE=26 Jun95 26 Jun95 06 Oct95 MGR
SOSLOGX .PUB      +PRODTAPE=26 Jun95 26 Jun95 06 Oct95 MGR
SOSLOGXJ.PUB      +PRODTAPE=26 Jun95 26 Jun95 06 Oct95 MGR
SOSMONJ .PUB      +PRODTAPE=26 Jun95 26 Jun95 02 Nov95 MGR
SOSPRANJ.PUB      +PRODTAPE=26 Jun95 26 Jun95 06 Oct95 MGR
RPTCARDJ.PUB      +PRODTAPE=22 Sep95 22 Sep95 09 Oct95 MGR
PERFCOLJ.PUB      +PRODTAPE=16 Aug95 16 Aug95 14 Nov95 MGR
SOSPRANL.PUB      +PRODTAPE=26 Jun95 26 Jun95 07 Nov95 MGR
SOSPRDMP.PUB      +PRODTAPE=26 Jun95 26 Jun95 06 Oct95 MGR
SOSRCOM .PUB      +PRODTAPE=26 Jun95 26 Jun95 06 Oct95 MGR
SOSSNOOP.PUB      +PRODTAPE=26 Jun95 26 Jun95 06 Oct95 MGR
XL       .PUB     +PRODTAPE=26 Jun95 26 Jun95 06 Oct95 MGR

ISFONT   .CHART   +PRODTAPE=26 Jun95 26 Jun95 30 Nov95 MGR
ISROOT   .CHART   +PRODTAPE=26 Jun95 26 Jun95 30 Nov95 MGR
SOSCHART.CHART    +PRODTAPE=26 Jun95 26 Jun95 30 Nov95 MGR

CPUUDEX1.GRAPH    +PRODTAPE=26 Jun95 26 Jun95 06 Oct95 MGR
CPUUTIL1.GRAPH    +PRODTAPE=26 Jun95 26 Jun95 06 Oct95 MGR
CPUUTIL2.GRAPH    +PRODTAPE=26 Jun95 26 Jun95 06 Oct95 MGR
CPUUTIL3.GRAPH    +PRODTAPE=26 Jun95 26 Jun95 06 Oct95 MGR
CPUUTIL4.GRAPH    +PRODTAPE=26 Jun95 26 Jun95 06 Oct95 MGR
DISCIO1 .GRAPH    +PRODTAPE=26 Jun95 26 Jun95 06 Oct95 MGR
DISCIO2 .GRAPH    +PRODTAPE=26 Jun95 26 Jun95 06 Oct95 MGR
Total # of sectors on page:  22,272 (  5 MBs)

TINDEX [2.2] - LPS Toolbox [A.01a]        (c) 1995 Lund Performance Solutions

NM STORE Tape: LPSTAPE, Ldev 7 (DDS), Created: 12/15/95 @  3:35 PM;
                                  Reel # 1  Page 2


FILE     .GROUP    .ACCOUNT  CREATED  MODIFIED ACCESSED CREATOR  LOCKWORD
-------------------------- -------- -------- -------- -------- --------
GRAPHCAT.GRAPH    +PRODTAPE=26 Jun95 26 Jun95 06 Oct95 MGR
GRAPHDEV.GRAPH    +PRODTAPE=26 Jun95 26 Jun95 06 Oct95 MGR
MEMUTIL1.GRAPH    +PRODTAPE=26 Jun95 26 Jun95 06 Oct95 MGR
MODESWI1.GRAPH    +PRODTAPE=26 Jun95 26 Jun95 06 Oct95 MGR
RESPTIM1.GRAPH    +PRODTAPE=26 Jun95 26 Jun95 06 Oct95 MGR
TRANMGR1.GRAPH    +PRODTAPE=26 Jun95 26 Jun95 06 Oct95 MGR

ANALRPT .SAMPLE   +PRODTAPE=26 Jun95 26 Jun95 06 Oct95 MGR
CPUUTIL .SAMPLE   +PRODTAPE=26 Jun95 26 Jun95 06 Oct95 MGR
DISCFREE.SAMPLE   +PRODTAPE=26 Jun95 26 Jun95 06 Oct95 MGR
DISCINFO.SAMPLE   +PRODTAPE=26 Jun95 26 Jun95 06 Oct95 MGR
DISCUTIL.SAMPLE   +PRODTAPE=26 Jun95 26 Jun95 06 Oct95 MGR
DISCVOL .SAMPLE   +PRODTAPE=26 Jun95 26 Jun95 06 Oct95 MGR
GLOBAL  .SAMPLE   +PRODTAPE=26 Jun95 26 Jun95 06 Oct95 MGR
MEMUTIL .SAMPLE   +PRODTAPE=26 Jun95 26 Jun95 06 Oct95 MGR
```

```
MODESWIT.SAMPLE   +PRODTAPE=26 Jun95 26 Jun95 06 Oct95 MGR
RESPHIST.SAMPLE   +PRODTAPE=26 Jun95 26 Jun95 06 Oct95 MGR
RESPTIME.SAMPLE   +PRODTAPE=26 Jun95 26 Jun95 06 Oct95 MGR
STOPS    .SAMPLE  +PRODTAPE=26 Jun95 26 Jun95 06 Oct95 MGR
WORKDETL.SAMPLE   +PRODTAPE=26 Jun95 26 Jun95 06 Oct95 MGR
WORKLOAD.SAMPLE   +PRODTAPE=26 Jun95 26 Jun95 06 Oct95 MGR

DBLOADJ .UTIL     +PRODTAPE=26 Jun95 26 Jun95 30 Nov95 MGR
DBLOADNG.UTIL     +PRODTAPE=26 Jun95 26 Jun95 30 Nov95 MGR
FILERPT .UTIL     +PRODTAPE=26 Jun95 26 Jun95 30 Nov95 MGR
FILERPTD.UTIL     +PRODTAPE=26 Jun95 26 Jun95 30 Nov95 MGR
FILERPTJ.UTIL     +PRODTAPE=26 Jun95 26 Jun95 30 Nov95 MGR
FILUSERS.UTIL     +PRODTAPE=07 Dec95 07 Dec95 07 Dec95 MGR
                                 Creator = MGR.HENSLEY
LZW      .UTIL    +PRODTAPE=26 Jun95 26 Jun95 30 Nov95 MGR
ONLINE   .UTIL    +PRODTAPE=27 Nov95 27 Nov95 30 Nov95 MANAGER
                                 Creator = MANAGER.SYS
PCLINK2  .UTIL    +PRODTAPE=26 Jun95 26 Jun95 30 Nov95 MGR
PSCREEN  .UTIL    +PRODTAPE=26 Jun95 26 Jun95 30 Nov95 MGR
QUAD     .UTIL    +PRODTAPE=26 Jun95 26 Jun95 30 Nov95 MGR
RAMUSAGE.UTIL     +PRODTAPE=27 Nov95 27 Nov95 30 Nov95 MGR
SL       .UTIL    +PRODTAPE=26 Jun95 26 Jun95 30 Nov95 MGR
SYSLOG   .UTIL    +PRODTAPE=26 Jun95 26 Jun95 30 Nov95 MGR

DEFRAGX  .DEFRAGX +PRODTAPE=10 Oct95 10 Oct95 08 Nov95 MGR
DEFRAGXH.DEFRAGX  +PRODTAPE=26 Jun95 26 Jun95 10 Oct95 MGR
HISTORY  .DEFRAGX +PRODTAPE=26 Jun95 26 Jun95 10 Oct95 MGR
INITIAL  .DEFRAGX +PRODTAPE=08 Sep95 08 Sep95 10 Oct95 MGR
JDEFRAGX.DEFRAGX  +PRODTAPE=07 Nov95 07 Nov95 07 Nov95 MGR

QXL      .QXL     +PRODTAPE=07 Jul95 07 Jul95 16 Nov95 MANAGER
QXLCVT   .QXL     +PRODTAPE=26 Jun95 26 Jun95 06 Oct95 MGR
QXMON    .QXL     +PRODTAPE=07 Jul95 07 Jul95 06 Oct95 MANAGER
QXMONJOB.QXL      +PRODTAPE=26 Jun95 26 Jun95 06 Oct95 MGR
QXLOG    .QXL     +PRODTAPE=26 Jun95 26 Jun95 06 Oct95 MGR
QXLOGJOB.QXL      +PRODTAPE=26 Jun95 26 Jun95 06 Oct95 MGR
JCONVQXL.QXL      +PRODTAPE=26 Jun95 26 Jun95 06 Oct95 MGR

Total # of sectors on page:  20,208 (  4 MBs)

TINDEX [2.2] - LPS Toolbox [A.01a]        (c) 1995 Lund Performance Solutions

NM STORE Tape: LPSTAPE, Ldev 7 (DDS), Created: 12/15/95 @  3:35 PM;
                                    Reel # 1  Page 3

FILE     .GROUP    .ACCOUNT  CREATED  MODIFIED ACCESSED CREATOR  LOCKWORD
-------- --------- --------- -------- -------- -------- -------- --------
JQXLA10A.JOB      +PRODTAPE=26 Jun95 26 Jun95 06 Oct95 MANAGER


***END OF TAPE SET***

Total # of sectors on page:      16 (  0 MBs)

Note: 2 files have creators in different accounts.

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                                0
    =   Same date as disk file                             90
    -   Disc file does not exist                            0
    ?   Error during comparison                             0
```

```
Largest file: #78 = (5,168 sectors (1 MBs)

Total # of sectors on tape:  42,496 (10 MBs); total # of files: 90

Storage summary...
Group    .Account  Files   Sectors      File# Page  Reel
--------,-------- ------- ---------- ------- ---- -------
JOB      .PRODTAPE       1          48       1    1       1
PUBSYS   .PRODTAPE       2         352       3    1       1
PUB      .PRODTAPE      30      16,656      33    1       1
CHART    .PRODTAPE       3       4,912      36    1       1
GRAPH    .PRODTAPE      13         464      49    2       1
SAMPLE   .PRODTAPE      14         368      63    2       1
UTIL     .PRODTAPE      14      11,328      77    2       1
DEFRAGX  .PRODTAPE       5       5,328      82    2       1
QXL      .PRODTAPE       7       3,024      89    3       1
JOB      .PRODTAPE       1          16      90    3       1
@        .PRODTAPE      90      42,496      90    3       1


VERIFY --> no errors

:
```

Figure 15.2 - 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.

Message	***Error Reading Tape***
Cause	Can occur for a variety of reasons.
Action	Ensure integrity of media, clean tape mechanism.
Message	***Failed on Forward-Skip-File on tape***
Cause	Error using SKIP option on cartridge, serial disk media.
Action	Don't use SKIP option. Ensure integrity of media.
Message	****Too many tape errors****
Cause	More than 100 verify errors or more than 9 tape errors were detected.
Action	Ensure integrity of media, clean tape mechanism.

Section 2

Developers Toolbox

Chapter 16

The AVATAR Tool

AVATAR's decompiler capabilities include the ability to find, view, and modify the contents of any Native Mode program file, object file, executable library, or relocatable library. The AVATAR command set includes about 40 commands that simplify tasks like disassembling and modifying program files. Other features are geared towards deciphering header information in executable libraries and extracting portions of code into assembly language source.

Warning: AVATAR was designed to be used by experienced software engineers. In terms of how it is used, AVATAR is very similar in feel to Hewlett-Packard's DEBUG. Therefore, if you are not comfortable using DEBUG you will not be comfortable using AVATAR. Proceed at your own risk, exercising appropriate caution.

AVATAR is more effectively used if you understand the following concepts:

1. HPPA assembly language
2. Procedure calling
3. Parameter passing conventions

Operation

The primary use of AVATAR is to perform operations on SOMs. A SOM is a file that conforms to HP's Standard Object Module conventions. There are four classes of files with which AVATAR is particularly familiar. Each of these four classes is easily identified by its filecode:

NMPRG	= native mode program files
NMXL	= native mode executable libraries
NMRL	= native mode relocatable libraries
NMOBJ	= native mode object files

In addition to working on the file classes listed above, AVATAR can also be used as a binary editor to display and modify most other MPE files.

When AVATAR is used as a decompiler, its output is displayed as assembly language and hexadecimal constants. To add symbolic information about register usage to the disassembled display, use AVATAR's SYN command.

A complete description of the assembly language can be found in HP's *Precision Architecture and Instruction Reference Manual*. Another useful manual is HP's *Procedure Calling Convention Reference Manual*, which describes how the general registers and stack frame are set up for procedure calls. (Use the CSEQ tool to display the calling sequences for MPE intrinsics.)

After starting AVATAR, the "AVATAR:" prompt will be displayed. The next step is usually to OPEN a file. At that point, commands are entered to accomplish the task at hand. The general form for entering commands is:

AVATAR: <command> [<expression>]

The sections that follow describe the syntax and usage for all of AVATAR's commands as well as the structure of an expression.

Capabilities

Program capabilities required include IA, BA, PM, DS, and PH. PM is required to run DEBUG.

Usage

AVATAR can be started from the supplied UDC or from a RUN statement. AVATAR does not use the INFO string or PARM.

To start AVATAR, use one of the following methods:

- UDC
  :AVATAR
- RUN
  :RUN AVATAR.PUB.LPSTOOLS

Expression Structure

<expression> ::= <term> [+ | - <term>]
<term> ::= <factor> [* | / <factor>]
<factor> ::= [+ | -] <primary>
<primary> ::= [<expression>]
[` <assembler instruction> `]
[<number>]
[SOM_HEADER]
[LST_HEADER]
[PROCTIME]
[<symbol>]
[" <symbol>"]

<assembler instruction> is a valid assembler instruction. The instruction is enclosed in back-quotes.

<number>::=
[$ <hexadecimal digits>]
[# <decimal digits>]
[% <octal digits>]
[<digits in current radix>]

<symbol> is the value of any symbol defined in the current SOM. If the symbol is not enclosed in quotes, then it cannot be one of the previously defined words (e.g., PROCTIME) and it can only contain characters from the set A-Z, a-z, 0-9, _, $, #, %.

If the name of the symbol is preceded with a question mark (?), then the value of a stub with that name is used.

Strings are also used in many commands. Strings can be given as a simple string or as a compound string. A simple string is zero or more characters enclosed in double-quotes. A compound string is a list of substrings, enclosed in braces ({}). A substring can be a string enclosed in double-quotes or a number representing the value of one byte. For example, "this is a simple string," while {"this is a compound string with a new-line character"$a}.

Foundation Topic Discussions

This section discusses concepts and terminology that you may find helpful in understanding the information presented about AVATAR. First, a brief background section introduces Standard Object Modules (SOMs), and then assembly language and mapped files are discussed in relation to how they are used in AVATAR.

Standard Object Modules

Standard Object Modules are the smallest unit which may be generated by a compiler. They correspond to a given order, regardless of the file type. For instance, the architecture of an NMPRG begins with header and procedural information that is important to the operating system. After this, data and code segments follow.

A set of SOMs is defined as a library which may be either executable (NMXL) or relocatable (NMRL). Each library will contain a library symbol table (LST) that describes its contents in terms of SOMs.

Relocatable libraries contain one or more SOMs that must be linked (using LINKEDIT) with the SOM that references it. Executable libraries contain one or more SOMs that have already been linked and are ready to execute. The SOMs in an executable library are dynamically loaded by MPE/iX when referenced.

Multiple SOMs can be stored in an object file, an executable library or a relocatable library. Once procedures are bound into a single SOM, they cannot be separated. AVATAR provides the capability to patch the assembler code of your compiled program. This means you now have the ability to support discontinued programs that may be important to your business or patch those almost-perfect programs when your vendor's bug priority list doesn't quite coincide with yours.

A SOM can contain many procedures that have been combined into a single SOM. Normally, once a set of procedures has been combined by a compiler into a SOM, they are not easily separated from the SOM. AVATAR's EXTRACT command breaks the SOM out into a separate ASCII file in assembler format that can be edited and assembled.

Assembly Language

Hewlett-Packard's Precision Architecture Assembly Language is a symbolic, more approachable, representation of MPE/iX machine language. Familiarity with assembly language may prove helpful in understanding AVATAR's output, capabilities, and features.

Mapped Files

"Mapped Files" refers to the virtual address space used by files. This gives the operating system direct reference to all types of information in a manner that is reminiscent of disk-caching. Every byte of every opened file has a unique virtual address. Portions of files are brought into real memory on demand, leaving behind other portions that are not yet required (*Note*: Use the KLONDIKE tool from the ***System Managers Toolbox*** to view how much of a file is in real memory). MPE/iX's treatment of virtual memory brings efficiency and flexibility to memory management that was non-existent with MPE V.

Command Summary

The following list provides a simple description of AVATAR 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 AVATAR requires in order to distinguish one command from another. The commands themselves are not case-sensitive.

Command Code	Description
<CR>	A Carriage Return displays more data after a DP, DC, DD, or DV command is issued
=	An equal sign calculates a value from an expression
Asm	Generates assembly code equivalent
AUx	Prints the auxiliary headers
CALCulate	Evaluates an expression and displays the result.
Callee	Locates all procedure calls over specified range
CALLS	Locates specified procedure calls over a range
CHecksum	Recalculates CHECKSUM
CLose	Closes file
COmpiler	Provides general information
COUnt	Displays symbol type, scope counts
DC	Displays data at a code address
DD	Displays data values at a code address
Debug	Invokes DEBUG
DIsasm	Displays assembler instruction equivalent
DP	Displays data on procedure name
DR	Displays real memory
DV	Displays data values from current file
Exit	Terminates AVATAR
EXTract	Extracts portion of code into ASCII file
Find	Searches symbol dictionary for string
FINDAll	Finds all symbol dictionary entries for a string
FIXup	Displays compiler fixup entries
FOrmat	Format options for data display
HELP	Invokes AVATAR help
Init	Displays compiler initialization entries
Look	Displays information on a symbol dictionary entry
LSt	Lists all module names in SOM
MC	Modifies data at a code address
MD	Modifies values from initialized area
MV	Modifies values within current SOM
Next	Repeats last display or modify command
Open	Opens a file
Quit	Exits the program
Radix	Changes default radix
Search	Searches entire SOM for string
SET/REset	Enables and disables options
SPace	Displays (sub)space information
STatistics	Displays scope & type statistics
STRIP	Remove symbolic information from SOM
SUbspace	Displays subspace information
SYMFormat	Format options for data display
SYMOpen	Opens a SYMOS file for examination
SYn	Sets synonyms for registers
Uncalled	Displays uncalled entry points
UNWind	Displays unwind descriptors

Command Definitions

This section describes AVATAR commands in detail. *Note*: Most of these commands require that an SOM has been previously selected.

<CR>

The carriage return **<CR>** can be used in conjunction with AVATAR's display commands (DC, DD, DP, and DV) to show additional screens of information without having to type NEXT in order to do so. If this type of response is not desired, you can restrict this unprompted display of additional information through the ***Developers Toolbox*** standard commands CRON and EATEMPTY.

= <expression>

The equal sign (=) operator when followed by an expression can be used to calculate the value of the expression. For example:

```
= 5+3
= 'nop'
```

ASM <assembler instruction>

The ASM (assemble) command is used to generate the opcode that corresponds with the valid assembler instruction that the user enters. Any symbols in the current SOM may be used to construct the assembler instruction. The output from this command is of the form:

value = <decimal for opcode>, $<hexadecimal for opcode>

For example:

```
AVATAR: asm nop
value = 134218304, $8000240
```

See the *Precision Architecture and Instruction Reference Manual* for a complete discussion of the *HP3000* instruction set.

AUX command

This command displays the auxiliary header information for the SOM. This information is displayed as a series of one or more auxiliary headers. Each auxiliary header is constructed of 6 fields:

MANDATORY	The MANDATORY field is used to indicate if this SOM contains information that the linker must understand.
COPY	The COPY field is used to indicate that this auxiliary header should be copied without change to any new SOM created from this SOM.
APPEND	The APPEND field is used to indicate entries with the same TYPE and APPEND fields should be merged together.
IGNORE	The IGNORE field is used to indicate this auxiliary header should be ignored if its TYPE field is unknown.
TYPE	The TYPE field is a numeric field that is used to describe the contents of this auxiliary header. The list of known values are provided next.
LENGTH	See the following table.

Known values for the Type field:

Value	Meaning	Associated Auxiliary Header
0	NULL	
1	LINK information	LINK aux header
2,7	HP Program	HP Program aux header
3	DEBUG	DEBUG aux header
4	HP-UX aux header	HP-UX aux header
5	IPL aux header	IPL aux header
6	User string auxiliary header	User String aux header
8	SOM	HP SOM aux header

The LENGTH field contains the number of bytes in the auxiliary header less 4 bytes.

Auxiliary Header Definitions

The various headers that can be used with the AUX command are described below:

LINK This auxiliary header is used to record the last time that the linker modified the SOM. The four elements in this header include:

aux header id
linker product id
linker version id
link time

HP Program This auxiliary header contains information that is used by the operating system to load an executable. The seven elements in this header include:

aux header id
entry name
unsat names
search list
capabilities
max stacksize
max heapsize

DEBUG This auxiliary header is used to record the last time that the debugger modified the SOM. The four elements in this header include:

aux header id
debugger product id
debugger version id
debug time

HP-UX This auxiliary header contains information that is used by the UX loader. The eleven elements in this header include:

aux header id
execute code size
execute code offset memory
execute code offset SOM
execute data size
execute data offset memory
execute data offset SOM
execute uninitialized data size
execute start entry
execute initialized data
execute loader flags

IPL This auxiliary header contains information that is used for loading bootable utilities. The six elements in this header include:

aux header id
file length
physical address destination
entry offset
bbs size
checksum

User String This auxiliary header is used to store user definable strings. Typically the user-definable strings are defined through compiler directives like VERSION and COPYRIGHT. The three elements in this header are:

aux header id
string length
string

HP SOM This auxiliary header contains information necessary to load executable SOMs. The seven elements of this header are:

aux header id
SOM flag
num of XRTs
unwind start
unwind end
recover start
recover end

```
:avatar
AVATAR [2.0] - LPS Toolbox [A.01a]          (c) 1995 Lund Performance Solutions

For Help at the AVATAR prompt enter   ?


AVATAR: open xl
----> LST Module Directory <-----     Starts @   #Length
1  hp30026_01                         $00071000   263744
2  HP30138                            $000b2000   971640
3  STIS209S                           $001a0000    23228
4  RLBPROCS                           $001a6000   308792
5  HP35360                            $001f2000   133108
6  SENTRYTI                           $00213000   853948
7  HP36395                            $002e4000    69008
8  SJAS424S                           $002f5000   131488
9  LIB1SRC                            $00316000   134036
10  SPECAUX                            $00337000   185892
11  HP315021                           $00365000   165332
12  HP31511_01                         $0038e000   107908
13  HP32715                            $003a9000   443348
14  HP36957                            $00416000   249952
15  HP36961                            $00454000    62456
16  PSICOMN                            $00464000   640984
17  STEALTH                            $00501000    43332
18  FMT                                $0050c000   539620
19  AHPDINT                            $00590000    31732
20  LANCELOT                           $00598000   335496
21  corelib_01                         $005ea000    24952
22  LSS                                $005f1000    18452
23  SNMP                               $005f6000    82456
```

```
24  NSR                                 $0060b000   403772
25  TELNETO                             $0066e000   256736
26  SOCKET                              $006ad000   241576
27  PSPNMSTB                            $006e8000    36368
28  S01STLIB                            $006f1000    21120
29  NMEVNT                              $006f7000    13680
30  SV1S209X                            $006fb000   728728
31  S25S391C                            $007ad000    17420
32  HP31501_02                          $007b2000   179812
33  HP315022                            $007de000    53700
34  corelib_02                          $007ec000    57884
35  S0FP935N                            $007fb000    11360
36  S27S391C                            $007fe000    31208
37  HP31501_03                          $00806000   302264
38  U_Qfabs                             $00850000    75304
39  SZAS393S                            $00863000    18608
40  S29S391C                            $00868000    17452
41  dbcore.p                            $0086d000  1362068
42  APALTERS                            $009ba000   409752
43  HPSQL2                              $00a1f000  2184008
44  HPSQL3                              $00c35000   751208
45  HPSQL4                              $00ced000   223244
46  HPSQL5                              $00d24000   501136
47  HPSQL6                              $00d9f000   111860
48  HPSQL7                              $00dbb000    35452
49  HPSQL8                              $00dc4000   102028
50  HPSQL9                              $00ddd000     9272
51  HP31900                             $00de0000   167440
52  B3821A1                             $00e09000    18552
53  B3821A2                             $00e0e000    22968
Select a module number > 17

      Module #  17: STEALTH
      Found 79 unwind entries.
      Searching 206 symbol dictionary entries
      Sorting 169 symbols
FILE TYPE : sharable, executable SOM

AVATAR[xl]: aux
mandatory       : FALSE
copy            : FALSE
append          : FALSE
ignore          : TRUE
type            :           8
length          :          36
HPE som flag          : FALSE
system som flag       : FALSE
Number of XRTs        :          24

mandatory       : TRUE
copy            : FALSE
append          : FALSE
ignore          : FALSE
type            :           1
length          :          32
debugger product id   :  HP30315
debugger version id   :  A.05.06

mandatory       : FALSE
copy            : FALSE
append          : TRUE
ignore          : FALSE
type            :           6
length          :          56
user string           : $Header: xl0.s.v 1.6 86/06/26 17:28:03 cary Exp $
```

```
mandatory       : FALSE
copy            : FALSE
append          : TRUE
ignore          : FALSE
type            :           6
length          :          60
user string           : @(#) STEALTH_01, A.00.50;  TUE, DEC  6, 1994 10:40 PM

mandatory       : FALSE
copy            : FALSE
append          : TRUE
ignore          : FALSE
type            :           6
length          :          68
user string           : [IND]@(#)C0010    ($Revision: 1.2 $) LSSLINKCONTROL  Inc
lude

mandatory       : FALSE
copy            : FALSE
append          : TRUE
ignore          : FALSE
type            :           6
length          :          52
user string           : a0012/d/dstatus/lliomsg/$revision: 1.2.1.2 $

mandatory       : FALSE
copy            : FALSE
append          : TRUE
ignore          : FALSE
type            :           6
length          :          52
user string           : a0000/d/ddiagio/lliomsg/$revision: 1.2.1.2 $

mandatory       : FALSE
copy            : FALSE
append          : TRUE
ignore          : FALSE
type            :           6
length          :          60
user string           : A0000/D/DFARTBL /FILE OPEN       /$Revision: 1.1.7.2 $

mandatory       : FALSE
copy            : FALSE
append          : TRUE
ignore          : FALSE
type            :           6
length          :          56
user string           : A0000/D/DMACTBL /FILE OPEN       /$Revision: 1.5 $

mandatory       : FALSE
copy            : FALSE
append          : TRUE
ignore          : FALSE
type            :           6
length          :          40
user string           : A0100/d/diopm/hlio/$Revision: 1.4 $

mandatory       : FALSE
copy            : FALSE
append          : TRUE
ignore          : FALSE
type            :           6
length          :          48
user string           : A0003/D/DOBJCL/VSM/$Revision: 1.15.6.2 $
```

```
mandatory       : FALSE
copy            : FALSE
append          : TRUE
ignore          : FALSE
type            :           6
length          :          44
user string           : A0003/D/DVSM/VSM/$Revision: 1.9.13.2 $

mandatory       : FALSE
copy            : FALSE
append          : TRUE
ignore          : FALSE
type            :           6
length          :          24
user string           : @(#) apatch4 1.1


AVATAR[x1]: exit
:
```

Figure 16.1 - AUX Example

CALCulate <expression>

The CALCULATE command will evaluate an expression and display the resultant value in hexadecimal and decimal.

<expression> An arithmetic expression

All calculations are done using 32-bit integer arithmetic. For example:

= 5+20-$15
= FOPEN

Callee [<calleename> <procedurename>]
[<calleename> <start offset> <end offset>]

This command is used to locate all calls to a given procedure **calleename** over the specified range. The **calleename** can be any symbol found in the currently selected SOM. The **calleename** cannot contain any wildcards and is case-sensitive. Ranges are specified in one of three ways: by **procedurename**, by an explicit offset range, or not specified. An unspecified range forces a search of the entire SOM. The range specifiers can be constructed using any valid expression.

CALLS [<procedurename>]
[<start offset> <end offset>]

This command is used to locate calls to a given procedure over the specified range. Ranges are specified in one of three ways: by **procedurename**, by an explicit offset range, or not specified. The latter forces a search of the entire SOM. The range specifiers can be constructed using any valid expression.

CHecksum

This command has no parameters. It works with the currently opened SOM. The SOM header is constructed of 124 bytes. The CHECKSUM entry is the last word in the header and is not included in its calculation. Anytime AVATAR changes part of the SOM header the CHECKSUM command should be used to recalculate the CHECKSUM. The CHECKSUM word is created by exclusive "OR-ing" all of the words in

the SOM header. This provides a simple, quick way of determining if the SOM contains a valid header. The CHECKSUM command will calculate the checksum for the currently opened SOM file and replace the existing value with the new one. This is necessary if you have modified the first 124 bytes of the SOM file.

CLose

This command closes any currently opened file. The OPEN command implicitly uses this command when switching files.

COmpiler

This command has no parameters. Output from this command provides general information about the SOM, as well as information specific to each module within the SOM. For example, a compiled C module could contain the following information. For a NMOBJ file on a Spectrum machine, general information items (numbers in hexadecimal) are:

system id	=	**20B**	Spectrum architecture
magic #	=	**106**	Relocatable SOM
version id	=	**5124000**	SOM structure version

SOM-specific information:

source file name = C2SPL
language name = HP-C
product id = HP31506
version id = C/XL Compiler Version A.01.22

Other magic numbers that might be generated for a NMOBJ file on a Spectrum machine include the following:

(These two entries have Library Symbol Table headers)

$104 = executable library (NMXL)
$619 = relocatable library (NMRL)

(These entries have SOM headers.)

$106 = relocatable SOM
$107 = non-shareable, executable SOM
$108 = shareable, executable SOM
$10B = shareable, demand-loaded executable SOM

COUnt

This command is used to display the symbol type and corresponding scopes for all symbols in the current SOM in tabular format. Provided in the following list is a complete listing and short definition of each symbol type for a SOM. Following the symbol types list is a similar list for symbol scopes.

SOM Symbol Types	
NULL	invalid symbol record
ABSOLUTE	absolute constant
DATA	normal initialized data
CODE	unspecified code, resolved at link time
PRI_PROG	primary program entry point
SEC_PROG	secondary program entry point

ENTRY	code entry point symbols
STORAGE	storage requirement; known length, unknown value
STUB	external call stub, or relocation stub
MODULE	source module name
SYM_EXT	extension record of the current entry
ARG_EXT	extension record of the current entry
MILLICODE	name of a millicode subroutine
PLABEL	procedure label
OCT_DIS	used by OCT (Object Code Translator)
MILLI_EXT	address of an external millicode subroutine

SOM Symbol Scopes

UNSAT	unsatisfied import request
EXTRN	import request to a symbol in another SOM
LOCAL	private symbol
UNIVERSAL	symbol to be exported outside the SOM

	SCOPE			
TYPE	**UNSAT**	**LOCAL**	**EXTERNAL**	**UNIVERSAL**
NULL				
ABSOLUTE	X	X		X
DATA	X	X		X
CODE	X	X		X
PRI_PROG				X
SEC_PROG				X
ENTRY		X		X
STORAGE	X	X		X
STUB		X	X	
MODULE		X		X
SYM_EXT				
ARG_EXT				
MILLICODE	X	X		X
PLABEL		X		
OCT_DIS		X		X
MILLI_EXT				X

DC <expression> [<display format>] [Length]

The DC command will display data at a given code address. The address can be specified as either a procedure name (including offset) or as a code offset. The default display will be in assembler format although a mixed ASCII and hexadecimal display is also available.

<expression>	= <procedure name> | <code offset>
<procedure-name>	Any procedure that has been defined in the current module of the opened SOM file.
<code-offset>	An expression giving the offset to the start of the current code module where data to display starts.
<display-format>	Either C or D. Default is C. If D is specified then data is displayed in hex and ASCII format. If C is specified then data is displayed as disassembled code.
Length	The number of output lines to display.

DD <data offset> [<display format>]

The DD command will display values (data) from within the current SOM. The default display mode will display data in both hexadecimal and ASCII formats. However, a disassembled format is also available.

<data offset>	An expression giving the start within an initialized block where display starts.
<display format>	Either C or D. Default is C. If D is specified then data is displayed in hex and ASCII format. If C is specified then data is displayed as disassembled code.

Debug

This command invokes the system debug program, DEBUG. There are no parameters for this command.

```
AVATAR: debug
DEBUG Intrinsic at: 300.00031040 mainline+$488
$6 ($55) nmdebug > e
AVATAR:
```

See the *System Debug Reference Manual* for details on using DEBUG.

DIsasm <expression>

The DISASM command shows the assembler instruction corresponding to an opcode defined by expression.

```
AVATAR: disasm $2b600000
ADDIL   $0,27
```

DP <procedure name>

The DP command will display data for a given procedure name located in the current procedure symbol table. *Note*: Symbol names are case-sensitive.

```
AVATAR: dp _start
_start      3 3 CODE    UNIVERSAL 44ec    5ea3
[C]  5ea0:   6bc23fd9 STW       2,-20(0,30)
[C]  5ea4:   6fc322c0 STWM      3,4448(0,30)
[C]  5ea8:   e8400130 BL        _init_c_globals,2 ; 5f48
[C]  5eac:   08000240 NOP
```

DR <expression>

The DR command will display real memory starting at the physical address given by the expression.

```
AVATAR: dr $4000
```

DV <data offset> [<display format>] [<numlines>]

<data offset>	An expression giving the offset from the start of the file where display starts.
<display format>	Either C or D. Default is D. If D is specified then data is displayed in hex and ASCII format. If C is specified then data is displayed as disassembled code.

<numlines> A decimal number indicating the number of lines to show. The default value is to keep on displaying until the next Ctrl+Y or "/" reply.

The DV command will display values (data) from the current file. Data will be displayed in hexadecimal and ASCII formats by default.

For example:

DV $100 Displays data starting at address hex 100
DV 5+20-$15 c Display lines at address 4, disassembled as code

To dump the SOM header for a NMOBJ, try the following:

```
:avatar
AVATAR [2.0] - LPS Toolbox [A.01a]          (c) 1995 Lund Performance Solutions

For Help at the AVATAR prompt enter   ?


AVATAR: open config.rel.ccscxl
FILE TYPE : relocatable SOM

AVATAR[config.rel.ccscxl]: dv 0 d
[V]       0: 020b0106 053113cc 00000000 00000000 .....1..........
[V]      10: 00000000 00000000 00000000 00000354 ...............T
[V]      20: 00000000 000007ac 40000000 000000b0 ........@.......
[V]      30: 00000003 0000011c 0000000a 000002ac ................
[V]      40: 00000000 000002ac 000000a8 00000000 ................
[V]      50: 00000000 00000354 00000001 0000058c .......T........
[V]      60: 0000000d 00000564 00000028 00000690 .......d...(....
[V]      70: 0000011c 00000378 000001ec 473a11bf .......x....G:..
[V]      80: 2e482e43 43534358 4c000000 00000000 .H.CCSCXL.......
[V]      90: 00000000 0000001f 00000000 00000000 ................
[V]      a0: 00000000 00000000 00000001 00000000 ................
[V]      b0: 00000004 c0000800 00000000 00000000 ................
[V]      c0: 00000003 ffffffff 00000000 ffffffff ................
[V]      d0: 00000000 00000010 c0001000 00000001 ................
[V]      e0: 00000003 00000002 ffffffff 00000000 ................
[V]      f0: ffffffff 00000000 00000020 60005000 ........... '.P.
[V]     100: 00000002 00000005 00000005 ffffffff ................
[V]     110: 00000000 ffffffff 00000000 00000000 ................
[V]     120: 58201000 00000080 00000000 00000000 X ..............
[V]     130: 00000000 00000008 0000002c ffffffff ...........,....
...
[V]     780: 65730000 0000000b 63686172 5f736967 es......char_sig
[V]     790: 6e656400 00000004 24667031 00000000 ned.....$fp1....
[V]     7a0: 00000004 24667032 00000000 00000000 ....$fp2........
   repeat last word until      7fc (#20 words)
AVATAR[config.rel.ccscxl]: exit
:
```

Figure 16.2 - Dumping the SOM

Exit

The Exit command terminates AVATAR.

EXtract <file name> <start > [<end>]

The EXTRACT command extracts a portion of code into an ASCII file. This file can then be used as input to the ASSEMBLER.

<file name> The name of a file to be created. The file must not already exist.

<start> The starting point from which the extraction begins.

<end> The last instruction to be extracted. If omitted, AVATAR tries to extract to the end of the procedure referred to in <start>.

```
:avatar
AVATAR [2.0] - LPS Toolbox [A.01a]          (c) 1995 Lund Performance Solutions

For Help at the AVATAR prompt enter   ?


AVATAR: open myfile
-----> LST Module Directory <-----     Starts @    #Length
1  SCRASMBX                            $00003000     97964

AVATAR[myfile]: extract file1 $1000 $2000
extraction complete, 9 lines.

AVATAR[myfile]: :print file1
        BREAK     1,0
        LDB       816(0,0),0
        LDB       4(0,0),0
        BREAK     1,0
        BREAK     1,0
        LDB       818(0,0),0
        LDB       4(0,0),0
        BREAK     1,0
        BREAK     1,0
ok
AVATAR[myfile]: exit
:
```

Figure 16.3 - EXTRACT Command

Note: In the above example, AVATAR is used to extract code from hexadecimal offset $4000 to $4020. The extracted code is stored in ASCII format in the file called FILE1. The MPE PRINT command is used to display a portion of the file that was created.

Find <string> <filter> | **<symbol> <filter>**

The FIND command will find all entries in the symbol dictionary that have a name that matches or partially matches the provided string. Entries in the symbol dictionary include items like procedures, global data items, and intrinsics. The search is limited to the current SOM.

<string> Any string of ASCII characters enclosed in double quotes.

<symbol> Any string of valid symbol characters.

<filter> A symbol type to be filtered out. The default is that all symbols are listed. The possible filter values are listed below:

Filter Types

ABSOLUTE	EXTERNAL	NULL	STORAGE
ARG_EXT	LOCAL	OCT_DIS	STUB
CODE	MILLICODE	PLABEL	SYM_EXT
DATA	MILLI_EXT	PRI_PROG	UNIVERSAL
ENTRY	MODULE	SEC_PROG	UNSAT

The FIND command generates a display such as the one below.

```
:avatar
AVATAR [2.0] - LPS Toolbox [A.01a]          (c) 1995 Lund Performance Solutions

For Help at the AVATAR prompt enter   ?


AVATAR: open myfile
----> LST Module Directory <----     Starts @    #Length
1  SCRASMBX                          $00003000     97964

AVATAR[myfile]: find "" CODE

Module: SCRASMBX
Symbol                          X P Symbol     Symbol
Name                              Type       Scope      Address  Value
-------                         - - ------     -------    -------- --------
$START$                         3 3 CODE       UNIVERSAL  f1b4     93eb
_start                          3 3 CODE       UNIVERSAL  f1c8     980f
$UNWIND_START                   0 0 CODE       UNIVERSAL  f1dc     10950
$UNWIND_END                     0 0 CODE       UNIVERSAL  f1f0     111c0
$RECOVER_START                  0 0 CODE       UNIVERSAL  f204     11388
$RECOVER_END                    0 0 CODE       UNIVERSAL  f218     11388
show_file_opens                 0 0 CODE       LOCAL      f27c     94a8
printf                          3 3 CODE       UNIVERSAL  f2f4     96df
$hi$164                         0 0 CODE       LOCAL      f31c     9590
main                            3 3 CODE       UNIVERSAL  f330     9597
memcpy                          3 3 CODE       UNIVERSAL  f394     968b
$hi$179                         0 0 CODE       LOCAL      f3e4     9660
$hi$1                           0 0 CODE       LOCAL      f59c     96bc
_get_file_blocks                3 3 CODE       UNIVERSAL  f5c4     9f23
_cinit                          3 3 CODE       UNIVERSAL  f5d8     9b2b
_outfmt                         3 3 CODE       UNIVERSAL  f5ec     a0ef
_startc                         3 3 CODE       UNIVERSAL  f63c     b3d7
strlen                          3 3 CODE       UNIVERSAL  f6b4     fcff
```

Figure 16.4 - FIND Command

Note: To display all symbols, pass a Null string to the FIND command.

FINDAll <string> <filter> | symbol> <filter>

The FINDALL command will find all entries in the symbol dictionary that have a name that matches or partially matches the provided string. Entries in the symbol dictionary include all procedures, global data items, intrinsics, etc. The search will include all modules in the currently opened SOM file, unlike the FIND command (which searches only the current module).

This is extremely helpful when looking through an XL or even NL.PUB.SYS.

<string> Any string of ASCII characters.

<symbol> Any string of valid symbol characters.

<filter> A symbol type to be filtered out. The default is that all symbols are listed.

The possible filter values are noted in the Filter Types list for the FIND command. See the section on "Filter Types" in the FIND command for a complete list.

```
AVATAR[myfile]: findall "start" code

Module: SCRASMBX
Symbol                          X P Symbol     Symbol
Name                              Type       Scope      Address  Value
--------                        - - -------    ------     -------- --------
_start                          3 3 CODE       UNIVERSAL  f1c8     980f
_startc                         3 3 CODE       UNIVERSAL  f63c     b3d7
splstartup                      3 3 CODE       UNIVERSAL  fa38     b517
$na_restart                     0 0 CODE       LOCAL      12404    7924

AVATAR[myfile]:
```

Figure 16.5 - FINDALL Command

FIXup

The FIXUP command will display all the compiler fixup entries for the current SOM. For example:

```
:avatar
AVATAR [2.0] - LPS Toolbox [A.01a]          (c) 1995 Lund Performance Solutions

For Help at the AVATAR prompt enter   ?


AVATAR: open testx.obj
FILE TYPE : relocatable SOM

AVATAR[testx.obj]: fixup
need data ref        :             0
argument relocation  :             0
expression type      :             2 symbol1 - offset + const
execution level      :             0
format               :             3 17 bit word displacement
field                :             0 F%
subspace offset      :            88
symbol1 name         : GETPRIVMODE
fixup constant       :            -8

need data ref        :             0
argument relocation  :             0
expression type      :             1 symbol1 - symbol2 + const
execution level      :             0
format               :             1 21 bit immediate
field                :             3 L%
subspace offset      :            96
symbol1 name         : item_status
symbol2 name         : $global$
fixup constant       :             0


AVATAR[testx.obj]:
```

Figure 16.6 - FIXUP Command

Key to FIXUP Output

Symbol_type

ABSOLUTE	Absolute constant
ARG_EXT	Extension record of the current entry
CODE	Unspecified code, resolved at link time
DATA	Normal initialized data
ENTRY	Code entry point symbols
MILLICODE	Name of a millicode subroutine

Symbol_type

MILLI_EXT	Address of an external millicode subroutine
MODULE	Source module name
NULL	Invalid symbol record
OCT_DIS	Used by OCT (Object Code Translator)
PLABEL	Procedure label
PRI_PROG	Primary program entry point
SEC_PROG	Secondary program entry point
STORAGE	Storage requirement with known length & unknown value
STUB	External call stub, or relocation stub
SYM_EXT	Extension record of the current entry

Symbol_scope

EXTRNAL	Import request to a symbol in another SOM
LOCAL	Private symbol
UNIVERSAL	Symbol to be exported outside the SOM
UNSAT	Unsatisfied import request

Check_level Determines the type checking error level that the linker uses while binding external references to procedures and global variables. All object modules indicate a checking level for each reference and each definition of a procedure or a global variable. When binding an external reference to a definition, the linker compares the type information at the lower of the two checking levels specified by the reference and the definition. If a type mismatch is found, the linker reports it as either a warning or an error.

The values for check_level are:

- **0** All type mismatches are warnings.
- **1** Mismatches of the procedure, function, or variable type are errors; all other mismatches are warnings.
- **2** Mismatches of the procedure, function, or variable type, and mismatches of the number of arguments for procedures or functions are errors; all other mismatches (i.e., parameter types) are warnings.
- **3** All type mismatches are errors.

Must_qualify Used to indicate if more than one entry has the same symbol name.

- **0** Multiple symbol not present
- **1** Multiple symbol present

Initially_frozen Code using this symbol will be locked into physical memory when the operating system is booted.

- **0** Not frozen
- **1** Frozen

Memory_resident Indicates that the code that will use the symbol is frozen in memory. This provides a way for frozen procedures to communicate.

0 Not in memory
1 In memory

Is_common Used to indicate if a symbol is in an initialized common data area.

0 Not in common
1 In common

Duplicate_common Used to indicate if the source language allows duplicate initialization.

0 No allowed
1 Allowed

Xleast Execution level that is required to call this entry point.

0:
1:
2:
3:

Argument Relocation This fixup information is used to communicate the locations of the first four parameters, and the function return parameter. The linker used this information to match up exported symbol information with fixup references. The four possible values for this field are:

0 "Do not relocate"
1 "Argument register"
2 "Floating point coprocessor register, low"
3 "Floating point coprocessor register, high"

Code offset Offset into the SOM where "symbol name" code begins.

Privilege_level Determines the privilege level used by the executable program file. This parameter changes the privilege level of all procedures in the symbol and export tables (of the relocatable object file) that were set during compilation. The values for privilege_level are:

0 System level access
1 Unused
2 Privileged level access
3 User level access

FORMAT <data offset> <format> <count>

The FORMAT command displays data relative to the start of the SOM file in one of many different formats.

<data offset> An expression giving the offset from the start of the file, where formatting starts.

<format> The format specifier. The valid format specifiers are listed below. The composition of each format is detailed in the "Format Specifier Definitions" section that follows.

<count> An integer giving the number of elements to format.

Format Specifier List

AUX_HEADER_ID	INIT_REC	SPACE_REC	SYMDICT_ARG_REC
ARG_DESCRIPTOR	LST_BITS	SUBSPACE_BITS	SYMDICT_EXT_REC
COMPILER_REC	LST_HEADER	SUBSPACE_REC	UNWIND_DESCRIPT
FIXUP_BITS	LST_SYMBOL	SYMBOL_DICT_BITS	UNWIND_ENTRY
FIXUP_REC	SOM_HEADER	SYMBOL_DICT_REC	

Sample output using the format specifier SOM_HEADER:

```
:avatar
AVATAR [2.0] - LPS Toolbox [A.01a]          (c) 1995 Lund Performance Solutions

For Help at the AVATAR prompt enter   ?


AVATAR: open testx.obj
FILE TYPE : relocatable SOM

AVATAR[testx.obj]: format 0 som_header 1
       0:
                                system_id :          523       20b
                                  a_magic :          262       106
                               version_id :     85082112   5124000
                              file_time_a :            0         0
                              file_time_b :            0         0
                              entry_space :            0         0
                           entry_subspace :            0         0
                             entry_offset :            0         0
                      aux_header_location :          128        80
                          aux_header_size :            0         0
                             som_length :         3576       df8
                             presumed_dp :            0         0
                space_dictionary_location :          128        80
                   space_dictionary_total :            2         2
              subspace_dictionary_location :          200        c8
                 subspace_dictionary_total :            5         5
                     loader_fixup_location :          400       190
                        loader_fixup_total :            0         0
                  space_strings_location :          400       190
                      space_strings_size :           92        5c
                    init_array_location :            0         0
                       init_array_total :            0         0
              compiler_dictionary_location :         1288       508
                 compiler_dictionary_total :            1         1
               symbol_dictionary_location :         1324       52c
                  symbol_dictionary_total :           39        27
                   fixup_request_location :         2104       838
                      fixup_request_total :           41        29
```

```
                symbol_strings_location :         2924       b6c
                   symbol_strings_size :          652       28c
              unloadable_space_location :         2104       838
                  unloadable_space_size :            0         0
                               checksum :    119096742   71945a6

AVATAR[testx.obj]:
```

Figure 16.7 - SOM_HEADER Format

Sample output using the format specifier LST_HEADER:

```
AVATAR[testx.obj]: format 0 lst_header
       0:
                                system_id :          523       20b
                                  a_magic :          262       106
                               version_id :     85082112   5124000
                                file_time :            0         0
                                 hash_loc :            0         0
                                hash_size :            0         0
                             module_count :            0         0
                             module_limit :          128        80
                                  dir_loc :            0         0
                               export_loc :         3576       df8
                             export_count :            0         0
                               import_loc :          128        80
                                  aux_loc :            2         2
                                 aux_size :          200        c8
                               string_loc :            5         5
                              string_size :          400       190
                                free_list :            0         0
                                 file_end :          400       190
                                 checksum :           92        5c

AVATAR[testx.obj]:
```

Figure 16.8 - LST_HEADER Format

Format Specifier Definitions

The LST_HEADER is composed of the following elements:

system_id
a_magic
version_id
file_time
hash_loc
hash_size
module_count
module_limit
dir_loc
export_loc
export_count
import_loc
aux_loc
aux_size
string_loc
string_size
free_list
file_end
checksum

The LST_BITS is composed of the following elements:

hidden
symbol_type
symbol_scope
check_level
must_qualify
initially_frozen
memory_resident
is_common
duplicate_common
xleast
arg_relocation

The LST_SYMBOL is composed of the following elements:

ls_bits
symbol_name_ptr
qualifier_name_ptr
symbol_info
symbol_descriptor
max_num_args
min_num_args
num_args
som_index
symbol_key
next_entry

The SOM_HEADER is composed of the following elements:

system_id
a_magic
version_id
file_time_a
file_time_b
entry_space
entry_subspace
entry_offset
aux_header_location
aux_header_size
som_length
presumed_dp
space_dictionary_location
space_dictionary_total
subspace_dictionary_location
subspace_dictionary_total
loader_fixup_location
loader_fixup_total
space_strings_location
space_strings_size
init_array_location
init_array_total
compiler_dictionary_location
compiler_dictionary_total
symbol_dictionary_location
symbol_dictionary_total
fixup_request_location
fixup_request_total
symbol_strings_location
symbol_strings_size
unloadable_space_location
unloadable_space_size
checksum

The AUX_HEADER_ID is composed of the following elements:

mandatory
copy
append
ignore
type
length

The SPACE_REC is composed of the following elements:

name_pointer
access_bits
sort_key
number
subspace_index
subspace_quantity
loader_fixup_index
loader_fixup_quantity
init_pointer_index
init_pointer_quantity

The SUBSPACE_BITS is composed of the following elements:

acb
memory_resident
duplicate_common
is_common
is_loadable
quadrant
initially_frozen
code_only
sort_key
replicate_init
continuation

The SUBSPACE_REC is composed of the following elements:

- space_index
- bits
- file_loc
- init_length
- startfixup_request_qty
- length
- reserved
- alignment
- name_pointer
- fixup_request_index

The COMPILER_REC is composed of the following elements:

- name
- language_name
- product_id
- version_id
- compile_time
- source_time

The FIXUP_BITS is composed of the following elements:

- need_data_reference
- arg_relocation
- expression_type
- execution_level
- fixup_format
- fixup_field

The FIXUP_REC is composed of the following elements:

- bits
- space_offset
- symbol_index_one
- symbol_index_two
- fixup_constant

The INIT_REC is composed of the following elements:

- space_index
- access_control
- has_data
- memory_resident
- initially_frozen
- new_locality
- file_location
- length
- offset

The SYMBOL_DICT_BITS is composed of the following elements:

- bits
- symbol_name_ptr
- qualifier_name_ptr
- symbol_info
- symbol_value

The ARG_DESCRIPTOR is composed of the following elements:

- packing
- alignment
- mode
- structure
- hash
- data_type

The SYMBOL_DICT_REC is composed of the following elements:

bits	symbol_info
symbol_name_ptr	symbol_value
qualifier_name_ptr	

The SYMDICT_EXT_REC is composed of the following elements:

type	symbol_descriptor
max_num_args	arg_descriptor[1]
min_num_args	arg_descriptor[2]
num_args	arg_descriptor[3]

The SYMDICT_ARG_REC is composed of the following elements:

type	arg_descriptor[3]
arg_descriptor[1]	arg_descriptor[4]
arg_descriptor[2]	

The UNWIND_DESCRIPT is composed of the following elements:

cannot_unwind	call_gr
millicode	save_sp
millicode_save_sr0	save_rp
region_description	save_mrp_in_frame
save_srs	cleanup_defined
entry_fr	hpe_interrupt_marker
entry_gr	hpux_interrupt_marker
args_stored	large_frame_r3
call_fr	total_framesize

The UNWIND_ENTRY is composed of the following elements:

starting_offset
ending_offset

HELP

The HELP command invokes AVATAR Help.

Init

The INIT command will display all the compiler initialization entries for the current SOM. This information comes from the INIT_REC portion of the SOM.

```
AVATAR[testx.obj]: init
Space Acc DRFN File loc Offset   Length
34275590 002 D... 00000000 00000000 00000000

AVATAR[testx.obj]:
```

Figure 16.9 - INIT Command

Space = index into the space dictionary

Acc = access control bits, valid values from 0-7

- 0 : read only data page
- 1 : normal data page
- 2 : normal code page
- 3 : dynamic code page
- 4 : gateway to PL0
- 5 : gateway to PL1
- 6 : gateway to PL2
- 7 : gateway to PL3

DRFN = bit encoded flag

- D : data defined in SOM for this space
- R : this subspace is locked into physical memory once the subspace goes into execution
- F : this subspace is locked into physical memory when the operating system is booted
- N : initialization pointers starts a new locality set

File-loc = if 'D' (data defined in SOM), then this entry points at the data to initialize one or more subspaces. Otherwise this field contains a 32-bit value used to initialize one or more subspaces

Length = if 'D', then this field contains the byte length of the area to initialize. Otherwise it is undefined.

Offset = byte offset where initialization is to start, relative to the start of the space.

Look [<string>]
[<symbol>]

The LOOK command will display all the available information about a given entry in the symbol dictionary. For example, this command can be used to find the argument types and order for any procedure, anywhere. Parameters for this command are case-sensitive.

<string> Any string of ASCII characters.

<symbol> Any string of valid symbol characters.

```
:avatar
AVATAR [2.0] - LPS Toolbox [A.01a]          (c) 1995 Lund Performance Solutions

For Help at the AVATAR prompt enter   ?


AVATAR: open nl.pub.sys
Assuming space $A for NL.PUB.SYS
----> LST Module Directory <-----     Starts @   #Length
1  HP31900                            $00104000 20109316
2  HP30600                            $01432000    65392
3  NIOCDM                             $01442000   167496
4  CONSOLE                            $0146b000   116348
5  LANCELOT                           $01488000   318400
6  NMS                                $014d6000   137440
7  HP32022                            $014f8000   818616
8  B3175A                             $015c0000    17868
9  apatch4.assemblr                   $015c5000    27280
10  PSICOMN                           $015cc000   175584
11  DCC                               $015f7000   355816
12  DLPI                              $0164e000    67044
13  STEALTH                           $0165f000   112504
14  LSS                               $0167b000    65388
15  NMA                               $0168b000    60660
```

```
16  SNMP                                $0169a000   299968
17  NSCORE                              $016e4000   103248
18  NSXPORT                             $016fe000  1317824
19  RPM                                 $01840000    46928
20  STREAM01                            $0184c000   284812
21  VTCORE                              $01892000   105028
22  NMS                                 $018ac000   110180
23  STREAM02                            $018c7000    24948
24  NMS                                 $018ce000    67632
25  COMPRESS                            $018df000    22268
26  TSTORE                              $018e5000    23652
27  MEDIAMGR                            $018eb000   126160
28  HEP                                 $0190a000    83040
Select a module number > 1

      Module #   1: HP31900
      Found 16441 unwind entries.
      Searching 88607 symbol dictionary entries
      Sorting 55562 symbols
FILE TYPE : sharable, executable SOM

AVATAR[nl.pub.sys]: look HPFOPEN
symbol name          : HPFOPEN
  address            : 1087c20
  symbol_type        : unspecified code
  symbol_scope       : exported symbol for other SOMs
  check_level        : 3
  must_qualify       : 0
  initially_frozen   : 1
  memory_resident    : 0
  is_common          : 0
  duplicate_common   : 0
  xleast             : 3
  privilege level    : 0
  code offset        : ed27dc  -  ed31b0    (630 instructions)

  procedure header   :
    packing          : XL packing and IEEE reals
    alignment        : byte aligned
    type             : procedure
    structure        : procedure
    data type        : hashed (1dd8) = (void) ?

  parameter #1       :
    packing          : XL packing and IEEE reals
    alignment        : word aligned
    type             : parameter, passed by value
    structure        : simple variable
    data type        : hashed (0d6a) = (anyvar size) ?

  parameter #2       :
    packing          : XL packing and IEEE reals
    alignment        : byte aligned
    type             : parameter, passed by reference
    structure        : simple variable
    data type        : hashed (4bb5)

  parameter #3       :
    packing          : XL packing and IEEE reals
    alignment        : byte aligned
    type             : parameter, passed by reference
    structure        : simple variable
    data type        : hashed (0d6a) = (anyvar size) ?

  parameter #4       :
    packing          : XL packing and IEEE reals
```

```
    alignment        : word aligned
    type             : parameter, passed by value
    structure        : simple variable
    data type        : signed word (32 bits)


  parameter #5      :
    packing          : XL packing and IEEE reals
    alignment        : byte aligned
    type             : parameter, passed by reference
    structure        : array
    data type        : hashed (765a)

  parameter #6      :
    packing          : XL packing and IEEE reals
    alignment        : word aligned
    type             : parameter, passed by value
    structure        : simple variable
    data type        : signed word (32 bits)


               ...

AVATAR[nl.pub.sys]:
```

Figure 16.10 - look HPFOPEN

```
AVATAR[testx.obj]: look item_status
symbol name          : item_status
  address            : 554
  symbol_type        : uninitialized data
  symbol_scope       : unsatisfied import request
  check_level        : 0
  must_qualify       : 0
  initially_frozen   : 0
  memory_resident    : 0
  is_common          : 0
  duplicate_common   : 0
  xleast             : 3
  storage req len    : fa0

AVATAR[testx.obj]:
```

Figure 16.11 - LOOK Item-Status

LSt

The LST command will display all the available modules in the currently opened SOM file and allow you to select a module. When a SOM with multiple entries is OPENed, the LST command is automatically invoked. For example, only SOMs with multiple entries require the use of the LST command.

Example: Only SOMs with multiple entries require the use of the LST command.

```
:avatar
AVATAR [2.0] - LPS Toolbox [A.01a]          (c) 1995 Lund Performance Solutions

For Help at the AVATAR prompt enter   ?


AVATAR: open xl
----> LST Module Directory <-----     Starts @    #Length
1  hp30026_01                         $00071000    263744
2  HP30138                            $000b2000    971640
```

```
3   STIS209S                          $001a0000    23228
4   RLBPROCS                          $001a6000   308792
5   HP35360                           $001f2000   133108
6   SENTRYTI                          $00213000   853948
7   HP36395                           $002e4000    69008
8   SJAS424S                          $002f5000   131488
9   LIB1SRC                           $00316000   134036
10  SPECAUX                           $00337000   185892
11  HP315021                          $00365000   165332
12  HP31511_01                        $0038e000   107908
13  HP32715                           $003a9000   443348
14  HP36957                           $00416000   249952
15  HP36961                           $00454000    62456
16  PSICOMN                           $00464000   640984
17  STEALTH                           $00501000    43332
18  FMT                               $0050c000   539620
19  AHPDINT                           $00590000    31732
20  LANCELOT                          $00598000   335496
21  corelib_01                        $005ea000    24952
22  LSS                               $005f1000    18452
23  SNMP                              $005f6000    82456
24  NSR                               $0060b000   403772
25  TELNETO                           $0066e000   256736
26  SOCKET                            $006ad000   241576
27  PSPNMSTB                          $006e8000    36368
28  S01STLIB                          $006f1000    21120
29  NMEVNT                            $006f7000    13680
30  SV1S209X                          $006fb000   728728
31  S25S391C                          $007ad000    17420
32  HP31501_02                        $007b2000   179812
33  HP315022                          $007de000    53700
34  corelib_02                        $007ec000    57884
35  S0FP935N                          $007fb000    11360
36  S27S391C                          $007fe000    31208
37  HP31501_03                        $00806000   302264
38  U_Qfabs                           $00850000    75304
39  SZAS393S                          $00863000    18608
40  S29S391C                          $00868000    17452
41  dbcore.p                          $0086d000  1362068
42  APALTERS                          $009ba000   409752
43  HPSQL2                            $00a1f000  2184008
44  HPSQL3                            $00c35000   751208
45  HPSQL4                            $00ced000   223244
46  HPSQL5                            $00d24000   501136
47  HPSQL6                            $00d9f000   111860
48  HPSQL7                            $00dbb000    35452
49  HPSQL8                            $00dc4000   102028
50  HPSQL9                            $00ddd000     9272
51  HP31900                           $00de0000   167440
52  B3821A1                           $00e09000    18552
53  B3821A2                           $00e0e000    22968
Select a module number > 1

      Module #   1: hp30026_01
      Found 395 unwind entries.
      Searching 3185 symbol dictionary entries
      Sorting 3185 symbols
FILE TYPE : sharable, executable SOM

AVATAR[x1]: lst
-----> LST Module Directory <-----    Starts @   #Length
1   hp30026_01                        $00071000   263744
2   HP30138                           $000b2000   971640
3   STIS209S                          $001a0000    23228
4   RLBPROCS                          $001a6000   308792
5   HP35360                           $001f2000   133108
```

```
6   SENTRYTI                                $00213000    853948
7   HP36395                                 $002e4000     69008
8   SJAS424S                                $002f5000    131488
9   LIB1SRC                                 $00316000    134036
10   SPECAUX                                $00337000    185892
11   HP315021                               $00365000    165332
12   HP31511_01                             $0038e000    107908
13   HP32715                                $003a9000    443348
14   HP36957                                $00416000    249952
15   HP36961                                $00454000     62456
16   PSICOMN                                $00464000    640984
17   STEALTH                                $00501000     43332
18   FMT                                    $0050c000    539620
19   AHPDINT                                $00590000     31732
20   LANCELOT                               $00598000    335496
21   corelib_01                             $005ea000     24952
22   LSS                                    $005f1000     18452
23   SNMP                                   $005f6000     82456
24   NSR                                    $0060b000    403772
25   TELNETO                                $0066e000    256736
26   SOCKET                                 $006ad000    241576
27   PSPNMSTB                               $006e8000     36368
28   S01STLIB                               $006f1000     21120
29   NMEVNT                                 $006f7000     13680
30   SV1S209X                               $006fb000    728728
31   S25S391C                               $007ad000     17420
32   HP31501_02                             $007b2000    179812
33   HP315022                               $007de000     53700
34   corelib_02                             $007ec000     57884
35   S0FP935N                               $007fb000     11360
36   S27S391C                               $007fe000     31208
37   HP31501_03                             $00806000    302264
38   U_Qfabs                                $00850000     75304
39   SZAS393S                               $00863000     18608
40   S29S391C                               $00868000     17452
41   dbcore.p                               $0086d000   1362068
42   APALTERS                               $009ba000    409752
43   HPSQL2                                 $00a1f000   2184008
44   HPSQL3                                 $00c35000    751208
45   HPSQL4                                 $00ced000    223244
46   HPSQL5                                 $00d24000    501136
47   HPSQL6                                 $00d9f000    111860
48   HPSQL7                                 $00dbb000     35452
49   HPSQL8                                 $00dc4000    102028
50   HPSQL9                                 $00ddd000      9272
51   HP31900                                $00de0000    167440
52   B3821A1                                $00e09000     18552
53   B3821A2                                $00e0e000     22968
Select a module number > 3

      Module #   3: STIS209S
      Found 81 unwind entries.
      Searching 160 symbol dictionary entries
      Sorting 160 symbols
FILE TYPE : sharable, executable SOM

AVATAR[x1]:
```

Figure 16.12 - LST Command

MC <expression> <value>

The MC command is used to modify data at a given code address. The address can be specified as any valid expression for the current SOM. The new value can be any valid value for the code space, including assembler instructions.

<expression> Is an arithmetic expression that represents an offset to the start of the current module.

<value> Is an expression representing the new data.

```
AVATAR: mc test_while_for "BL FOPEN,2"

EN_test_while_for          0  0  CODE          LOCAL  43fc   58b0
[C]          58B0:       6bc23fd9  STW          2, -20(0,30)
```

Figure 16.13 - MC Command

MD <expression> <value>

The MD command modifies values from an initialized data area within the current SOM.

<expression> Is an expression that represents an offset into the initialized data area in the current SOM.

<value> Is an expression representing the new data.

```
AVATAR: md _db_area+123
More than one symbol qualifies, please select:
1:                         DATA          LOCAL  43c0   40000008
2:                         DATA          LOCAL  44b0   40000008

AVATAR:
```

Figure 16.14 - MD Command

MV <expression> <value>

The MV command will modify data values within the current SOM.

<expression> Is an arithmetic expression that represents an offset to the start of the current file.

<value> Is an expression representing the new data.

```
AVATAR: mv 0 100
[V]        0: 00000100   6c756465  203c7374   64696f2e  .... lude <stdio.
AVATAR: mv  0  $dc
[V]        0: 000000dc   6c756465  203c7374   64696f2e  .... lude <stdio.
```

Figure 16.15 - MV Command

Next

The NEXT command repeats the last DV, DC or DD command starting from the point where it left off. *Note*: An MC, MV or MD command will reposition the pointer as well.

Also see: EATEMPTY and CRON in the "*Standard Commands and Options*" section.

Open <filename> [READ]

filename must be a valid MPE/iX file descriptor (no wildcards). If the specified file is a Native Mode executable file (SOM) and it contains more than one LST entry, then the user is asked to select which LST entry to analyze. If the specified file is not an SOM, then the file is mapped into virtual memory where the user is restricted to basic operations, such as displaying and modifying memory. If the "Read" option is specified then the file is opened for Read-Access only. The default is read/write access.

Quit

The QUIT command exits the program.

Radix <mnemonic>

The RADIX command changes the default radix, which is the base that all input is assumed to use.

<mnemonic> = [DECIMAL]
[HEX]
[OCTAL]

Search <expression>
<string>

The SEARCH command will search the entire opened SOM for a given value of a string. If the string is found, a line of data is displayed.

<expression > Is an expression that results in an integer value to be searched for in the file.

<string> Is a quoted string to be searched for in the file.

Examples of the SEARCH command for various strings:

Search $8861
Search "A string"
Search `LDIL $1000,0`

Searching an ASCII file:

```
:avatar
AVATAR [2.0] - LPS Toolbox [A.01a]          (c) 1995 Lund Performance Solutions

For Help at the AVATAR prompt enter   ?


AVATAR: open testx.c

AVATAR[testx.c]: search "printf"
[V]    1d14: 20207072 696e7466 28224149 4646494c   printf("AIFFIL
[V]    1ea4: 20207072 696e7466 28224149 4646494c   printf("AIFFIL
[V]    26c0: 20207072 696e7466 28227465 73742052   printf("test R
[V]    28f4: 20207072 696e7466 28224149 46414343   printf("AIFACC
[V]    2b20: 20207072 696e7466 28225c6e 4c6f6f6b   printf("\nLook

AVATAR[testx.c]: exit
:
```

Figure 16.16 - SEARCH Command

Searching a Native Mode program file:

```
:avatar
AVATAR [2.0] - LPS Toolbox [A.01a]          (c) 1995 Lund Performance Solutions

For Help at the AVATAR prompt enter   ?

AVATAR: open testx
----> LST Module Directory <----     Starts @    #Length
1  SCRASMBX                          $00004000     98008

AVATAR[testx]: search $08000240
[V]     8030: 08000240 873f235a 8f3e8242 0b40561d ...@.?#Z.>.B.@V.
[V]     815c: 08000240 e8194002 08000240 b3202000 ...@..@....@.  .
[V]     8164: 08000240 b3202000 08000240 ebe0c000 ...@.  ....@....
[V]     816c: 08000240 ebe0c000 081a025d e8001c2e ...@.......]....
[V]     817c: 08000240 e8001c9e 08000240 e8001c2e ...@.......@....
[V]     8184: 08000240 e8001c2e 08000240 e8001d06 ...@.......@....
[V]     818c: 08000240 e8001d06 08000240 e8001d7e ...@.......@...~
[V]     8194: 08000240 e8001d7e 08000240 e801008a ...@...~...@....
[V]     819c: 08000240 e801008a 08000240 e8001c0e ...@.......@....
[V]     81a4: 08000240 e8001c0e 08000240 e80101da ...@.......@....


AVATAR[testx]: exit
:
```

Figure 16.17 - SEARCH a Native Mode Program

SET | REset

The SET/RESET command enables and disables options within AVATAR.

SPace

The SPACE command will display space and subspace information about the current SOM.

```
:avatar
AVATAR [2.0] - LPS Toolbox [A.01a]          (c) 1995 Lund Performance Solutions

For Help at the AVATAR prompt enter   ?


AVATAR: open testx.obj
FILE TYPE : relocatable SOM

AVATAR[testx.obj]: space
SPACE RECORD
  space_name          : $TEXT$
  space_number        : 0
AC  MRCLFPIA Q SK Loc/Init InitLn    Start    Length  Align Fidx Fqty
2c  ...L.... 0 00 000001f0 000000 00000000 00000000 000008 ffff 0000  $LIT$
2c  ...L.P.. 0 00 000001f0 000210 00000000 00000210 000008 0000 0025  $CODE$
2c  ...L.... 0 40 00000400 000020 00000210 00000020 000008 0025 0004  $UNWIND$
SPACE RECORD
  space_name          : $PRIVATE$
  space_number        : 0
AC  MRCLFPIA Q SK Loc/Init InitLn    Start    Length  Align Fidx Fqty
1f  ...L.... 1 00 00000420 0000e8 40000000 000000e8 000008 ffff 0000  $DATA$
1f  ...L.... 1 00 00000000 000000 400000e8 00000000 000008 ffff 0000  $BSS$

AVATAR[testx.obj]: exit
:
```

Figure 16.18 - SPACE Command

STatistics

The STATISTICS command produces the following display.

```
:avatar
AVATAR [2.0] - LPS Toolbox [A.01a]          (c) 1995 Lund Performance Solutions

For Help at the AVATAR prompt enter   ?


AVATAR: open testx
-----> LST Module Directory <-----     Starts @   #Length
1  SCRASMBX                            $00004000    98008

AVATAR[testx]: stat
Module: SCRASMBX
            scope        UNSAT      LOCAL  EXTERNAL UNIVERSAL      TOTAL
      DATA       :            0        359         0        29        388
                              0      11540         0       960      12500
      CODE       :            0        419         0        48        467
                              0      12788         0      1580      14368
      PRI_PROG   :            0          0         0         1          1
                              0          0         0        32         32
      ENTRY      :            0         10         0        43         53
                              0        328         0      1396       1724
      STUB       :            0          0        24         0         24
                              0          0       804         0        804
      MILLICODE  :            0          0         0       101        101
                              0          0         0      3564       3564
      PLABEL     :            0          1         0         0          1
                              0         28         0         0         28

      TOTAL      :            0        789        24       222       1035
                              0      24684       804      7532      33020


AVATAR[testx]: exit
:
```

Figure 16.19 - STATISTICS Command

STRIP

The STRIP command will remove symbolic information from SOM.

```
:avatar
AVATAR [2.0] - LPS Toolbox [A.01a]          (c) 1995 Lund Performance Solutions

For Help at the AVATAR prompt enter   ?


AVATAR: open testx
-----> LST Module Directory <-----     Starts @   #Length
1  SCRASMBX                            $00004000    98008

AVATAR[testx]: strip
Stripping 13,972 bytes of symbol names.

AVATAR[testx]: exit
:
```

Figure 16.20 - STRIP Command

SUbspace

The SUBSPACE command will display subspace information on the current SOM.

```
:avatar
AVATAR [2.0] - LPS Toolbox [A.01a]          (c) 1995 Lund Performance Solutions

For Help at the AVATAR prompt enter   ?


AVATAR: open testx.obj
FILE TYPE : relocatable SOM

AVATAR[testx.obj]: subspace
AC  MRCLFPIA Q SK Loc/Init InitLn    Start   Length  Align Fidx Fqty
2c  ...L.... 0 00 000001f0 000000 00000000 00000000 000008 ffff 0000  $LIT$
2c  ...L.P.. 0 00 000001f0 000210 00000000 00000210 000008 0000 0025  $CODE$
2c  ...L.... 0 40 00000400 000020 00000210 00000020 000008 0025 0004  $UNWIND$
1f  ...L.... 1 00 00000420 0000e8 40000000 000000e8 000008 ffff 0000  $DATA$
1f  ...L.... 1 00 00000000 000000 400000e8 00000000 000008 ffff 0000  $BSS$

AVATAR[testx.obj]: exit
:
```

Figure 16.21 - SUBSPACE Command

SYMformat <HEADER | LNTT | SLT | VT >

This command is used to format and display various portions of the SYMOS file. At this time only 4 of the listed commands have been implemented. Using this command requires an expert-level knowledge of the MPE/iX operating system. The commands that are currently available are the following:

<HEADER>	This command is used to format and display header information.
<LNTT>	This command is used to format and display the LNTT information.
<SLT>	This command is used to format and display the SLT information.
<VT>	This command is used to format and display the VT information.

SYM-based Examples

These Examples illustrate usage for SYMOPEN, SYMFORMAT HEADER, SYMFORMAT VT, and SYMFORMAT LNTT.

```
:avatar
AVATAR [2.0] - LPS Toolbox [A.01a]          (c) 1995 Lund Performance Solutions

For Help at the AVATAR prompt enter   ?


AVATAR: symopen symos.osb79.telesup
Found $DEBUG$ space, initializing debug info

AVATAR: symformat header
                          num procedures :          128        80
                               num files :          128        80
                             num modules :            0         0
                  pre-processed by pxdb : TRUE
                             big header : TRUE
                              sa header : TRUE
                             old globals :            1         1
                                 globals :        24615      6027
                                    time :            0         0
                               pg_entries :            0         0
                                   num 7 :            0         0
                                   num 8 :            0         0
                                   num 9 :            0         0
                                  num 10 :            0         0
                                  num 11 :            0         0

                                lntt index :        10971      2adb
                              instructions :        27580-    2758c
                                alias name :            0
                                      name :       722707   DAT__LOAD
                                statements :        27584-    27588
                                   num 9 :              1         1

                                lntt index :        20293      4f45
                              instructions :        275ac-    275b8
                                alias name :            0
                                      name :       d896d6   DAT__STMGT
                                statements :        275b0-    275b4
                                   num 9 :              1         1


AVATAR: symformat lntt
    0: Source File      "DHEADNM.HPESTD.OFFICIAL"          Slt=0   Lang = PASCAL
    1: Source File      "GACD.SYMTOOLS.OFFICIAL" Slt=1     Lang = PASCAL
    2: Source File      "GACD.SYMTOOLS.OFFICIAL" Slt=2     Lang = PASCAL
    3: Source File      "DHPEARCH.HPESTD.OFFICIAL"         Slt=3   Lang = PASCAL
    4: Source File      "GACD.SYMTOOLS.OFFICIAL" Slt=4     Lang = PASCAL
    5: Source File      "DHPEOS.HPESTD.OFFICIAL" Slt=5     Lang = PASCAL
    6: Source File      "GACD.SYMTOOLS.OFFICIAL" Slt=6     Lang = PASCAL
    7: Source File      "DHPEUSER.HPESTD.OFFICIAL"         Slt=7   Lang = PASCAL
    8: Source File      "GACD.SYMTOOLS.OFFICIAL" Slt=8     Lang = PASCAL
    9: Source File      "DKSOBJ.HPESTD.OFFICIAL" Slt=9     Lang = PASCAL
    a: Source File      "GACD.SYMTOOLS.OFFICIAL" Slt=a     Lang = PASCAL
    b: Source File      "DHPESTAT.HPESTD.OFFICIAL"         Slt=b   Lang = PASCAL
    c: Source File      "GACD.SYMTOOLS.OFFICIAL" Slt=c     Lang = PASCAL
    d: Source File      "DPTPRIM.PORTS.OFFICIAL" Slt=d     Lang = PASCAL
    e: Source File      "GACD.SYMTOOLS.OFFICIAL" Slt=e     Lang = PASCAL
    f: Source File      "DPTWAITQ.PORTS.OFFICIAL"          Slt=f   Lang = PASCAL
   10: Source File      "GACD.SYMTOOLS.OFFICIAL" Slt=10    Lang = PASCAL
   11: Source File      "DUTIL.UTIL.OFFICIAL"    Slt=11    Lang = PASCAL
   12: Source File      "GACD.SYMTOOLS.OFFICIAL" Slt=12    Lang = PASCAL
```

```
   13: Source File      "DSYSGLOB.HPESTD.OFFICIAL"        Slt=13  Lang = PASCAL
   14: Source File      "GACD.SYMTOOLS.OFFICIAL" Slt=14   Lang = PASCAL
   15: Source File      "DREALGLB.HPESTD.OFFICIAL"        Slt=15  Lang = PASCAL
   16: Source File      "GACD.SYMTOOLS.OFFICIAL" Slt=16   Lang = PASCAL
AVATAR: symformat vt
[       1] :*
[       3] :GACD.SYMTOOLS.OFFICIAL
[      1a] :DHEADNM.HPESTD.OFFICIAL
[      32] :GACD.SYMTOOLS.OFFICIAL
[      49] :DHPEARCH.HPESTD.OFFICIAL
[      62] :DHPEOS.HPESTD.OFFICIAL
[      79] :DHPEUSER.HPESTD.OFFICIAL
[      92] :DKSOBJ.HPESTD.OFFICIAL
[      a9] :DHPESTAT.HPESTD.OFFICIAL
[      c2] :DPTPRIM.PORTS.OFFICIAL
[      d9] :DPTWAITQ.PORTS.OFFICIAL
[      f1] :DUTIL.UTIL.OFFICIAL
[     105] :DSYSGLOB.HPESTD.OFFICIAL
[     11e] :DREALGLB.HPESTD.OFFICIAL
[     137] :DICS.HPESTD.OFFICIAL
[     14c] :DPSD.HPESTD.OFFICIAL
[     161] :DSYSFAIL.HPESTD.OFFICIAL
[     17a] :DSTRTYPE.LLIOMSG.OFFICIAL
[     194] :DKSPORT.PORTS.OFFICIAL
[     1ab] :DVSM.VSM.OFFICIAL
[     1bd] :DVSMPM.VSM.OFFICIAL
[     1d1] :DOBJCL.VSM.OFFICIAL
[     1e5] :DTAB.TBLMGT.OFFICIAL
AVATAR: exit
:
```

Figure 16.22 - SYMOPEN/SYMFORMAT: LNTT, VT

SYMOpen <symos name>

This AVATAR command lets you open a SYMOS file for investigation with the SYMFORMAT command. SYMOS files are the compiled data structure definitions for MPE/iX. Typically, they are used by HP engineers to symbolically debug operating system problems. Using this command implies a vast knowledge of the MPE/iX operating system.

SYn [SPLASH]
[SYSTEM]
[NONE]
[REG <general register number> <synonym>]
[CR <control register number> <synonym>]
[R]

The SYN command sets up synonyms for the general registers and the special registers. These synonyms will be shown in the disassembled format of any instruction.

<general register number>	A number between 0 and 31 inclusive, designating the general register for which a synonym is set up.
<special register number>	A number between 0 and 31 inclusive, designating the special register for which a synonym is set up.
<synonym>	Any name used as a synonym.
NONE	Resets all synonyms.
SYSTEM	Sets synonyms to reflect the normal system usage for registers.
SPLASH	Sets synonyms to reflect the normal SPLash! usage for synonyms.
R	Displays a letter "R" in front of the register during disassembly.

UNCALLED [N]

This command is used to display entry points that are called less than or equal to "N" times.

```
AVATAR: open cprog
1      C2SPL
Found 8 unwind entries.
Searching 74 symbol dictionary entries
Sorting 74 symbols
AVATAR: uncalled 0

Checked 1091 locations; found 0 calls and 0 stmnts.

Module: C2SPL
Symbol                  X   P   Symbol     Symbol
Name                            Type       Scope          Address      Value
------                  -   -   ------     ------         -------      -----
$START$                 3   3   CODE       UNIVERSAL      41a4         5113
main                    3   3   CODE       UNIVERSAL      426c         516b
test_func               3   3   CODE       UNIVERSAL      4280         5187
callsplash              3   3   CODE       UNIVERSAL      42d0         53b3
callspash2              3   3   CODE       UNIVERSAL      4370         558b
test_while_for          3   3   CODE       UNIVERSAL      4410         58b3
_start                  3   3   CODE       UNIVERSAL      44ec         5ea3
_init_c_globals         3   3   CODE       UNIVERSAL      4730         5f4b
$UNWIND_START           0   0   CODE       UNIVERSAL      41B8         6048
$UNWIND_END             0   0   CODE       UNIVERSAL      41cc         60c8
$RECOVER_END            0   0   CODE       UNIVERSAL      47f4         6108
$RECOVER_START          0   0   CODE       UNIVERSAL      41e0         6108
```

Figure 16.23 - UNCALLED Command

UNWind

This command is used to display the unwind descriptors for the current SOM.

```
AVATAR: open cprog
1  C2SPL
Found 8 unwind entries.
Searching 74 symbol dictionary entries
Sorting 74 symbols

AVATAR: unwind

                                        Starting      Ending        Total
Procedure Name                          Offset        Offset        Frame Size
--------------                          --------      ------        ----------
1$START$                                00005110      0000512c      00000010
2main                                   00005168      00005180      00000006
3test_func                              00005184      000052f4      0000000c
4EN_callsplash                          000053b0      00005584      00000247
5EN_callsplash2                         00005588      00005808      00000247
6EN_test_while_for                      000058b0      00005db8      0000024a
7_start                                 00005ea0      00005f24      0000022c
8_init_c_globals                        00005f48      00006044      0000000f
```

Figure 16.24 - UNWIND Command

AVATAR Examples

Figure 16.25 uses the FIND command to locate external procedure calls. Remember, most symbol parameters are case-sensitive.

```
:avatar
AVATAR [2.0] - LPS Toolbox [A.01a]          (c) 1995 Lund Performance Solutions

For Help at the AVATAR prompt enter   ?


AVATAR: open fscheck.mpexl.telesup
----> LST Module Directory <----     Starts @    #Length
1  OCQUEUE                           $00006000    199120

AVATAR[fscheck.mpexl.telesup]: find "" EXTERNAL

Module: OCQUEUE
Symbol                          X P Symbol     Symbol
Name                              Type       Scope      Address  Value
-------                         - - ------     ------     -------- -------
P_NEW_HEAP                      0 2 STUB       EXTERNAL   262d4    c7ba
P_DISPOSE_HEAP                  0 2 STUB       EXTERNAL   26324    c84a
FCHECK                          0 2 STUB       EXTERNAL   26a2c    d2da
FERRMSG                         0 2 STUB       EXTERNAL   26a40    d2fa
PRINTFILEINFO                   0 2 STUB       EXTERNAL   26a54    d31a
PRINT                           0 2 STUB       EXTERNAL   26a68    d33a
FGETINFO                        0 2 STUB       EXTERNAL   26acc    d412
CCODE                           0 2 STUB       EXTERNAL   26ae0    d432
U_set_escapecode                0 2 STUB       EXTERNAL   26af4    d452
FFILEINFO                       0 2 STUB       EXTERNAL   26b08    d492
FRELATE                         0 2 STUB       EXTERNAL   26b30    d4b2
U_get_escapecode                0 2 STUB       EXTERNAL   26b44    d472
FCONTROL                        0 2 STUB       EXTERNAL   26ba8    d986
FREAD                           0 2 STUB       EXTERNAL   26c0c    df62
genmsg                          0 2 STUB       EXTERNAL   26c5c    e252
WHO                             0 2 STUB       EXTERNAL   26d38    e342
FCLOSE                          0 2 STUB       EXTERNAL   26d9c    e7a2
U_nonlocal_escape               0 2 STUB       EXTERNAL   26db0    e7c2
P_STRWRITECHR                   0 2 STUB       EXTERNAL   26e28    e8be
P_STRWRITESTR                   0 2 STUB       EXTERNAL   26e3c    e8de
HPFOPEN                         0 2 STUB       EXTERNAL   26e50    e8fe


AVATAR[fscheck.mpexl.telesup]:
```

Figure 16.25 - FIND Command (External)

Occasionally, you may want to know what external procedures a program calls. The FIND command can easily locate all external procedure calls.

Figure 16.26 uses the LOOK command to determine parameter types.

```
AVATAR[fscheck.mpexl.telesup]: look direcfind
symbol name          : direcfind
  address            : 28084
  symbol_type        : stub
  symbol_scope       : import request from another SOM
  check_level        : 1
  must_qualify       : 0
  initially_frozen   : 0
  memory_resident    : 0
  is_common          : 0
  duplicate_common   : 0
  xleast             : 0
  xrt offset         : ee0
```

```
  privilege level    : 2
  code offset        : 15c5c

  procedure header   :
    packing          : XL packing and IEEE reals
    alignment        : word aligned
    type             : function return
    structure        : simple variable
    data type        : signed word (32 bits)


  parameter #1       :
    packing          : XL packing and IEEE reals
    alignment        : half word aligned
    type             : parameter, passed by value
    structure        : simple variable
    data type        : signed half word (16 bits)


  parameter #2       :
    packing          : XL packing and IEEE reals
    alignment        : word aligned
    type             : parameter, passed by value
    structure        : simple variable
    data type        : signed word (32 bits)


  parameter #3       :
    packing          : XL packing and IEEE reals
    alignment        : byte aligned
    type             : parameter, passed by reference
    structure        : array
    data type        : hashed (500b) = pac16 / sp_fm_t_name ?

  parameter #4       :
    packing          : XL packing and IEEE reals
    alignment        : byte aligned
    type             : parameter, passed by reference
    structure        : array
    data type        : hashed (500b) = pac16 / sp_fm_t_name ?

  parameter #5       :
    packing          : XL packing and IEEE reals
    alignment        : byte aligned
    type             : parameter, passed by reference
    structure        : array
    data type        : hashed (500b) = pac16 / sp_fm_t_name ?

  parameter #6       :
    packing          : XL packing and IEEE reals
    alignment        : word aligned
    type             : parameter, passed by reference
    structure        : array
    data type        : hashed (583f)

AVATAR[fscheck.mpexl.telesup]:
```

Figure 16.26 - LOOK Command (direcfind)

Because the parameter to the LOOK command was entered in lower case type, we know immediately that it is not an intrinsic call but rather an external procedure. There are six parameters for this procedure and we can see that the first 3 parameters are simple variables while the last 3 are array (or record) parameters.

Figure 16.27 shows how the EXTRACT command is used.

```
AVATAR[fscheck.mpexl.telesup]: dc _start

_start                              2 2 CODE       UNIVERSAL 261f8    29596

        ; ***************************************************************************

[C]   29594:      6bc23fd9 STW       2,-20(0,30)       ; $ffffffec, sp-20
[C]   29598:      37de0080 LDO       64(30),30         ; $40,       sp+64
[C]   2959c:      6bc03ff9 STW       0,-4(0,30)        ; $fffffffc, sp-4
[C]   295a0:      23f42000 LDIL      $29000,31
[C]   295a4:      e7e02928 BLE       1172(4,31)        ;->?P_INIT_ARGS
[C]   295a8:      081f0242 COPY      31,2

[C]   295ac:      23f42000 LDIL      $29000,31
[C]   295b0:      e7e02968 BLE       1204(4,31)        ;->?U_INIT_TRAPS
[C]   295b4:      081f0242 COPY      31,2

[C]   295b8:      0800025a COPY      0,26
[C]   295bc:      23f42000 LDIL      $29000,31
[C]   295c0:      e7e029a8 BLE       1236(4,31)        ;->?P_INIT_HEAP
[C]   295c4:      081f0242 COPY      31,2

[C]   295c8:      34190002 LDI       1,25
[C]   295cc:      0819025a COPY      25,26
[C]   295d0:      23f42000 LDIL      $29000,31
AVATAR[fscheck.mpexl.telesup]: =$2d5a4-_start
value = 16398, $400e

AVATAR[fscheck.mpexl.telesup]: extract scode _start _start+$9e
extraction complete, 58 lines.

AVATAR[fscheck.mpexl.telesup]: :print scode
        .space $TEXT$,sort=2048
        .subspa $X28E$SM_MS$,quad=0,align=4,access=40,code_only
_start
        .export _start,CODE

         ; ***************************************************************************

        STW       2,-20(0,30)       ; $ffffffec, sp-20
        LDO       64(30),30         ; $40,       sp+64
        STW       0,-4(0,30)        ; $fffffffc, sp-4
        LDIL      L%0x29000,31
        BLE       1172(4,31)        ;->P_INIT_ARGS
        COPY      31,2

        LDIL      L%0x29000,31
        BLE       1204(4,31)        ;->U_INIT_TRAPS
        COPY      31,2

        COPY      0,26
        LDIL      L%0x29000,31
        BLE       1236(4,31)        ;->P_INIT_HEAP
        COPY      31,2

        LDI       1,25
        COPY      25,26
        LDIL      L%0x29000,31
        BLE       1268(4,31)        ;->P_SET_COMPACTION
        COPY      31,2

        LDIL      L%0x29000,31
        BLE       1300(4,31)        ;->P_GET_PARM
        COPY      31,2
```

```
        STW       28,8012(0,27)     ; $1f4c,      dp+8012
        LDO       6472(27),26       ; $1948,      dp+6472
        LDI       1,24
        LDI       1534,25
        LDIL      L%0x29000,31
        BLE       1332(4,31)        ;->P_GET_INFO
        COPY      31,2

        LDIL      L%0x28000,31
        BLE       1884(4,31)        ;->process_events
        COPY      31,2

        LDIL      L%0x29000,31
        BLE       1364(4,31)        ;->P_TERMINATE
        COPY      31,2

        LDIL      L%0x29000,31
        BLE       1396(4,31)        ;->U_EXIT
        COPY      31,2

        LDW       -84(0,30),2       ; $fffffffac, sp-84
        BV        0(2)
        LDO       -64(30),30        ; $fffffffc0, sp-64
         ;    ------------------------------------------------------------

        BREAK     0,0
ok
AVATAR{fscheck.mpexl.telesup}: exit
;
```

Figure 16.27 - EXTRACT Command

AVATAR Error Messages

Message	***Expected "D" or "C" as data type***
Cause	DISPLAY command expects known data type modified.
Action	Use only the "D" (hex/ASCII dump) or "C" (disassemble) data type modifiers.
Message	***Invalid primary***
Cause	User entered unknown AVATAR command or option.
Action	Review command syntax.
Message	***Invalid start address***
Cause	EXTRACT command start address must be within procedure.
Action	Use AVATAR to review the address range for the procedure being extracted.
Message	***Invalid type***
Cause	User entered invalid symbol type for lookup symbol command.
Action	Review symbol types. The valid types are: ABSOLUTE, DATA, CODE, PRI_PROG, SEC_PROG, ENTRY, STORAGE, STUB, MODULE, SYM_EXT, ARG_EXT, MILLICODE, PLABEL, OCT_DIS, MILLI_EXT
Message	***Start must be less than end***
Cause	EXTRACT command expects end offset to be larger than beginning offset.
Action	Review EXTRACT command syntax.

Chapter 17

The CAPTURE Tool

CAPTURE provides a simple method for capturing all or part of your terminal or PC screen buffer. Output may be sent to your line printer or to a disk file. CAPTURE's many options and compatibility with various file formats allow it to be used on non-HP terminals and block-mode applications. CAPTURE can run as a stand-alone program or it can be referenced from an XL and called within your program.

Operation

To use CAPTURE as a stand-alone program, simply type: CAPTURE at the system prompt. Doing this will cause CAPTURE to copy your entire screen memory to the line printer. Like all ***LPS-Tools***, CAPTURE displays the standard ***LPS-Tools*** banner. In this case, however, the banner display follows the screen dump. See the following screen display for an example of how this works.

```
:listf @.pub.lpstools

FILENAME

ACAP          AVATAR        BETIMES       BLAZE         CAPTURE       CSEQ
ETC           FASTLIB       GRANT         KLONDIKE      KNOCKOUT      LP
LPSCFG        LPSCRYPT      LZWPAGES      MAGNET        MODA          PAGES
REDWOOD       REP           SERIAL        SHOT          SPOOK         SPOOK0
TINDEX        UDC           UDC0          VIVITAR

:capture
*******CAPTUREd on  FRI, DEC 15, 1995, 12:01 PM
Printed 12 lines.
CAPTURE [2.0] - LPS Toolbox [A.01a]          (c) 1995 Lund Performance Solutions

For Help, :RUN CAPTURE.PUB.LPSTOOLS,HELP
:
```

Figure 17.1 - CAPTURE Screen

Capabilities

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

Usage

CAPTURE optionlist

optionlist = [option, < option >]
option = COMPRESS | CUT firstcol/lastcol | FF | FFL | FLAT | HELP | Landscape | LEFT column | NOCHECK | NOENHANCE | NOENHOFFEOL | NORESETL | NOSETMSG | NOSTRIP | NOSUMMARY | OFFSET | PARTIAL | QUIET | RIGHT column

Note: There are many keywords (see below) available for use with CAPTURE using the "INFO=keyword" option. You may supply the full keyword, or only the portion indicated with capital letters. More than one keyword may be used; the space character is the delimiter.

Option Summary

Unlike other ***LPS-Tools***, CAPTURE is a single-command tool, where the CAPTURE command is the only command that can be specified. Several options, however, can be used to further define the task at hand. These options are listed next.

Option	Description
COMPRESS	Compressed portrait output to LaserJet
CUT firstcol/lastcol	Specify columns to capture
FF	Formfeed line printer
FFL	Formfeed LaserJet
FLAT	Directs output to a disk file
HELP	Starts Capture's help subsystem
Landscape	Landscape output to LaserJet
LEFT column	Specify left column to capture
[NO]CHECK	Limits error checking activity
[NO]ENHance	Strips display enhancements
[NO]ENHOFfeol	Adds printer escape sequence at end of line
[NO]RESETL	Reset LaserJet
[NO]SETMSG	Controls SETMSG use during capture
[NO]STRIP	Checks for stripped enhancements before outputting
[NO]SUMmary	Suppresses line & timestamp on the screen
OFFSET	Specify offset in output column
PARTial	Captures a user-specified range of lines
QUIET	Suppresses error messages
RIGHT column	Specify right column to capture

CAPTURE Commands

CAPTURE (parm=0)
(No parameters)

Running CAPTURE without any options causes your entire screen memory to be captured.

CAPTURE starts copying from the top of terminal memory (line number 1) and copies through the line that contains the cursor's original location. CAPTURE does not alter terminal memory. The output from CAPTURE will be sent to the line printer attached to your system. The formal file equation that CAPTURE (as well as all other ***LPS-Tools***) uses is LPSLP. To redirect CAPTURE's output to a line printer other than LP, simply issue an appropriate file equation.

Options Definitions

CAPTURE options may be specified as keywords in the INFO string or as bits in the PARM value. The following list shows the keywords and PARM value for each option.

COMPRESS (parm = 4096)

Format for LaserJet compressed output. This option can be used in Landscape mode.

CUT firstcol/lastcol (no parm)

This option is used to specify the column range (where "**n**" equals the column number) used in the capture.

FF (parm = 256)

This option instructs capture to send a formfeed command at the end of the screen capture. It is used for line printer output.

FFL (parm = 512)

This option is the same as FF except that it is used for LaserJet output.

FLAT (parm = 2)

This option is used to tell CAPTURE that you wish to redirect output to a disk file. The formal file designator for the disk file is also called FLAT. You may redirect output to a file of another name by using an appropriate file equation.

Note: The file is built with a default record length of 80 bytes. This filename must not exist prior to running CAPTURE. If a system problem prevents the saving of the file as permanent, an attempt will be made to save it as session temporary. When FLAT is used, only the lines of text in terminal memory are copied to the disk file. There will be no additional information lines appended by CAPTURE for audit purposes, such as information summaries or date and time stamps. CAPTURE will not purge an existing file named FLAT.

HELP (parm = 32768)

Starts Capture's help subsystem.

Landscape (parm = 2048)

Format for LaserJet landscape output.

LEFT column (parm = 2)

This option is used to specify the starting (left) column for the capture.

[NO]CHECK (parm = 4)

This option inhibits CAPTURE from checking certain error conditions. This can be useful if you have non-HP terminals that are similar to standard, but that would be ignored by CAPTURE if it detected the error conditions. Using NOCHECK also causes CAPTURE to ignore errors that might be generated when running in BATCH mode.

Be sure that you understand what you are doing when you use this option. Ports and jobs could be hung if this option is used improperly.

[NO]ENHance (parm = 16)

This option strips display enhancements before sending the file as output.

[NO]ENHOFFEOL (parm = 32)

ENHOFFEOL causes CAPTURE to add "<esc>&d@" at the end of any line that contains display enhancements. This is useful for LaserJet printing.

[NO]RESETL (parm = 8192)

Resets the LaserJet with an <esc> "E" before sending the capture.

[NO]SETMSG (parm = 128)

This option controls whether or not capture uses SETMSG ON (default) or SETMSG OFF.

[NO]STRIP

This option determines whether or not enhancements are stripped from the output before sending it to the output device.

[NO]SUMMARY (parm = 8)

This option suppresses the CAPTURE line and timestamp on the display screen. This summary information is displayed at the start of each screen capture process.

OFFSET

This option is used to specify a column offset for the capture output.

PARTIAL (parm = 1)

Using this option tells CAPTURE to capture only a portion of the lines in terminal memory. CAPTURE will interactively request that you mark the last line and then the first line of text that you want to CAPTURE. Use of this option will suppress the information summary (see SUMMARY) from the display.

QUIET (parm = 16384)

Suppresses Capture's error messages.

RIGHT column

This option is used to specify the ending (right) column for the capture.

CAPTURE Examples

Use CAPTURE to select portions of the terminal screen, send the contents to a flat file, or combine various options. Combine all three (CAPTURE, PARTIAL, and FLAT) of CAPTURE's options to realize a partial screen capture to a flat file. See the screen example below. Do this with the command CAPTURE PARTIAL, or PARTIAL FLAT.

```
:listf @.pub.lpstools

FILENAME

ACAP          AVATAR        BETIMES       BLAZE         CAPTURE       CSEQ
ETC           FASTLIB       GRANT         KLONDIKE      KNOCKOUT      LP
LPSCFG        LPSCRYPT      LZWPAGES      MAGNET        MODA          PAGES
REDWOOD       REP           SERIAL        SHOT          SPOOK         SPOOK0
TINDEX        UDC           UDC0          VIVITAR

:capture partial flat

**Move cursor onto the LAST line you want printed and hit return...
**Move cursor onto the FIRST line you want printed and hit return...

*******CAPTUREd on  FRI, DEC 15, 1995, 12:08 PM
CAPTURE [2.0] - LPS Toolbox [A.01a]        (c) 1995 Lund Performance Solutions

For Help, :RUN CAPTURE.PUB.LPSTOOLS,HELP
:
```

Figure 17.2 - Capturing a Portion of Screen Memory

The following example demonstrates how to set a file equation so that CAPTURE's output goes to the filename of your choice.

```
:report @.lpstools
ACCOUNT        FILESPACE-SECTORS      CPU-SECONDS     CONNECT-MINUTES
   /GROUP        COUNT     LIMIT     COUNT     LIMIT     COUNT     LIMIT
LPSTOOLS        120480        **        44        **       205        **
   /CFG             32        **         0        **         0        **
   /CM            3424        **         0        **         0        **
   /CMD              0        **         0        **         0        **
   /DATA          8080        **         0        **         0        **
   /EXTERNAL       112        **         0        **         0        **
   /HELP         33920        **         0        **         0        **
   /JOB            528        **         0        **         0        **
   /O              400        **         0        **         0        **
   /PUB          70032        **        44        **       205        **
   /PUBSYS         352        **         0        **         0        **
   /RL             496        **         0        **         0        **
   /SOURCE          32        **         0        **         0        **
   /USL           2016        **         0        **         0        **
   /XL            1056        **         0        **         0        **
:file flat=lpstrept
:capture flat
*******CAPTUREd on  FRI, DEC 15, 1995, 12:10 PM
Captured 21 lines. (Flat, NoAuthor)
CAPTURE [2.0] - LPS Toolbox [A.01a]        (c) 1995 Lund Performance Solutions

For Help, :RUN CAPTURE.PUB.LPSTOOLS,HELP
:
```

Figure 17.3 - Sending CAPTURE Output to a File

Figure 17.4 shows the Capture of columns 10 through 30 and starting output in the capture file at column 8.

```
:capture cut 10/40, offset=8
*******CAPTUREd on  FRI, DEC 15, 1995, 12:12 PM
Printed 21 lines. (NoEnhance, EnhOffEol, Cut 10/40, Off 8)
CAPTURE [2.0] - LPS Toolbox [A.01a]        (c) 1995 Lund Performance Solutions

For Help, :RUN CAPTURE.PUB.LPSTOOLS,HELP
:
```

Figure 17.4 - Column CAPTURE

Using CAPTURE as a Callable Procedure

CAPTURE may be used as a callable procedure from both CM and NM programs. To include CAPTURE in a CM program, refer to the USL file in the USL group of the LPSTOOLS account. For inclusion in an NM program, take a look at the NMOBJ file in the O group. The following display is from the file CAPTURE.PASCAL.LPSTOOLS. Use it as a reference in implementing CAPTURE in your program for sending screen memory to the line printer. Also provided are examples in C, COBOL, and SPL, as well as RL and XL versions of the executables.

```
$standard_level  'os_features'$

program capture_demo (output, info, parm);

type
  pac80              = packed array [1 . . 80] of char;

var
  buf                : pac80;
  err                : shortint;
  i                  : integer;
  info               : string [80];
  leftcol            : shortint;
  len                : shortint;
  offsetcols         : shortint;
  options            : shortint;
  parm               : shortint;
  rightcol           : shortint;

$include 'capturep.external.lpstools'$

begin

for i  := 1 to strlen (info) do
      buf  [i]   :=  info  [i] ;

options  :=  parm;
if parse_capture_options  (buf, strlen  (info) , options,
                           leftcol, rightcol, offsetcols)
             = 0 then                    {0 = bad, 1 = good}
      writeln ('Error # ', err:1,
               ' in parsing CAPTURE options, ignored.');

err := capture (0, options, 0, 0);

if  err < > 0 then
      writeln  ('CAPTURE  error #', err:1);

end.
```

Figure 17.5 - CAPTURE as a Callable Procedure

Using CAPTURE Procedures in COBOL

The following code fragment illustrates the use of CAPTURE in COBOL code where whatever is on the screen is sent to the line printer.

```
$control uslinit

IDENTIFICATION DIVISION.
program-id. foo.
author.stan sieler.

ENVIRONMENT DIVISION.
configuration section.
source-computer. HP3000.
object-computer. HP3000.
special-names.
       condition-code is cond-code.

DATA DIVISION.
working-storage section.
01     buf           pic  x(255).
01     err           pic  s9(04)   usage is computational.
01     leftcol       pic  s9(04)   usage is computational.
01     len           pic  s9(04)   usage is computational.
01     offsetcols    pic  s9(04)   usage is computational.
01     options       pic  s9(04)   usage is computational.
01     rightcol      pic  s9(04)   usage is computational.

PROCEDURE DIVISION.
enter-routine.
       move "ENHOFFEOL" to buf.
       move 9 to len.
       move 0 to options.
       call "PARSE_CAPTURE_OPTIONS" using buf, \len\, options,
                                          leftcol, rightcol, offsetcols
              giving err.
       call "CAPTURE" using \0\, \options\, \0\, \0\,
                            \leftcol\, \rightcol\, \offsetcols\

              giving err.
       if (err not = 0)
              display "CAPTURE error #", err.
       stop run.
```

Figure 17.6 - CAPTURE Procedures in COBOL

Using CAPTURE Procedures in SPLash!

The following code fragment illustrates the use of CAPTURE in SPLash!, the native mode SPL compiler. Whatever is on the screen is sent to the line printer.

```
integer Procedure capture (quiet, options, printer, recchars,
                           left'col, right'col, offset'cols);
      value  quiet, options, printer, recchars,
                           left'col, right'col, offset'cols;
      logical quiet, options;
      integer                      left'col, right'col, offset'cols;
      integer                            printer, recchars;
option variable, intrinsic, native, nocc,      ! no PARM mask!!
            external;



logical procedure Parse'capture'options  (itemp, ileft, options,
                                  left'col, right'col, offset'cols);
      value  itemp, ileft;
      integer left'col, right'col, offset'cols;
      integer ileft;
      logical options;
      byte pointer itemp;
  option external, variable, intrinsic, native, nocc;   ! no mask!

...
logical      err, options, quiet;
integer      left'col, right'col, offset'cols, printer, rec'chars;
byte array buf' (0:79);
...
err  :=  parse'capture'options(buf',move buf' :="COMPRESSED",options
                          left'col, right'col, offset'cols);
capture (quiet, options, printer, rec'chars,
               left'col, right'col, offset'cols);
```

Figure 17.7 - CAPTURE in SPLash

CAPTURE Error Messages

Message	***Could not open FLAT file.***
Cause	Possible bad file-equation for FLAT.
Action	Check file equation for FLAT with the HP command LISTEQ.
Message	***Could not open LPSLP***
Cause	Possible bad file-equation for LPSLP.
Action	Check file equation for LPSLP with the HP command LISTEQ.
Message	***Error in attempt to turn off echo.***
Cause	As part of the screen capture process, CAPTURE needs to disable echo on the terminal - temporarily.
Action	Try running CAPTURE again with PARM=4.
Message	***Error writing to LPSLP.***
Cause	Possible bad file-equation for LPSLP.
Action	Check file equation for LPSLP with the HP command LISTEQ.
Message	***I/O error in reading terminal status***
Cause	During an "**fcontrol(,4,3)**" (a three second read) CAPTURE received an error status.
Action	Try running CAPTURE again with PARM=4.
Message	***I/O error on read from screen***
Cause	CAPTURE failed to read a line of text from the screen via an "ESC d" command.
Action	Terminal may not be compatible with HP26xx commands. Try again with PARM=4.
Message	***Not a 26xx terminal***
Cause	CAPTURE determined that the terminal was not compatible with the HP 26xx command set.
Action	Depending on the terminal, running with PARM=4 may allow correct CAPTURE operation.
Message	***Too many empty lines found (more than 99)***
Cause	CAPTURE remembers how many empty (consecutive) lines it has read. Currently the maximum allowed is 99. CAPTURE does this so that a "runaway" screen capture will not send (possibly thousands) unwanted empty lines to LPSLP.
Action	Make sure that the CAPTURE range (for a partial capture) is valid.
Message	***Too many lines found (more than 9000)***
Cause	CAPTURE can only screen capture 9000 lines at a time.
Action	Use CAPTURE's partial option to break the screen capture into smaller pieces.
Message	***Unknown CAPTURE option:***
Cause	An invalid option was input.
Action	Make sure that the option used is spelled correctly. It may have captured the screen contents anyway and disregarded the invalid option. Try CAPTURE again, using the correct option.

Chapter 18

The CHRONOS Tool

CHRONOS is a library of procedures that manipulate date and time information in a variety of formats. Information can easily be converted from one form to another, including forms that permit arithmetic calculations. It is also possible to increment or decrement time or date values.

CHRONOS supports a date range from year 0 to 4095, offering an immediate solution to "turn of the century" problems.

Operation

CHRONOS is most typically used to translate a date or time from a "stored" format in a data base (i.e., 960331) to a "display" format on a screen or report (i.e., March 31, 1996), or to reformat a date from a data entry field (i.e., 033196) to a "stored" format (i.e., 960331), or to calculate the amount of time between two events.

CHRONOS can be called like an MPE intrinsic. This means that the user intrinsic file, CHRONOS.INTRIN.LPSTOOLS (in SYSINTR format), should be specified in your source along with the CHRONOS intrinsic declaration. Parameter specifications are used to further define the operation. Therefore calling CHRONOS boils down to determining what kind of operation you want performed and passing the correct parameters to CHRONOS. The kind of operation you want performed is specified in the CHRONOS **mode** parameter. There are literally hundreds of possible configurations that you can specify. Appendix H, "*CHRONOS Modes*," provides a comprehensive listing of all modes.

Because CHRONOS provides so many conversions, not all parameters may be required for each call. Parameter omission is language dependent, and you should consult your language documentation for details. *HP C/iX*, *HP Pascal/iX*, and *SPLash!* all use the comma to omit parameters. For ANSI-C compatibility, use the keyword "NULL" to omit parameters.

The examples provided show you how to handle parameter specification. The syntax examples show you the ordering sequence of the parameters and the data type for each parameter.

Date and Time Formats

CHRONOS supports several date and time formats:

chronos-stamp

CHRONOS supports an internal format called **chronos-stamp**. The **chronos-stamp** is a 6-byte time field with millisecond precision. For example, the 6-byte **chronos-stamp** for January 28, 1993 at precisely 4:38:00.1269 p.m. is (in hexadecimal) $7C90E42704F5 (see the section entitled, "*CHRONOS Time and Date Stamp*," for a bit-level description).

Gregorian (formatted)

The formatted Gregorian date and time uses 8 bytes of storage. The field separator for the date defaults to the slash (/). The field separator for the time defaults to the colon (:). The standard US formatting for the date and time for the last day of 1995 at noon would look like 12/31/95 12:00:00. You may choose any symbol as a field separator when a call is made to CHRONOS.

The date can be returned in one of three ways:

year-month-day (e.g., 95/12/31)
day-month-year (e.g., 31/12/95)
month-day-year (e.g., 12/31/95)

Gregorian (unformatted)

The unformatted Gregorian date and time uses 6 bytes of storage. The standard US-style for the unformatted date and time for the last day of 1995 at noon would look like 123195 120000.

The date can be returned in one of three ways:

year-month-day (e.g., 951231)
day-month-year (e.g., 311295)
month-day-year (e.g., 123195)

Julian

The Julian year is returned in a 2-byte array and is not terminated with any special character. Leading zero digits are padded with ASCII "0" not ASCII spaces. For example, for 1995 the Julian year would return 95. The Julian day of the year is returned in a 3-byte array and is not terminated with any special character. Leading zero digits are padded with ASCII "0" not ASCII spaces. For example, the Julian day for Feb 1 would return 032.

String

CHRONOS provides four ways to format string output:

day-month-year
month-day-year
dayname-day-month-year
dayname-month-day-year

The length of the string output is always 30 and is not terminated with any special character. Unused characters are set to ASCII spaces:

day-month-year	looks like "28 January 1995	"
month-day-year	looks like "January 28, 1995	"
dayname-day-month-year	looks like "Thursday, 28 January 1995	"
dayname-month-day-year	looks like "Thursday, January 28, 1995	"

CHRONOS can convert any of the above formats into any of the other formats. In addition, by specifying an increment, CHRONOS can increment the time or date either forward or backward.

Providing the optional parameter "**day_of_week**" will cause CHRONOS to return the numerical day of the week, where Sunday=0, Monday=1, and so forth.

Providing the optional parameter "**century**" allows the user to change the default century, or to obtain the current century. For example, this parameter returns 1900 currently.

CHRONOS Intrinsic

CHRONOS Intrinsic performs the requested date/time conversion:

int chronos (parameter1, parameter2 [,parameter3, ... parameter15])

The Parameter Set is listed next where each parameter is either an integer, character array, or byte:

Parameter	Name	Type	Comment
1	status	integer — 32-bit signed	Required
2	mode	integer — 32-bit signed	Required
3	chronos_stamp	character array	Optional
4	formatted_date	character array	Optional
5	formatted_time	character array	Optional
6	unformatted_date	character array	Optional
7	unformatted_time	character array	Optional
8	date_symbol	byte	Optional
9	time_symbol	byte	Optional
10	increment	integer — 32-bit signed	Optional
11	chronos_string	character array	Optional
12	julian_year	character array	Optional
13	julian_date	character array	Optional
14	day_of_week	integer — 32-bit signed	Optional
15	century	integer — 32-bit signed	Optional

Return Value

CHRONOS returns a 32-bit integer encoded as follows:

< 0 :*Error*

-1 = bad parameter, check the status variable for more information.
-23= conversion error, check the status variable for more information.

= 0 :*No error*

> 0 :*Warning* conversion probably worked, check status variable for more information.

Parameters

status Integer by reference (required). Contains the status of the call to CHRONOS. The sign of the return value describes the kind of status where a negative value denotes an error and a positive value denotes a warning. The absolute value of status is the number of the error or warning. A zero value means the call was successful.

mode Integer by value (required). Contains the bit-encoded directions for the conversion. See the section entitled "*CHRONOS Mode*" for complete information.

chronos_stamp Byte array by reference (optional). Contains the 6-byte CHRONOS time and date stamp. See the section entitled "*CHRONOS Date and Time Stamp*" for complete information.

formatted_date Byte array by reference (optional). Contains the 8-byte string that represents the month, day and year in various formats. For example, 03/12/92. The number separator defaults to the slash (/). Use the **date_symbol** parameter to specify an alternate separator symbol.

formatted_time Byte array by reference (optional). Contains the 8-byte string that represents the hour, month and second formatted as hh:mm:ss where hh is in 24 hour format. The number separator defaults to the colon (:). Use the **time_symbol** parameter to specify an alternate symbol.

unformatted_date Byte array by reference (optional). Contains the same information as **formatted_date**, except that the number separator has been omitted. The length of the array is 6 bytes.

unformatted_time Byte array by reference (optional). Contains the same information as **formatted_time**, except that the number separator has been omitted. The length of the array is 6 bytes.

date_symbol Byte by value (optional). Contains the single ASCII character that will be used to separate the numbers in the **formatted_date** string.

time_symbol Byte by value (optional). Contains the single ASCII character that will be used to separate the numbers in the **formatted_time** string.

increment Integer by value (optional). Contains the signed value that can be used to add or subtract values from the time or date as specified by the mode parameter.

chronos_string Byte array by reference (optional). The 30-byte array that contains the formatted date string in one of several formats as specified by the mode parameter. For example: Monday, January 14, 1995.

julian_year Byte array by reference (optional). The 2-byte array that contains the last two digits of the year. For example: "95" for the year 1995.

julian_date Byte array by reference (optional). The 3-byte array that contains the Julian date of the current year. For example: "312" for the 312th day of the year.

day_of_week Integer by reference (optional). If provided, this parameter returns the numerical day of the week. The number returned is in the range 0..6 where 0=Sunday, 1=Monday, and so forth.

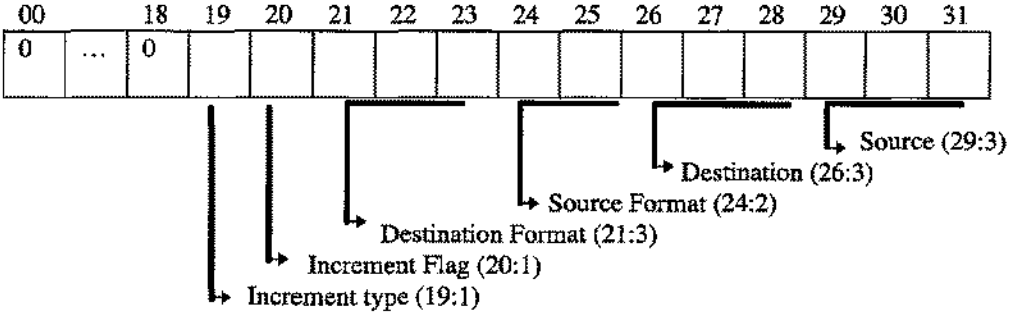
century Integer by reference (optional). Can be used to specify the century, or will return the current century if passed in with a value of zero (0).

Operation

This section provides how-to information for two key topics. First, information on how to specify the CHRONOS **mode** parameter is discussed. This section is followed by chronos-stamp specifications.

CHRONOS_MODE

The CHRONOS **mode** parameter is used to specify the type of operation you want performed. The CHRONOS **mode** is a 32-bit integer where bits 0 to 18 should be zero and bits 19 through 31 are encoded as follows:



Each of the encoded bit fields (source, destination, source format, etc.) is discussed next.

Source(29:3) and Destination(26:3) Bit Mapping

000	System Local Time and Date from the CALENDAR intrinsic (Source only)	
001	CHRONOS time and date stamp	Required parameter: chronos_stamp
010	Formatted string	Required parameter: formatted_date Optional parameter: formatted_time
011	Unformatted string	Required parameter: unformatted_date Optional parameter: formatted_time
100	Julian date and year	Required parameter: julian_year, julian_date
101	String	Required parameter: chronos_string (Destination only)

Source format (24:2) Bit Mapping

00	MDY	(month, day, year)
01	DMY	(day, month, year)
10	YMD	(year, month, day)

Note: Only meaningful for formatted string and unformatted string modes.

Destination format (21:3) Bit Mapping

000	MDY	(month, day, year)
001	DMY	(day, month, year)
010	YMD	(year, month, day)

Note: Only meaningful for formatted string and unformatted string modes. For example, if the Destination field is "**101 (STRING)**," then the Destination format is bit mapped as follows:

000	dayname, monthname, day, year	(e.g. MON, JANUARY 21, 1995)
001	dayname, day, monthname, year	(e.g. MON, 21 JANUARY 1995)
010	monthname, day, year	(e.g. JANUARY 21, 1995)
011	day, monthname, year	(e.g. 21 JANUARY, 1995)

Increment Flag (20:1)

This bit is a flag that is used to determine if a time or date field should be incremented.

0	no increment
1	increment wanted
	Check bit (19:1) to determine if source time or date increment is desired.

Increment Type (19:1)

This bit field is used in conjunction with bit field (20:1) and the increment parameter to specify an increment value and type. If the value for this bit field is zero, then the increment parameter contains the number of days to be added or subtracted to the source date. If the value for this bit field is one, then the increment parameter contains the number of minutes to be added or subtracted to the source time.

0	source date increment (in days)
1	source time increment (in minutes)

Note: Some combinations of mode values and parameters can result in superfluous information being passed to CHRONOS. If CHRONOS can detect such a case, a warning will be returned. See Appendix H entitled, "*CHRONOS Modes*," for a complete list of all mode numbers. Because there are some "don't care" cases, there are several mode numbers that produce the same results.

CHRONOS_STAMP

CHRONOS has a unique format for storing the precise "definition" of a moment in time, including century through millisecond and all components in between. This is accomplished by using a "bit-mapping" technique in a 6-byte field:

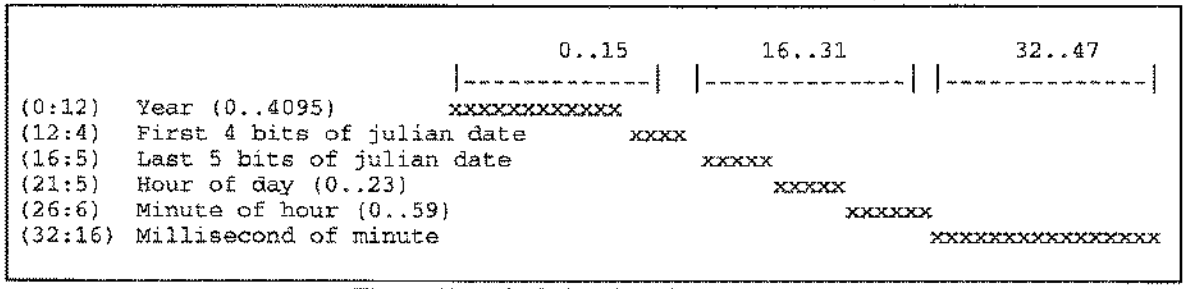


Figure 18.1 - Defining CHRONOS_STAMP

When **chronos_stamps** are being stored as data, it may be desirable to zero out all or portions of the time maps. For instance, if the **chronos_stamp** is being used as a Key into a data base record based on date, the time portion would cause multiple entries for the same date to be created.

If Keys are being set up based on the date and time of an entry, for instance in an auditing situation for tracking when data was placed in the data base, the milliseconds might cause multiple entries for the same minute.

CHRONOS Examples

Figure 18.2, *Figure 18.3* and *Figure 18.4* in this section were compiled with HP's C/iX compiler using the following command statements:

Compile statement:

ccxl exam1,,$NULL;info="-Aa -Wc, -e"

Link statements (i.e., linking to the RL's **chronos.rl.lpstools** and **libcinit.lib.sys**)

:link from=$OLDPASS;TO=exam1.pub;rl=chronos.rl.lpstools,libcinit.lib.sys

Figure 18.2 shows how CHRONOS will use the system-local date and return the chronos-string in dayname-month-day-year format:

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#pragma intrinsic_file "CHRONOS.INTRIN.LPSTOOLS"
#pragma intrinsic chronos
#pragma intrinsic_file ""

void example_1(void);

int main( )
{
    example_1( );
    return;
}
int example_1( )
{
int status;
int mode;
int result;
char chronos_str[30];

    mode = 0x0028;
    result = chronos(&status,mode,,,,,,,,chronos_str);
    if (result)  /* error */
        /* check status */ ;
    else
        printf("%.30s\n",chronos_str);
}
```

Figure 18.2 - System-Local Date

Figure 18.3 is an example of how to call CHRONOS twice, the first time to get the current date and time and return it as formatted date and time. Then, call CHRONOS again to subtract 2 hours from the formatted time.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#pragma intrinsic_file "CHRONOS.INTRIN.LPSTOOLS"
#pragma intrinsic choronos
#pragma intrinsic_file ""

void example_2(void);

int main( )
{
    example_2( );
    return;
}
int example_2( )
{
int status;
int mode;
int result;
int increment;
char fdate[8],ftime[8];

    mode = 0x0010;
    result = chronos(&status,mode,,fdate,ftime);
    if (result)  /* error */
        /* check status */;
    else {
        mode = 0x1812;
        increment = -120;  /* Subtract 120 minutes (2 hours) */
        result = chronos(&status,mode,,fdate,ftime,,,,,increment);
        if (result)  /* error */
            /* check status */ ;
        else{
            printf("\n[%.8s]",fdate);
            printf("\n[%.8s]",ftime);
            }
        }
    }
```

Figure 18.3 - Calling CHRONOS Twice

Figure 18.4 is an example of rewriting the previous example to perform the same function with only one call to CHRONOS:

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#pragma intrinsic_file "CHRONOS.INTRIN.LPSTOOLS"
#pragma intrinsic chronos
#pragma intrinsic_file ""

void example_3(void);

int main( )
{
   example_3( );
   return;
}

int example_3( )
{
int status;
int mode;
int result;
int increment;
char fdate[8],ftime[8];

   mode = 0x1810;
   increment = -120;
   result = chronos(&status,mode,,fdate,ftime,,,,,increment);
   if (result)  /* error */
      /* check status */;
   else{
      printf("\n[%.8s]",fdate);
      printf("\n[%.8s]",ftime);
      }
}
```

Figure 18.4 - Calling CHRONOS Once

The following figure shows how to call CHRONOS in Pascal (see the **testchro.pascal** file):

```
$sysintr 'CHRONOS.INTRIN.LPSTOOLS'$
program example_4(output);

type
   chronos_string_type = packed array [1..30] of char;

var
   chronos_string : chronos_string_type;
   status  : integer;
   mode    : integer;
   result  : integer;

function chronos:integer; intrinsic;

begin

   status := 0;
   chronos_string :='                                    ';
   mode := hex('0028');
   result := chronos(status, mode,
                     ,,,,,,,,
                     chronos_string);
   if (result = 0)
     then
       writln('[',chronos_string:30,']')
     else
       writeln('mode=',mode:4,
               'result=',result:4,' status=',status:4);

end.
```

Figure 18.5 - Pascal Sample Calling CHRONOS

Figure 18.6 shows how to call CHRONOS in SPLash! (See the **testchro.spl** file):

```
$native  << SPLash! >>
begin

logical array msg(0:39);
byte array m(*)=msg;
integer i;

byte array chronos'string(0:29);
double    result;
double    status;
double    mode;

intrinsic print,ascii,dascii;
intrinsic (chronos.intrin) chronos;

mode := $0028d;
status := 0d;
result := chronos(status,mode,,,,,,,,
                  chronos'string);

if (result=0d) then
  print (chronos'string,-30,0)
else
  begin
    i := move m := "mode=";
    i := i + dascii(mode,10,m(i));
    i := i + move m(i) := " result=";
    i := i + ascii(integer(result),10,m(i));
    i := i + move m(i) := " status=";
    i := i + ascii(integer(status),10,m(i));
    print (msg,-i,0);
  end;

end.
```

Figure 18.6 - SPLash! Sample Calling CHRONOS

Figure 18.7 shows how to call CHRONOS in COBOL (see the **testchro.cobol** file:

```
IDENTIFICATION DIVISION.
PROGRAM-ID. COBTEST.
AUTHOR. SRN.
*-------------------------------------------------------------------------*
*                          Compilation/Run Instructions                   *
*-------------------------------------------------------------------------*
*             FILE SYSINTR.PUB.SYS=CHRONOS.INTRIN.LPSTOOLS                *
*             COB85XL COBTEST,,$NULL                                      *
*             PURGE COBTESTP                                              *
*             LINK FROM=$OLDPASS;TO=COBTESTP;RL=CHRONOS.RL.LPSTOOLS       *
*             RUN COBTESTP                                                *
*-------------------------------------------------------------------------*

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SOURCE-COMPUTER. HP3000.
OBJECT-COMPUTER. HP3000.
SPECIAL-NAMES.
    CONDITION-CODE IS COND-CODE.

DATA DIVISION.
WORKING-STORAGE SECTION.
```

```
*-----------------------------------------------------------------------*
*                          CHRONOS Variables                            *
*-----------------------------------------------------------------------*
01   CHRON-STATUS                          PIC    S9(09)   VALUE 0      COMP.
     88 CHRON-OP-VALID                                     VALUE 0.
     88 CHRON-BAD-PARAM                                    VALUE -1.
     88 CHRON-CONVERSION-ERROR                             VALUE -23.
01   CHRON-MODE                            PIC    S9(09)   VALUE 0      COMP.
01   CHRON-STAMP                           PIC     X(06) .
01   CHRON-DATE-FMT.
     03 CHRON-DATE-FMT-MD.
        09 CHRON-DATE-FMT-M                 PIC     X(02)   VALUE SPACES.
        09 FILLER                           PIC     X(04)   VALUE SPACES.
     03 CHRON-DATE-FMT-Y                    PIC     X(02)   VALUE SPACES.
01   CHRON-TIME-FMT                        PIC     X(08)   VALUE SPACES.
01   CHRON-DATE-UFMT.
     03 CHRON-D1-UFMT                       PIC     9(02)   VALUE 00.
     03 CHRON-D2-UFMT                       PIC     9(02)   VALUE 00.
     03 CHRON-D3-UFMT                       PIC     9(02)   VALUE 00.
01   CHRON-TIME-UFMT.
     03 CHRON-T1-UFMT                       PIC     9(02)   VALUE 00.
     03 CHRON-T2-UFMT                       PIC     9(02)   VALUE 00.
     03 CHRON-T3-UFMT                       PIC     9(02)   VALUE 00.
01   CHRON-DATE-SYMBOL                     PIC     X(01)   VALUE "/".
01   CHRON-TIME-SYMBOL                     PIC     X(01)   VALUE ":".
01   CHRON-INCREMENT                       PIC    S9(09)   VALUE 0      COMP.
01   CHRON-STRING                          PIC     X(30)   VALUE SPACES.
01   CHRON-JUL-YR                          PIC     X(02)   VALUE SPACES.
01   CHRON-JUL-DY                          PIC     X(03)   VALUE SPACES.
01   CHRON-DAY-OF-WEEK                     PIC    S9(09)   VALUE 0      COMP.
01   CHRON-CENTURY                         PIC    S9(09)   VALUE 0      COMP.
01   CHRON-RESULT                          PIC    S9(09)   VALUE 0      COMP.
01   CHRON-DISPLAY                         PIC    ------- 9.
*-----------------------------------------------------------------------*
*                          CHRONOS Modes                                *
*    Examples:                                                          *
*    CHRON-FMTMDY-STAMP              FROM formatted MM/DD/YY            *
*                                      TO chronos timestamp (binary)    *
*    CHRON-STAMP-UFMTMDY             FROM chronos timestamp (binary)    *
*                                      TO unformatted MMDDYY            *
*    CHRON-SYS-FMTMDY                FROM system date                   *
*                                      TO formatted MM/DD/YY            *
*    CHRON-UFMTYMD-UFMTMDY           FROM unformatted YYMMDD            *
*                                      TO unformatted MMDDYY            *
*-----------------------------------------------------------------------*
01   CHRON-FMTMDY-SYS                      PIC    S9(09)   VALUE 2      COMP.
01   CHRON-FMTMDY-STAMP                    PIC    S9(09)   VALUE 10     COMP.
01   CHRON-FMTMDY-UFMTMDY                  PIC    S9(09)   VALUE 26     COMP.
01   CHRON-FMTMDY-UFMTYMD                  PIC    S9(09)   VALUE 538    COMP.
01   CHRON-FMTYMD-STAMP                    PIC    S9(09)   VALUE 138    COMP.
01   CHRON-JUL-UFMTYMD                     PIC    S9(09)   VALUE 540    COMP.
01   CHRON-STAMP-FMTMDY                    PIC    S9(09)   VALUE 17     COMP.
01   CHRON-STAMP-UFMTMDY                   PIC    S9(09)   VALUE 25     COMP.
01   CHRON-STAMP-UFMTYMD                   PIC    S9(09)   VALUE 537    COMP.
01   CHRON-SYS-STAMP                       PIC    S9(09)   VALUE 8      COMP.
01   CHRON-SYS-FMTMDY                      PIC    S9(09)   VALUE 16     COMP.
01   CHRON-SYS-FMTMDY-INC                  PIC    S9(09)   VALUE 2064   COMP.
01   CHRON-SYS-JUL                         PIC    S9(09)   VALUE 32     COMP.
01   CHRON-SYS-STRING                      PIC    S9(09)   VALUE 40     COMP.
01   CHRON-SYS-UFMTMDY                     PIC    S9(09)   VALUE 24     COMP.
01   CHRON-SYS-UFMTYMD                     PIC    S9(09)   VALUE 536    COMP.
01   CHRON-SYS-UFMTYMD-INC                 PIC    S9(09)   VALUE 2584   COMP.
01   CHRON-UFMTMDY-STAMP                   PIC    S9(09)   VALUE 11     COMP.
01   CHRON-UFMTMDY-FMTMDY                  PIC    S9(09)   VALUE 19     COMP.
01   CHRON-UFMTMDY-UFMTMDY                 PIC    S9(09)   VALUE 27     COMP.
01   CHRON-UFMTMDY-UFMTYMD                 PIC    S9(09)   VALUE 539    COMP.
```

```
01    CHRON-UFMTYMD-FMTMDY                    PIC     S9(09)    VALUE 147    COMP.
01    CHRON-UFMTYMD-UFMTMDY                   PIC     S9(09)    VALUE 155    COMP.
01    CHRON-UFMTYMD-STAMP                     PIC     S9(09)    VALUE 139    COMP.
01    CHRON-UFMDY-UFMDY-ID                    PIC     S9(09)    VALUE 2075   COMP.
01    CHRON-UFYMD-UFYMD-ID                    PIC     S9(09)    VALUE 2715   COMP.

*-----------------------------------------------------------------------------*
*                              Prompt Answer Variables                        *
*-----------------------------------------------------------------------------*
01    PA-USER-INPUT                           PIC   X(08) .
01    PA-USER-JUL-DATE.
      03 PA-USER-JUL-YR                       PIC   X(02) .
      03 PA-USER-JUL-DY                       PIC   X(03) .

PROCEDURE DIVISION.
*-----------------------------------------------------------------------------*
*     SUMMARY: This routine will demonstrate how LPS uses CHRONOS.            *
*-----------------------------------------------------------------------------*
P099-MAINLINE.
      PERFORM P400-PROCESS-CURRENT      THRU P400-EXIT.
      PERFORM P401-TEST1-UYMD-FMDY      THRU P401-EXIT.
      PERFORM P402-TEST2-UMDY-UYMD      THRU P402-EXIT.
      PERFORM P403-TEST3-JUL-UYMD       THRU P403-EXIT.
P099-END.
      STOP RUN.
$PAGE
*-----------------------------------------------------------------------------*
*     This routine converts the system date to a string.                      *
*        FROM = SYSTEM-DATE            TO = CHRON-STRING                      *
*-----------------------------------------------------------------------------*
P400-PROCESS-CURRENT.
      MOVE SPACES                         TO PA-USER-INPUT
                                             CHRON-STRING.
      MOVE CHRON-SYS-STRING               TO CHRON-MODE.
      PERFORM P555-CALL-CHRONOS          THRU P555-EXIT.
      DISPLAY "*** Current Date ***",     CHRON-STRING.
P400-EXIT.
      EXIT.
*-----------------------------------------------------------------------------*
*     This routine accepts unformatted date of 'YYMMDD' and                   *
*     converts it to a formatted date of 'MM/DD/YY'. It also                  *
*     converts the time if input.                                             *
*        FROM = CHRON-DATE-UFMT           TO = CHRON-DATE-FMT                 *
*        FROM = CHRON-TIME-UFMT           TO = CHRON-TIME-FMT                 *
*-----------------------------------------------------------------------------*
P401-TEST1-UYMD-FMDY.
      MOVE SPACES                         TO PA-USER-INPUT
                                             CHRON-DATE-FMT
                                             CHRON-TIME-FMT.

      DISPLAY SPACES.
      DISPLAY " > Input YYMMDD: ".
      ACCEPT PA-USER-INPUT.
      MOVE PA-USER-INPUT                  TO CHRON-DATE-UFMT.
      MOVE SPACES                         TO PA-USER-INPUT.
      DISPLAY " > Input HHMMSS: ".
      ACCEPT PA-USER-INPUT.
      MOVE PA-USER-INPUT                  TO CHRON-TIME-UFMT.
      MOVE CHRON-UFMTYMD-FMTMDY           TO CHRON-MODE.
      PERFORM P555-CALL-CHRONOS          THRU P555-EXIT.
      DISPLAY "  Formatted MM/DD/YY = ", CHRON-DATE-FMT
      DISPLAY "  Formatted HH:MM:SS = ", CHRON-TIME-FMT.
P401-EXIT.
      EXIT.
```

```
*-------------------------------------------------------------------------*
*    This routine accepts an unformatted date of 'MMDDYY' and             *
*    converts it to another unformatted date of 'YYMMDD'.                 *
*       FROM = CHRON-DATE-UFMT          TO = CHRON-DATE-UFMT               *
*-------------------------------------------------------------------------*
P402-TEST2-UMDY-UYMD.
     MOVE SPACES                          TO PA-USER-INPUT.
     DISPLAY SPACES.
     DISPLAY " > Input MMDDYY: ".
     ACCEPT PA-USER-INPUT.
     MOVE PA-USER-INPUT                   TO CHRON-DATE-UFMT.
     MOVE CHRON-UFMTMDY-UFMTYMD           TO CHRON-MODE.
     PERFORM P555-CALL-CHRONOS          THRU P555-EXIT.
     DISPLAY " Unformatted YYMMDD =", CHRON-DATE-UFMT.
P402-EXIT.
     EXIT.
*-------------------------------------------------------------------------*
*    This routine accepts a Julian date of 'YYDDD' and converts           *
*    it to an unformatted date 'YYMMDD'.                                  *
*       FROM = CHRON-JUL-DATE            TO = CHRON-DATE-UFMT              *
*-------------------------------------------------------------------------*
P403-TEST3-JUL-UYMD.
     MOVE SPACES                          TO PA-USER-INPUT
                                             CHRON-DATE-UFMT.
     DISPLAY SPACES.
     DISPLAY " > Input Julian Date (YYMMM):".
     ACCEPT PA-USER-INPUT.
     MOVER PA-USER-INPUT                  TO PA-USER-JUL-DATE.
     MOVE PA-USER-JUL-YR                  TO CHRON-JUL-YR.
     MOVE PA-USER-JUL-DY                  TO CHRON-JUL-DY.
     MOVE CHRON-JUL-UFMTYMD               TO CHRON-MODE.
     PERFORM P555-CALL-CHRONOS          THRU P555-EXIT.
     DISPLAY " Unformatted YYMMDD = ", CHRON-DATE-UFMT.
P403-EXIT.
     EXIT.
$PAGE
*-------------------------------------------------------------------------*

*    This routine calls "CHRONOS" and checks for an error state.          *

*-------------------------------------------------------------------------*
P555-CALL-CHRONOS.
     MOVE 0                                  TO CHRON STATUS
                                                CHRON-RESULT.
     CALL INTRINSIC "CHRONOS"             USING CHRON-STATUS,
                                                CHRON-MODE,
                                                CHRON-STAMP,
                                                CHRON-DATE-FMT,
                                                CHRON-TIME-FMT,
                                                CHRON-DATE-UFMT,
                                                CHRON-TIME-UFMT,
                                                CHRON-DATE-SYMBOL,
                                                CHRON-TIME-SYMBOL,
                                                CHRON-INCREMENT,
                                                CHRON-STRING,
                                                CHRON-JUL-YR,
                                                CHRON-JUL-DY,
                                                CHRON-DAY-OF-WEEK,
                                                CHRON-CENTURY,
                                            GIVING CHRON-RESULT.
     IF CHRON-RESULT NOT EQUAL TO 0
        PERFORM P606-CHRONOS-ERROR THRU P606-EXIT.
P555-EXIT.
     EXIT.

$PAGE
```

```
*-------------------------------------------------------------------------*
*    This routine displays error information for CHRONOS.                 *
*-------------------------------------------------------------------------*
P606-CHRONOS-ERROR
     MOVE CHRON-STATUS                  TO CHRON-DISPLAY.
     DISPLAY "-WARN- CHRON STATUS = ", CHRON-DISPLAY.
     MOVE CHRON-RESULT                  TO CHRON-DISPLAY.
     DISPLAY "-WARN- CHRON RESULT = ", CHRON-DISPLAY.
     MOVE CHRON-MODE                    TO CHRON-DISPLAY.
     DISPLAY "-WARN- CHRON MODE   = ", CHRON-DISPLAY.
P606-EXIT.
     EXIT.
```

Figure 18.7 - COBOL Sample Calling CHRONOS

CHRONOS Error Messages

Listed below are the CHRONOS error message numbers and their respective meaning:

Number	Meaning
2	Source (29:3) not in bit range 000..101
3	Missing source parameter chronos_stamp
4	Missing source parameter formatted_time or formatted_date
5	Missing source parameter unformatted_time or unformatted_date
6	Missing source parameter julian_time or julian_date
7	chronos_string cannot be used as source
8	System Local cannot be used as the destination
9	Missing destination parameter chronos_string
10	Destination (26:3) not in bit range 000..101
11	chronos_string destination format not in bit range 000..011
12	Destination format not in bit range 000..010
13	Source format not in bit range 00..10
14	*NOT USED*
15	Missing destination parameter julian_time or julian_date
16	Missing destination parameter formatted_time or formatted_date
17	Missing destination parameter unformatted_time or unformatted_date
18	Missing destination parameter chronos stamp
19	Bad source numbers in one or both unformatted parameters
20	Bad source numbers in one or both julian parameters
21	Bad source numbers in one or both formatted parameters
22	Returned when something is wrong with the source or destination parameters (which was initially undetected), causing a conversion error.

Errors 19 - 21 are triggered when the following conditions apply:
For unformatted, formatted, or Julian conversions, these errors result when the numbers are not in range or are not formatted correctly. The CHRONOS function will return ASCII zeros in the destination field.

Error 22 is returned by CHRONOS when it finds a source or destination field that it does not understand.

Chapter 19

The CSEQ Tool

CSEQ reports the calling sequence of intrinsics. The intrinsic may be a native mode intrinsic, compatibility mode intrinsic, or both. Also CSEQ can report on user-defined intrinsic files via the SPLINTR or SYSINTR commands.

Operation

CSEQ is used to display native mode and compatibility mode intrinsic calling sequences as defined by either the SYSINTR or SPLINTR files. The default startup condition for CSEQ assumes that the user is interested in reporting on native mode intrinsics from SYSINTR.PUB.SYS, AIFINTR.PUB.SYS, PEINTR.PE.SYS, or SPLINTR.PUB.SYS. At that point it is simply a matter of entering the name of the intrinsic for which you are interested. See the sample output provided next for an illustration on how this works.

Native Mode Output

When CSEQ is asked to display the calling sequence of a native mode intrinsic, it generates output like the following example.

```
:cseq
CSEQ [2.0] - LPS Toolbox [A.01a]           (c) 1995 Lund Performance Solutions

For Help at the CSEQ prompt enter   ?


CSEQ [nm]: HPDEBUG

Procedure HPDEBUG (
   Parm_0       : (# actual parameters)    {R26}
   status       : anyvar record  ;         {R25, @32 -> 32} := nil
   cmdstr       : anyvar record  ;         {R24, @32 -> 8192} := nil
   itemnum1     :        int32   ;         {R23}
   itemval1     :        int32   ;         {SP-$0034}
   itemnum2     :        int32   ;         {SP-$0038}
   itemval2     :        int32   ;         {SP-$003c}
   itemnum3     :        int32   ;         {SP-$0040}
   itemval3     :        int32   ;         {SP-$0044}
   itemnum4     :        int32   ;         {SP-$0048}
   itemval4     :        int32   ;         {SP-$004c}
   itemnum5     :        int32   ;         {SP-$0050}
   itemval5     :        int32   )         {SP-$0054}
   {Item/value pairs:                                                    }
   {  1, file#  File# is an open file, which will be                     }
   {            used for Debug output. The value 1 is                    }
   {            ok, and means $STDLIST.                                  }
   {  2, welcome.  0 = don't print Debug's welcome                       }
   {            banner, 1 = print it (default = 1).                      }
   {Note: Recommended cmdstr:                                            }
   {    "~ignore ; {...your stuff...}; c~"                               }
   {    (tries to guarantee that an error in your stuff                  }
   {    won't leave the user in debug)                                   }
      extensible 2
      uncheckable_anyvar

CSEQ [nm]:
```

Figure 19.1 - Native Mode Intrinsic Calling Sequence

The first line of output means that the intrinsic HPDEBUG is in the SYSINTR file in UPPERCASE. If the procedure name had been reported in lowercase then that would be the exact name of the procedure. When you enter a procedure name in CSEQ, it first tries uppercase and then lowercase automatically.

For HPDEBUG, CSEQ noticed that it was an untyped-procedure. If it had a type (e.g., integer) then it would report it as a "**function... :integer**."

After reporting all of the parameters, CSEQ reports general information about the intrinsic. The intrinsic is marked as "**extensible 2**" and "**uncheckable anyvar.**" These are explained below:

extensible 2	The intrinsic must be called with at least the first two parameters and the number of actual parameters is passed in as a hidden value in register R26. Note the first two parameters might have default values.
uncheckable anyvar	Any parameters declared as "**anyvar**" normally have a hidden size parameter passed in just after the actual parameter. "**Uncheckable anyvar**" means that no hidden size parameters are passed in. If this intrinsic had not been "uncheckable" then it would have reported the location of the hidden size parameters.

Compatibility Mode Output

When CSEQ is asked to display the calling sequence of a compatibility mode intrinsic, it generates output like the following example.

```
:cseq
CSEQ [2.0] - LPS Toolbox [A.01a]          (c) 1995 Lund Performance Solutions

For Help at the CSEQ prompt enter   ?


CSEQ [nm]: cm

CSEQ [cm]: FCHECK

procedure FCHECK (
   filenum              : value integer,        ! Q - %11
   fserrorcode          : ref   integer,        ! Q - %10
   translog             : ref   integer,        ! Q - %7
   blocknum             : ref   double,         ! Q - %6
   numrecs              : ref   integer);       ! Q - %5
      option variable;    ! var mask @ Q - %4
      ! CCE: ok
      ! CCL: error: filenum not valid, or internal error
   ! popular errors:
   !   0 = EOF
   !   20 = invalid operation
   !   21 = Data parity error
   !   22 = Read timeout (see FCONTROL #4)
   !   23 = End of tape
   !   24 = device not ready
   !   25 = no write ring
   !   26 = transmission error
   !   32 = ABORTIO
   !   38 = tape parity error
   !   39 = recovered tape error (FSETMODE bit 12)
   !   43 = write exceeds record size
   !   50 = nonexistent account
   !   51 = nonexistent group
   !   52 = nonexistent perm file
   !   53 = nonexistent temp file
   !   54 = invalid file reference
```

```
CSEQ [cm]: DASCII

integer procedure DASCII (
   dword                : value double,          ! Q - %7 (2 halfs)
   base                 : value integer,         ! Q - %5
   string               : ref   byte array);     ! Q - %4
     := #chars                  ! result @ Q - %10
   ! Bases: 8, 10, -10, 16
   ! Note: bases 8 and 16 return "wrong" #chars.
   ! Note: -10 moves backwards.

CSEQ [cm]: exit
:
```

Figure 19.2 - Compatibility Mode Intrinsic Calling Sequence

All "**Q-**" addresses are valid as of the start of the intrinsic. Since most parameters are one halfword in size (16-bits), CSEQ doesn't list their size. Instead, only those parameters that are larger than one halfword are flagged with a size, as in "**Parm 1**" in DASCII.

In the above example, FCHECK is an untyped procedure and DASCII is type "**integer**" (returns a 16-bit value). Since DASCII returns a result, the stack storage location for the result is shown.

After all of the parameters, if any, CSEQ reports general information about the intrinsic. FCHECK was marked as "option variable," which means it has a parameter mask at Q-4. Option variable procedures with more than 16 parameters have a two-halfword parameter mask stored at Q-5 and Q-4.

Note: For some intrinsics, CSEQ displays detailed parameter information. FFILEINFO, for instance, has an additional 100 lines of itemnum information that can be displayed after the normal parameter list information. If you want the itemnum information only, precede the intrinsic name with a plus (+) sign. For example: **+ffileinfo**.

Capabilities

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

Usage

CSEQ can be run via the supplied UDC or with the MPE RUN statement. CSEQ can accept input through the INFO string parameter or directly from the user in query mode.

- UDC
  :CSEQ [< commands | [+]intrinsics >]
- RUN
  :RUN CSEQ.PUB.LPSTOOLS;INFO="[< commands | [+]intrinsics >]

Command Summary

The following list provides a simple description of CSEQ commands that you can use to quickly locate the command that suits the task at hand. *Note*: Portions of command codes are printed in uppercase to denote the part of the command that CSEQ requires in order to distinguish one command from another.

Command Code	Description
ALL	Lists all matching intrinsics for the current mode (CM, NM or BOTH).
ALLCM	Displays all CM intrinsics of a class
ALLNM	Displays all NM intrinsics of a class
BOTH	Displays both NM & CM intrinsic information
CLOSE	Closes a sysintr, splintr, or file #
CM	Displays CM intrinsic information only
Exit	Terminates CSEQ
HELP	Invokes CSEQ help
INTrinsic	Displays information about a specific intrinsic
NM	Displays NM intrinsic information only
SET/REset	Enables and disables options
SPLINTR	Opens an MPE V intrinsics file
STATus	Displays information about currently opened files
SYSINTR	Opens an MPE/iX intrinsics file

Command Definitions

This section describes CSEQ commands in detail.

ALL

The ALL command provides a means for listing all intrinsics or all intrinsics that have a common prefix. ALL will list all matching intrinsics for the current mode (CM, NM or BOTH).

ALLCM [intrinsic name]

This CSEQ command will display parameter information for the specified class of intrinsics. If an "intrinsic name" is not specified, then all of the compatibility mode intrinsics will be displayed. Partial names can be specified to display a class of intrinsics. *Note*: The plus (+) option, which displays **itemnum** information only, is not available for this command, but the minus (-) option, which disables extra information displays is available. For example: "ALLCM MY" could be used to display all of the intrinsics that start with the letters "MY".

ALLNM [intrinsic name]

This CSEQ command will display parameter information for the specified class of intrinsics. If an "intrinsic name" is not specified, then all of the native mode intrinsics will be displayed. Partial names can be specified to display a class of intrinsics. *Note*: The plus (+) option, which displays **itemnum** information only, is not available for this command. However, the minus (-) option can be used to disable extra information displays. For example: "ALLNM HP" could be used to display all of the intrinsics that start with the letters "HP."

BOTH

The BOTH command tells CSEQ to display the calling sequence for both Native Mode and Compatibility Mode intrinsics. *Note*: The plus (+) option, which displays **itemnum** information only, is not available for this command. The minus (-) option disables extra information displays.

After issuing BOTH, entering the intrinsic name ASCII would result in:

```
:cseq
CSEQ [2.0] - LPS Toolbox [A.01a]           (c) 1995 Lund Performance Solutions

For Help at the CSEQ prompt enter   ?


CSEQ [nm]: both

CSEQ [both]: ASCII

NM:
Function ASCII (
   word          :        UInt16   ;        {R26}
   base          :        int16    ;        {R25}
   string        : anyvar record   )        {R24}
      := #chars  : int16      {R28}
   {Bases: 10, 8, -10, and (MPE XL) base 16                                  }
   {Note: bases 8 & 16 return wrong # of characters!                         }
   {Note: -10 moves backwards.                                               }
      uncheckable_anyvar

CM:
integer procedure ASCII (
   word                 : value logical,         ! Q - %6
   base                 : value integer,         ! Q - %5
   string               : ref   byte array);     ! Q - %4
     := #chars                    ! result @ Q - %7
   ! Bases: 10, 8, -10, and (MPE XL) base 16
   ! Note: bases 8 & 16 return wrong # of characters!
   ! Note: -10 moves backwards.

CSEQ [both]: exit
:
```

Figure 19.3 - BOTH Command Screen

CLOSE ["sysintr" | "splintr" | <file #> | ALL]

The Close command limits CSEQ's scan to the specified intrinsic files. See examples for an illustration of how this works.

CM

The CM command tells CSEQ to display the calling sequence for Compatibility Mode intrinsics. If CM is followed by an intrinsic name, it is looked up immediately.

Exit

The Exit Command terminates CSEQ.

HELP

The HELP command invokes the CSEQ help facility.

INTrinsic <intrinsic_name>

The INTRINSIC command was added to CSEQ's command list so that a user could specify the name of an intrinsic which might also be the name of a CSEQ command. For example, "**CSEQ[NM] : intrinsic Help**."

NM

The NM command tells CSEQ to display the calling sequence for Native Mode intrinsics.

SPLINTR <filename>

The SPLINTR command tells CSEQ to look for Compatibility Mode intrinsics in a different intrinsic file. CSEQ opens the specified CM intrinsic file. If the new file cannot be opened, CSEQ will report an error and revert to SPLINTR.PUB.SYS.

STATus

The STATUS command tells CSEQ to display a brief report on which intrinsic files are being used.

SYSINTR <filename>

The SYSINTR command tells CSEQ to look for Native Mode intrinsics in a different intrinsic file. CSEQ opens the specified NM intrinsic file. If the new file cannot be opened, CSEQ will report an error and revert to SYSINTR.PUB.SYS.

SET | REset

The SET and RESET commands are used to specify the following options.

SET C	Display intrinsic declarations using C language syntax.
SET EXTRAS	Tells CSEQ that you want to see extra comments about intrinsics. SET NOEXTRAS suppresses the extra comments.
SET LANGuage	The SET LANGuage option is used to report the language in which each NM intrinsic was called. The language is shown as a number rather than its full name. Most intrinsics will come up as Language=1, however the CLOCK and PAUSE intrinsic will be displayed as language 0.
SET MACRO	The SET MACRO option is used to tell CSEQ to emit a Debug/iX macro to show the parameters of each subsequent intrinsic call. Use Reset MACRO to disable this option. CSEQ will change its prompt to look like CSEQ[nm,macro]: when this option is active. *Note*: SET PE performs an implicit RESET MACRO.

SET NOEXTRASONLY	Tells CSEQ that you want to see only extra comments above intrinsics and nothing about the actual calling sequence. This is useful for intrinsics with many parameters, like HPFOPEN. Default is RESET EXTRASLONLY or SET NOEXTRASONLY.
SET PARMS	When reset, tells CSEQ to list only the names of intrinsics and not any parameters or functional results. This is most useful in conjunction with the ALL command. When set, CSEQ lists the parameters of intrinsics.
SET PARMTRUNC	In SET, this option tells CSEQ that if it sees a parameter name beginning with an ellipsis (...), it should skip the rest of the parameters for the intrinsic. (CSEQ.DATA has two instances of such parameters: HPFOPEN and HPDEVCREATE.)
SET PE	This option is used to select whether or not CSEQ displays NM intrinsic parameters relative to the Procedure Exit (AIF) parameter data structure. Use RESET PE to de-select this option. Default is RESET PE.
SET PLUSPLUS	This option is used in conjunction with the UNNAMED option.
SET SORT	Determines whether or not the output is displayed in increasing alphabetical order. The default is SET SORT. Use RESET SORT to display output in non-alpha order.
SET UNNAMED	This option displays parameters as parm#1, parm#2, etc. rather than the name (i.e., length, buffer). The default is RESET UNNAMED.

CSEQ Examples

Following are some examples of the information discussed in the previous sections.

```
:cseq
CSEQ [2.0] - LPS Toolbox [A.01a]            (c) 1995 Lund Performance Solutions

For Help at the CSEQ prompt enter   ?


CSEQ [nm]: both

CSEQ [both]: print

NM:
Procedure PRINT (
   message      : anyvar record  ;        {R25, R26}
                                          {Address type = LongAddr}
   length       :        int16   ;        {R24}
   control      :        int16   )        {R23}
      uncheckable_anyvar

CM:
procedure PRINT (
   message              : ref   logical array,  ! Q - %6
   length               : value integer,        ! Q - %5
   control              : value integer);       ! Q - %4

CSEQ [both]: exit
:
```

Figure 19.4 - CSEQ Output Using the Both Option

Figure 19.5 shows how CSEQ's "**allnm**" command works. If a partial intrinsic name is given, then all intrinsics that match that partial description are displayed. In this example, two intrinsics matched the partial description.

```
:cseq
CSEQ [2.0] - LPS Toolbox [A.01a]            (c) 1995 Lund Performance Solutions

For Help at the CSEQ prompt enter   ?


CSEQ [nm]: allnm aiffile1


{sorting}
{Intrinsic file AIFINTR.PUB.SYS}

Procedure AIFFILELGET (
   Parm_0         : {# actual parameters}    {R26}
   status         : var     record  ;        {R25, @32 -> 32, align 32}
   itemnum_array: anyvar record  ;           {R24, @32 -> 32000, align 32}
   item_array     : anyvar record  ;         {R23, @32 -> 64000, align 32}
   item_status_array: anyvar record  ;       {SP-$0034, @32 -> 32000, align 32}
   fnum           :         int32   ;        {SP-$0038}
   pid            :         record  ;        {SP-$0040,8, #bits = 64} := 0
   ufid           : var     record  ;        {SP-$0044, @32 -> 160} := nil
                                             {align 32}
   uid            :         int32   )        {SP-$0048} := 0
      extensible 8
      uncheckable_anyvar

Procedure AIFFILELPUT (
   Parm_0         : {# actual parameters}    {R26}
   status         : var     record  ;        {R25, @32 -> 32, align 32}
   itemnum_array: anyvar record  ;           {R24, @32 -> 32000, align 32}
   item_array     : anyvar record  ;         {R23, @32 -> 64000, align 32}
   item_status_array: anyvar record  ;       {SP-$0034, @32 -> 32000, align 32}
   fnum           :         int32   ;        {SP-$0038}
   pid            :         record  ;        {SP-$0040,8, #bits = 64} := 0
   ufid           : var     record  ;        {SP-$0044, @32 -> 160} := nil
                                             {align 32}
   uid            :         int32   ;        {SP-$0048} := 0
   ver_item_nums: anyvar record  ;           {SP-$004c, @32 -> 32000} := nil
                                             {align 32}
   ver_items      : anyvar record  ;         {SP-$0050, @32 -> 64000} := nil
                                             {align 32}
   ver_item_statuses: anyvar record  )       {SP-$0054, @32 -> 32000} := nil
                                             {align 32}
      extensible 11
      uncheckable_anyvar


Found 4 NM intrinsics.

CSEQ [nm]: exit
:
```

Figure 19.5 - allnm Command

Figure 19.6 shows how the **"set pe"** command affects CSEQ's NM output. When enabled, this command is used to display an intrinsic's parameters as offsets from the parameter area of a procedure exit handler.

```
:cseq
CSEQ [2.0] - LPS Toolbox [A.01a]            (c) 1995 Lund Performance Solutions

For Help at the CSEQ prompt enter   ?


CSEQ [nm]: set pe

Note: SET PE means CSEQ will display parms as "PA-$####".
      PA means: Parms Area end, which is the third parameter to
      a Procedure Exit handler, and is found in R23.

ok

Option settings:
   C                : reset
   EXTRAS           : SET
   EXTRASONLY       : reset
   LANGuage         : reset
   MACro            : reset
   PARMS            : SET
   PARMTRUNC        : SET
   PE               : SET
   SORT             : SET
   UNNAMED          : reset

CSEQ [nm, pe]: print

Procedure PRINT (
   message       : anyvar record  ;        {PA-$0008,8}
                                           {Address type = LongAddr}
   length        :        int16   ;        {PA-$000a}
                                           {PA-$000c, #bits = 16 FILLER}
   control       :        int16   )        {PA-$000e}
                                           {PA-$0010, #bits = 16 FILLER}
      uncheckable_anyvar

CSEQ [nm, pe]: exit
:
```

Figure 19.6 - set pe Command

Figure 19.7 shows how to use the status and close commands.

```
:cseq
CSEQ [2.0] - LPS Toolbox [A.01a]          (c) 1995 Lund Performance Solutions

For Help at the CSEQ prompt enter   ?


CSEQ [nm]: status

Intrinsic files open:

   Mode  File#  File Name                      Address
   ----  -----  ---------------------------    --------
    NM      15  SYSINTR.PUB.SYS                  $4163917c
    NM      17  AIFINTR.PUB.SYS                  $41638cec

    cm      19  SPLINTR.PUB.SYS

(case insensitive)

CSEQ [nm]: close 19
closed.

CSEQ [nm]: status

Intrinsic files open:

   Mode  File#  File Name                      Address
   ----  -----  ---------------------------    --------
    NM      15  SYSINTR.PUB.SYS                  $4163917c
    NM      17  AIFINTR.PUB.SYS                  $41638cec


(case insensitive)

CSEQ [nm]: exit
:
```

Figure 19.7 - Status and Close Commands

CSEQ Error Messages

Message	***CSEQ does not have an open command***
Cause	User accidentally typed "open"
Action	Use **sysintr (filename)** or **splintr (filename)**

Chapter 20

The EZHELP Tool

EZHELP is a file-browsing tool for MPE HELP catalogs that brings the advantages of terminal-based windowing to these standard information resources. EZHELP is a dual-purposed application. First, EZHELP functions as a windowed replacement for MPE HELP. Popup, scrollable windows contain lists of topics and items for easy information retrieval. Related information is easily accessed for any MPE command. You can pop up window displays on **Examples, Parms,** and **Operations** in just a few keystrokes.

EZHELP also offers a way to interactively view any other help catalog you may have on your system, delivering the same kind of look and feel interface to these information resources that you find with the system help catalog running under EZHELP. Because these files follow a standard structure, EZHELP is able to read the structure and dynamically arrange the information into a format that can be used in a windowing environment.

Note: In this document, the "system help file" and "HELP catalog" refer to the MPE HELP formatted files. Refer to the HP manual entitled "*Message Catalogs Programmer's Guide*" for more detail.

About HELP Catalogs

HELP catalogs follow a certain syntax that arranges information according to a set structure. In general, information is grouped according to whether it is an Entry or an Item. Entries are high-level descriptors, including commands, system utilities, and so forth. Items are categories of information that are provided for each Entry. **Examples**, **Operation**, and **Parms** are typical Items. EZHELP organizes the Entry and Item information into window displays that complement each other.

Note: HELP catalog format statements (STARTHELP, STOPHELP, SUBSET, SUBITEM) cause no specific action in EZHELP. For example, *Figure 20.1* shows what CICAT.PUB.SYS (the file for MPE HELP) contains for the ABORT command.

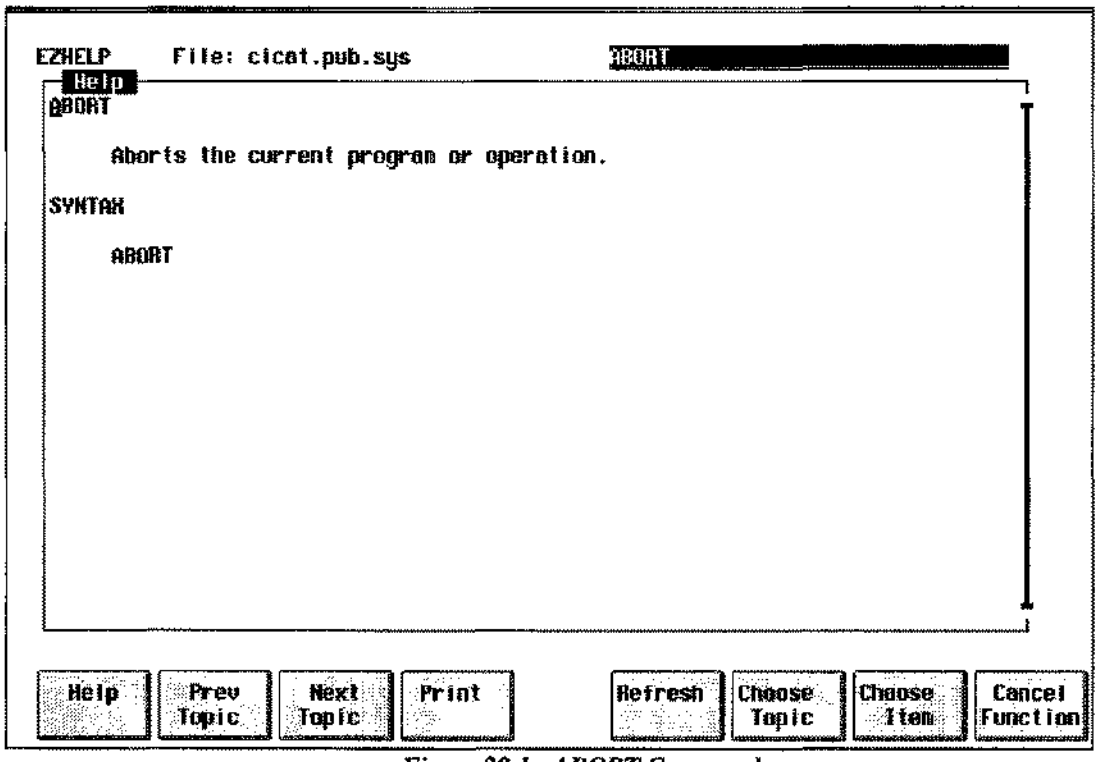


Figure 20.1 - ABORT Command

Operation

EZHELP uses a PC-style interface which allows you to choose actions from a menu. There are selections for opening up the system help file, as well as for a help catalog file of your choice. When you choose a help catalog file, EZHELP asks you to key in the name of the file.

An easier way to choose a help catalog file is to select it from a popup picklist of file names. To display the HELP Catalog Picklist, press the F2 [Picklist] key when EZHELP asks you for the file name (see the section on "*HELP Catalog Picklist*" for more information on what this is about.

If you want to directly access an MPE topic without having to work through the menu system, you can type "**ezhelp <MPE COMMAND>**" at the colon prompt.

How EZHELP Formats Information

When a help catalog is chosen, EZHELP opens the file and prepares it for display.

First, EZHELP reads the catalog file and creates a sorted list of all the Entry lines (ENTRYs are referred to as Topics in EZHELP). This list is placed in a scrollable window called "Topics." From this list, you select the Topic to be displayed.

Second, EZHELP finds the first Entry line in the help catalog and treats it as the first topic. So, when you run a help catalog under EZHELP, the initial topic displayed is the first Entry that EZHELP found in the file.

Displaying other topics is achieved by pressing the **Choose Topic** function key. When you press this key, EZHELP displays the sorted list it prepared when you first selected the catalog file. Use **Pg Dn** or **Pg Up** keys to scroll through the list, and press the **Return** key to choose the Topic to display.

Other function key operations are available to assist you with both Topic and Item selections. Refer to "*Function Keys*" for a complete description.

Cross-Reference Navigating

Hypertext-like cross-referencing is the ability to display information on related topics mentioned in the current window. Typically, related topics in the current window are selected by moving the cursor from one topic to the next. Having several topics in a single window, then, means that the user can branch to a new information display for a given topic just by pressing **Return** at a highlighted topic. True hypertext functionality implies that selectable topics in a window are informationally related to the current topic.

Since EZHELP has only a superficial knowledge of the data it is displaying, it would be impossible to provide the kind of true hypertext, relational links for sophisticated cross-reference support. However, it does provide a mechanism for superficial cross-referencing when the "Help" window is active (this is the window that contains text for a given topic).

Cross-referencing in EZHELP, then, is limited to accessing information displays for Topics mentioned in the current "Help" window (remember that Topics are those ENTRYs that are listed in the "Topics" window). EZHELP scans the current window to determine if there are any Topics in it. Any Topic, whether it is related to the current one or not, becomes a cross-reference candidate if it is in the current window.

Selecting a cross-reference Topic is done by pressing the letter "**t**." When "**t**" is pressed, EZHELP scans the text on the screen and then looks for matches against other Entry lines. If a match is found, the cursor will be positioned at the beginning of the match. Pressing **Return** will cause the screen to switch to the new Topic window. Similarly, moving the cursor to a word and then pressing **Return** will tell EZHELP to check the text by the cursor against the Topic list. Again, switching to the new Topic if a match occurs.

To return to a previous screen, press "**p**." Remember, the "**t**" and "**p**" options are only available in the Help window for a Topic.

Changing the HELP Catalog Picklist

The HELP Catalog Picklist is the picklist of help catalog file names used for selecting a catalog to display. This list is displayed by pressing the F2 **[Picklist]** key when EZHELP prompts for the catalog file name. You can add or remove catalog files from this list on an as-needed basis. For instance, the picklist provided with EZHELP may contain system help catalogs that your system doesn't have. If this is the case, you may want to remove these filenames from the list.

The picklist itself is stored in a flat file called EZHPCK.HELP. The first line in this file should not be modified. It contains formatting commands used to define the window. Right below this line are the names of the catalog files. Simply key in the name of the additional file, or delete unneeded file names as required. Because this window was defined with "unlimited" scrolling capabilities, the list can be as long as you need it to be.

Note: If you own MAGNET (part of ***System Managers Toolbox***), you can use it to build a help catalog picklist for you. For example, to scan the entire file system for help catalogs you could enter:

```
:magnet "-F@.@.@ -m -o mypick '\ENTRY' '\ITEM' "
```

This sends the filelist to the file "**mypick**" in LISTF,6 format. Then, add the format line to this file and rename it "**ezhpck.help.lpstools**." This replaces the file provided with EZHELP.

Capabilities

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

Function Keys

This section discusses function key operations that are specific to EZHELP.

PREV TOPIC

This function is used to return to the Topic previously displayed. Topics refer to the ENTRY topic, such as a command, tool, error, and so forth. It is similar to the PREVIOUS function that is standard across all windowed-based ***LPS-Tools***.

NEXT TOPIC

This function is used to display the next Topic in the catalog file where a Topic refers to the ENTRY topic (such as a command, tool, error, and so forth).

CHOOSE TOPIC

This function is used to choose a Topic from the Topic List. The **Pg Dn** and **Pg Up** keys are used to navigate the Topic List.

CHOOSE ITEM

This function is used to choose an Item for the currently selected Topic. It is operational only when a Topic is displayed. When you press the **Choose Item** key, EZHELP displays a picklist containing a list of ITEMS. These ITEMS include **Example**, **Operation**, and **Parm**.

Using EZHELP

This section provides step-by-step instruction for using the EZHELP program. It takes a tutorial-like approach that leads you through basic EZHELP operations. When you are finished with this section, you should have a very clear idea of how to use this tool.

Several screen captures are provided to guide you through each step.

Starting EZHELP

To start EZHELP, type "EZHELP" at the colon prompt and press **Return**.

MPE Help users have the option of directly displaying an MPE topic by typing "**ezhelp <mpe command>**". For example, typing "**ezhelp getlog**" displays the GETLOG Entry from MPE Help. Using EZHELP in this way bypasses the opening screen displays and prompts. For now, however, it is assumed that you will be using the EZHELP menus.

The next display shows the EZHELP main menu bar.

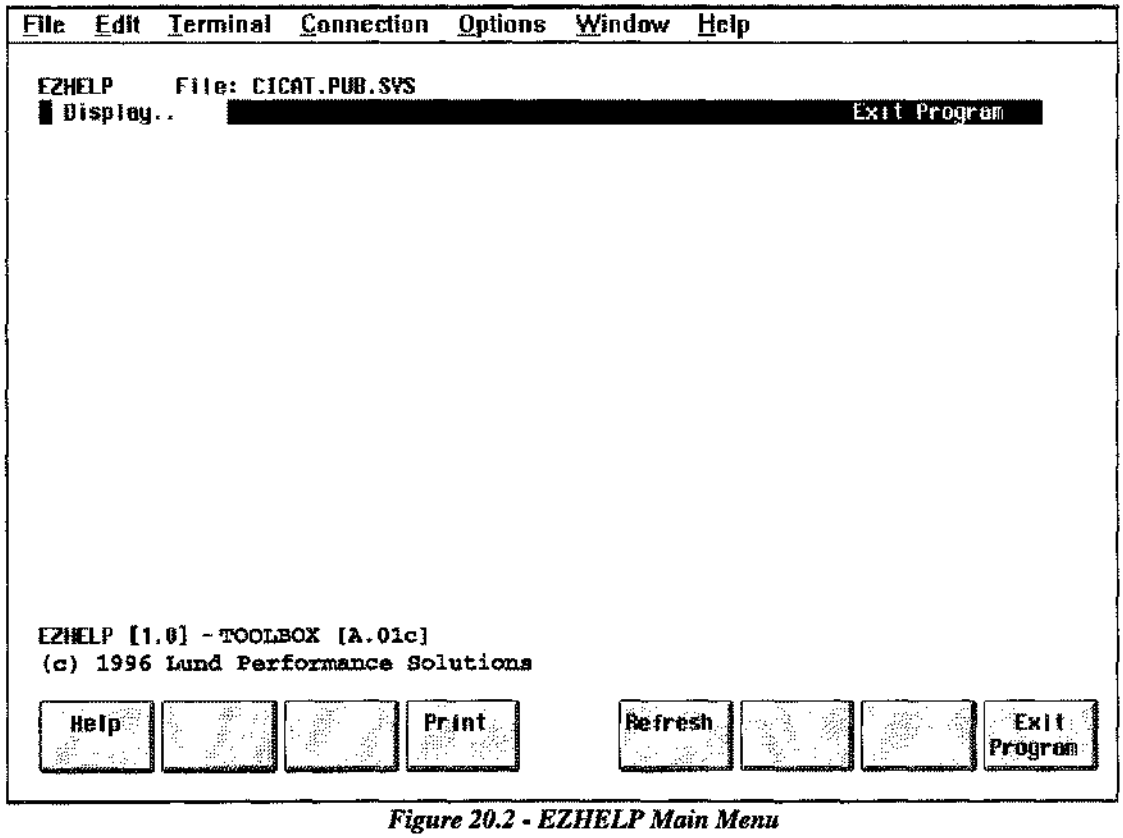


Figure 20.2 - EZHELP Main Menu

When you first run EZHELP, it displays copyright and banner information at the bottom of the screen. The top row of the screen is an information line that contains the EZHELP name and the name of the currently opened help catalog. The second row is a menu bar that functions as the main menu. The option to the far right, EXIT, terminates EZHELP and returns control to MPE. The other option, **Display**, is discussed next.

Using the DISPLAY Menu

The EZHELP main menu contains the Display pull-down menu option. From this menu, use **System help** to select the system help file (CICAT.PUB.SYS) or **Open** to select a Help catalog of your own choosing.

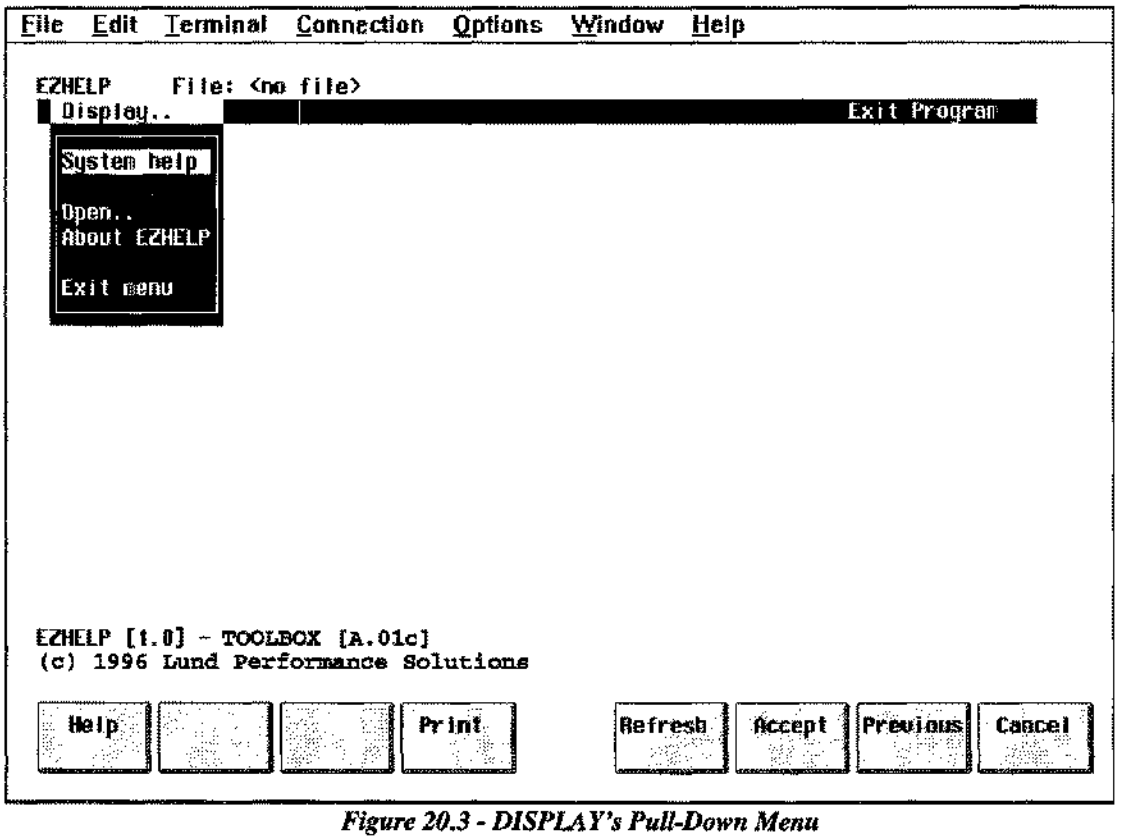


Figure 20.3 - DISPLAY's Pull-Down Menu

To select an item from the menu, use the arrow key to highlight the option and then press **Return**.

System help is used to navigate the system help facility. EZHELP enhances the use of system help by providing access to the information contained in the CICAT.PUB.SYS file in a windowed environment.

Open is used to select a help catalog of your choosing. This option is discussed in the section entitled "*Viewing Other HELP Catalogs*."

Using System Help

Running EZHELP's windowed interface for the system help catalog is achieved via the **System help** option. This section leads you through the various selections that are available, using examples to demonstrate the basic operations.

The next screen shows the System Help screen displayed when you select the **System help** option. For this version of CICAT, the first ENTRY is HELPMENU. Thus, HELPMENU becomes the first topic displayed when you select the **System help** option.

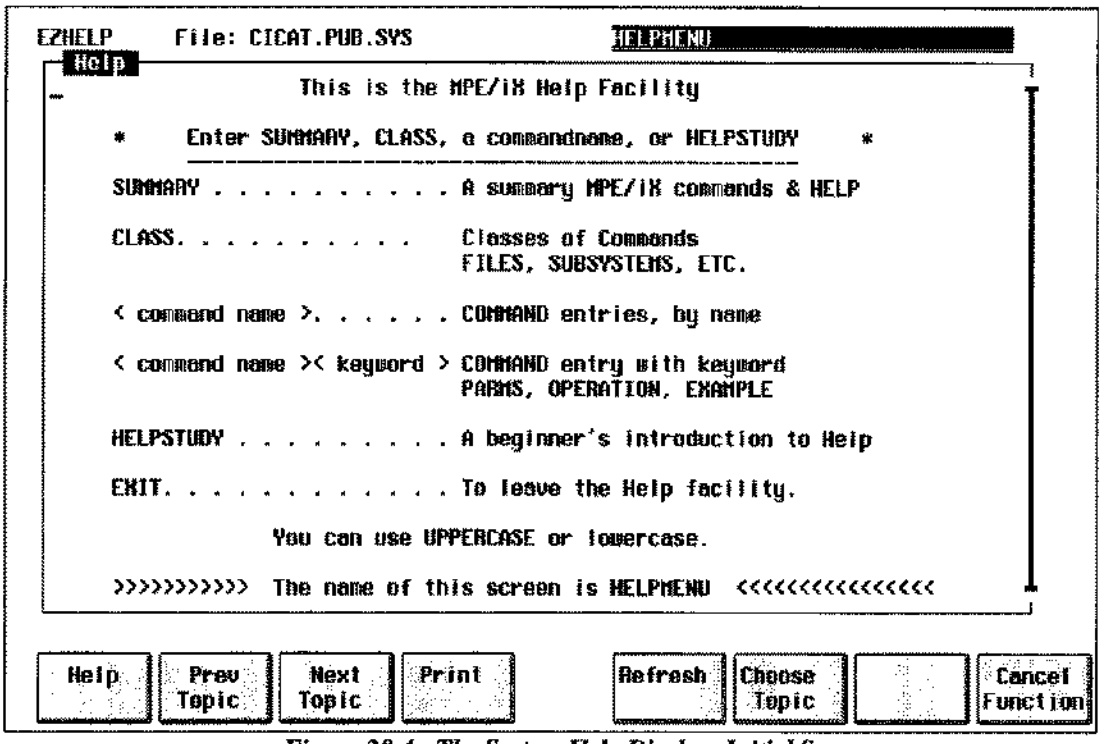


Figure 20.4 - The System Help Display: Initial Screen

Selection of topics is done through the **Choose Topic** function key. When you press this key, EZHELP pops up a scrollable window containing a list of all possible Topics.

Press F6 **[Choose Topic]** to display the pop up window of selectable topics.

The next screen shows the Topics window that is displayed whenever you select **Choose Topic**.

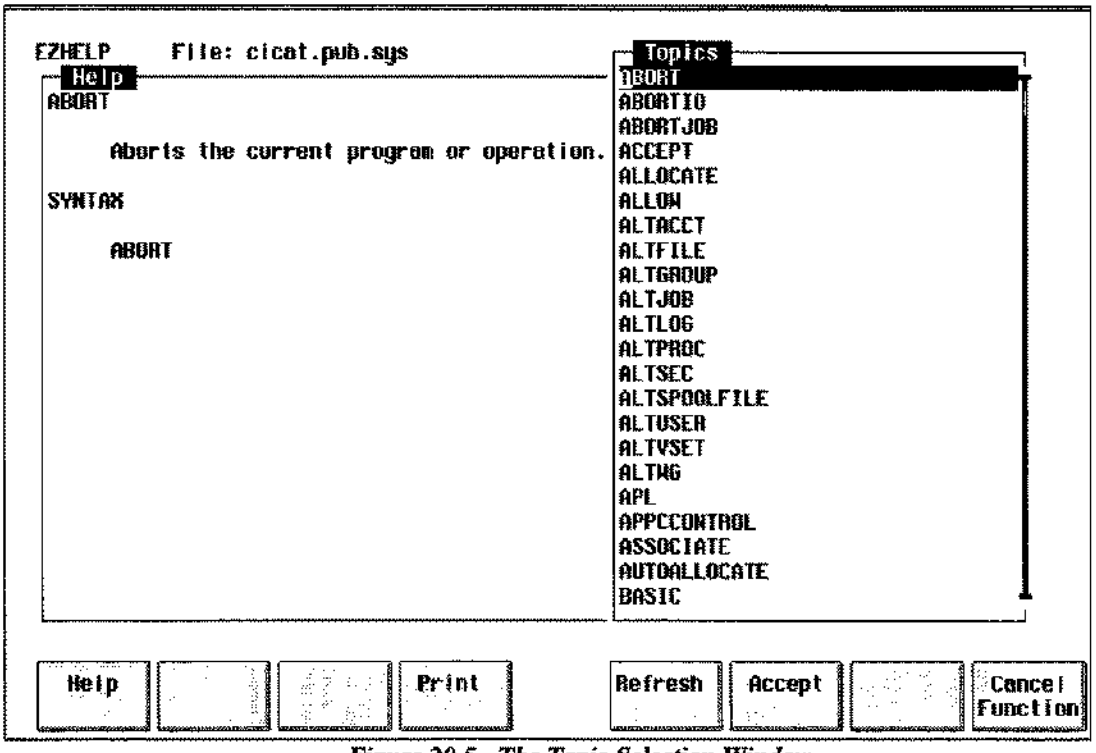


Figure 20.5 - The Topic Selection Window

Use the arrow keys to highlight a topic and then press **Return**.

For example, to find information on GETLOG, use the **Pg Dn** or arrow keys to highlight GETLOG in the Topics window and press **Return**.

The GETLOG entry is displayed next.

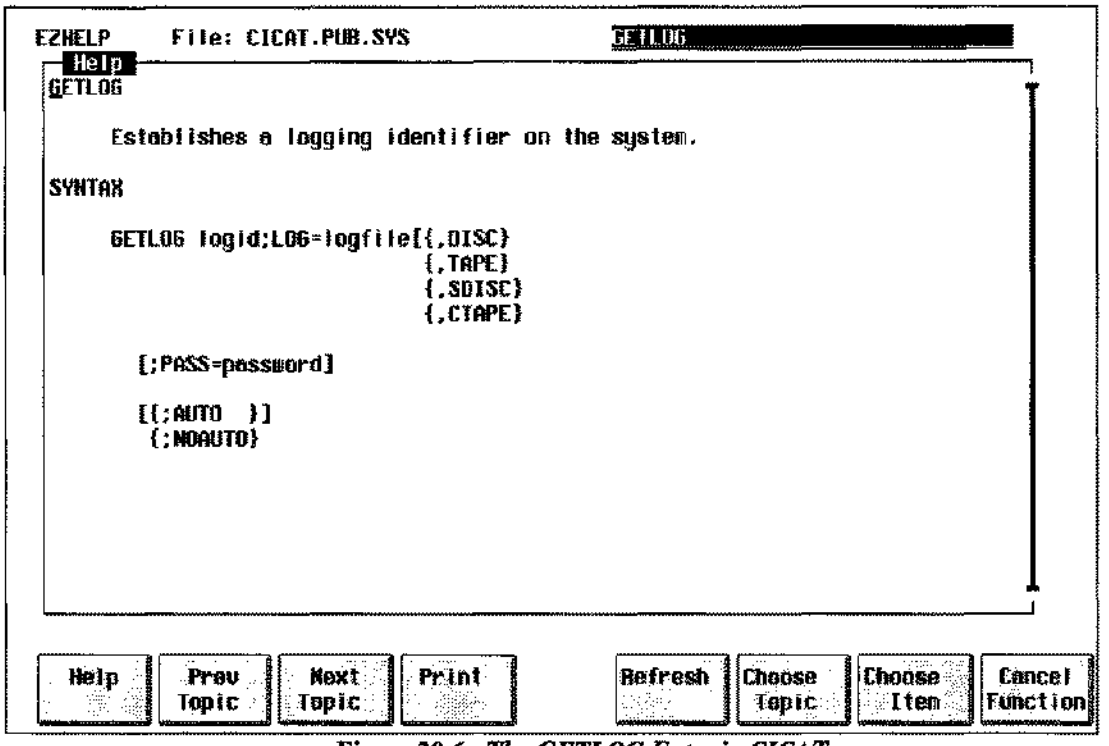


Figure 20.6 - The GETLOG Entry in CICAT

Notice that several function keys become operational once an entry is displayed.

Use the **Choose Item** function key to display additional information about the GETLOG command. For MPE commands in the CICAT file, the item choices are **Examples**, **Parms**, and **Operation**. Items are displayed in the popup, picklist window. As with any EZHELP picklist, use the arrow keys to select the item of interest and press **Return** to display the information.

The next screen display shows the Items pop up menu that is displayed when you press the F7 **[Choose Item]** function key.

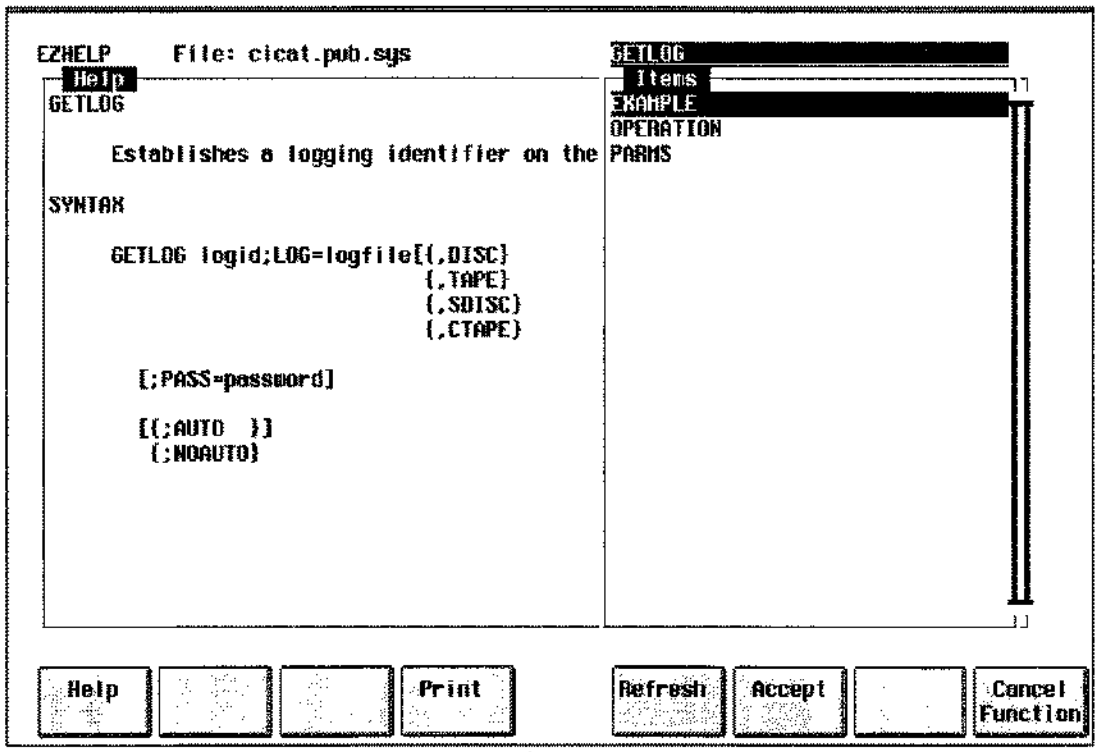


Figure 20.7 - The Item Selection Window

The screen below shows the Example text for GETLOG. Had you selected **Parms** or **Operation,** a screen containing information on those items would be displayed.

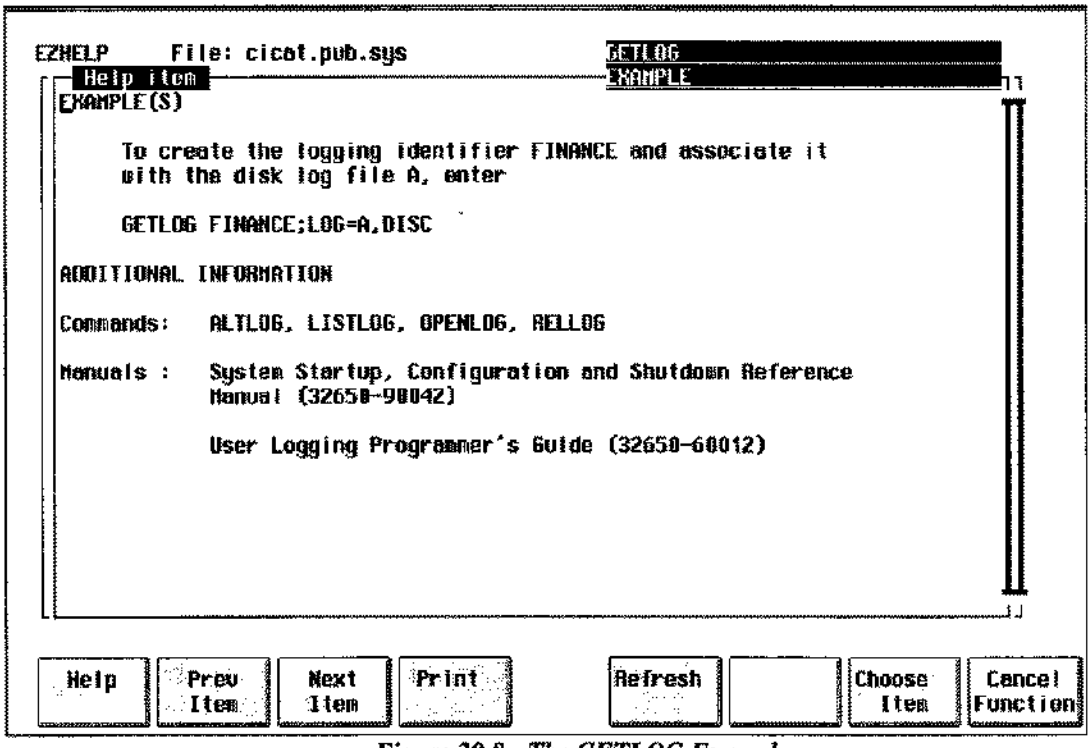


Figure 20.8 - The GETLOG Example

Viewing Other HELP Catalogs

EZHELP can be used to view any help catalog that follows the structure used by the MPE HELP catalog. To view one of these catalogs, choose the Open command in the Display pull-down menu.

EZHELP will open the file you specify, dynamically arranging the file contents into information window displays and pop up lists from which you may select items and topics as needed.

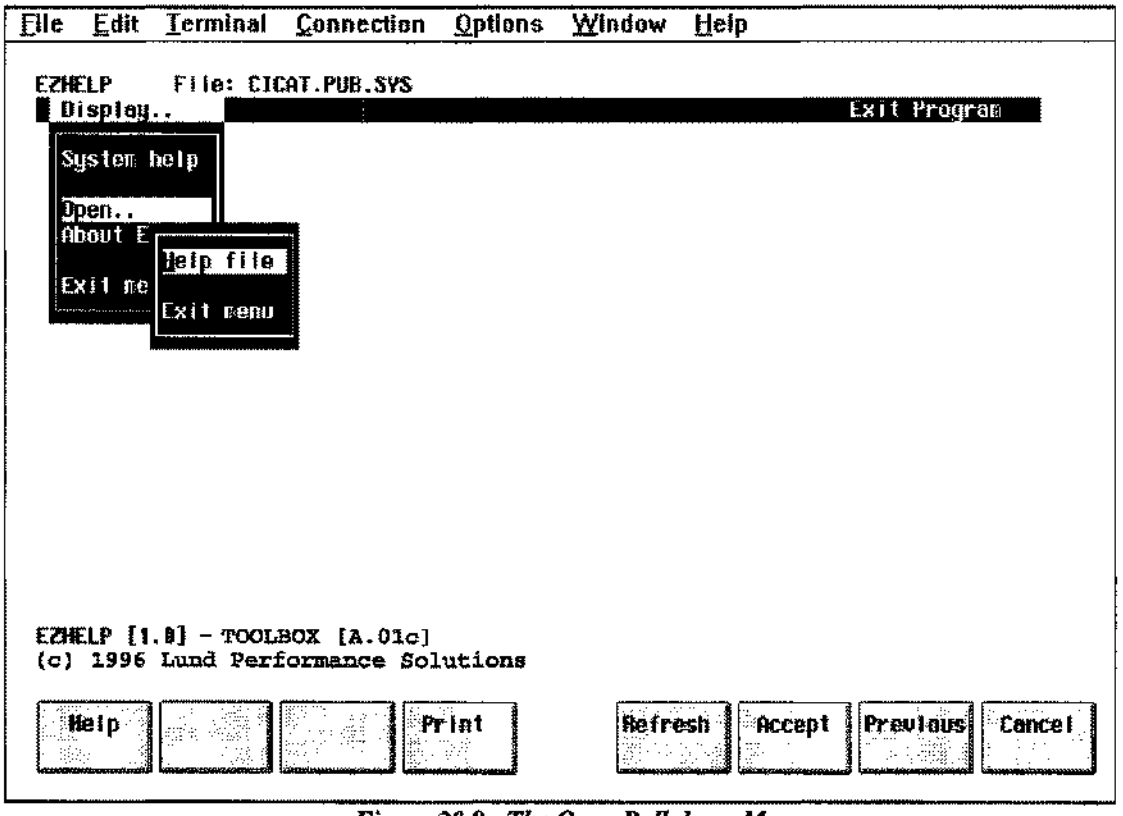


Figure 20.9 - The Open Pull-down Menu

Once you select "**Open**," a pull-down menu listing the available options appears. Use the arrow keys to choose an option and then press **Return**. The Help file option is used to specify the name of the catalog file you wish to view. When you choose this option, a window is displayed that prompts you for a filename.

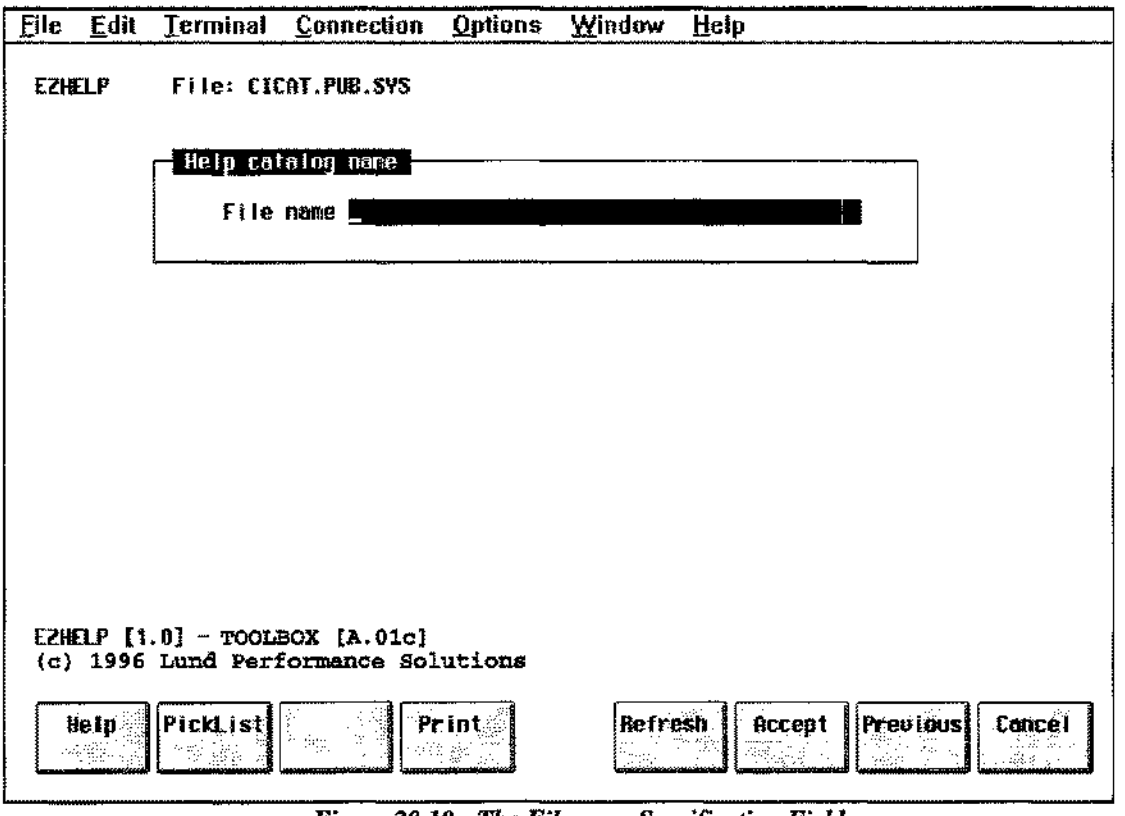


Figure 20.10 - The Filename Specification Field

Enter the filename of the help catalog you wish to view under the EZHELP interface in the **File name** field and press **Return**; or press F2 **[Picklist]** for the System Help Files Picklist. To select a help catalog file, use the arrow key to highlight the filename, then, press **Return** to open the file.

Other EZHELP Options

Other options that are selectable in the EZHELP menus include the **About EZHELP** option in the Display pull-down menu. The **About EZHELP** option simply lists the version information for the current release of EZHELP. This screen display is shown next.

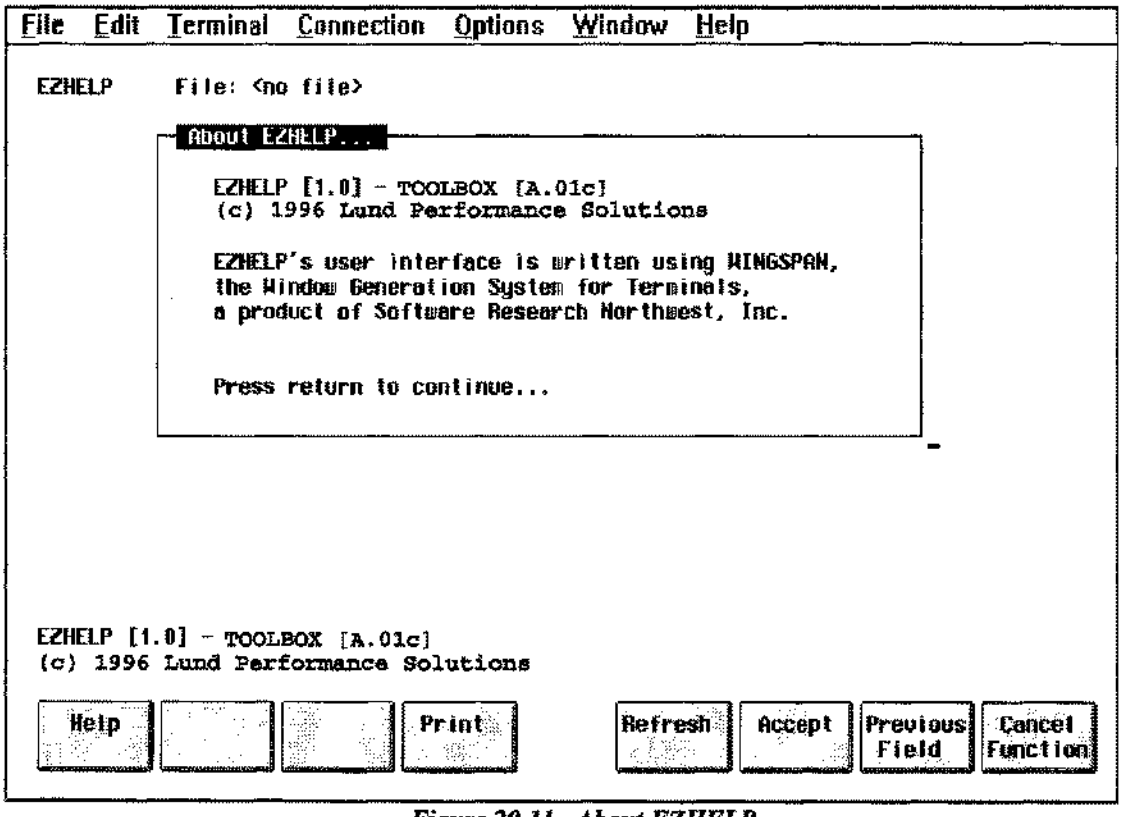


Figure 20.11 - About EZHELP

Should you need assistance in navigating through EZHELP, the context-sensitive help facility is always available to provide information about the task at hand.

To access Help, simply press the F1 [**Help**] function key. The help screen shown next is produced whenever you press F1 while the Display menu option in the EZHELP main menu is highlighted.

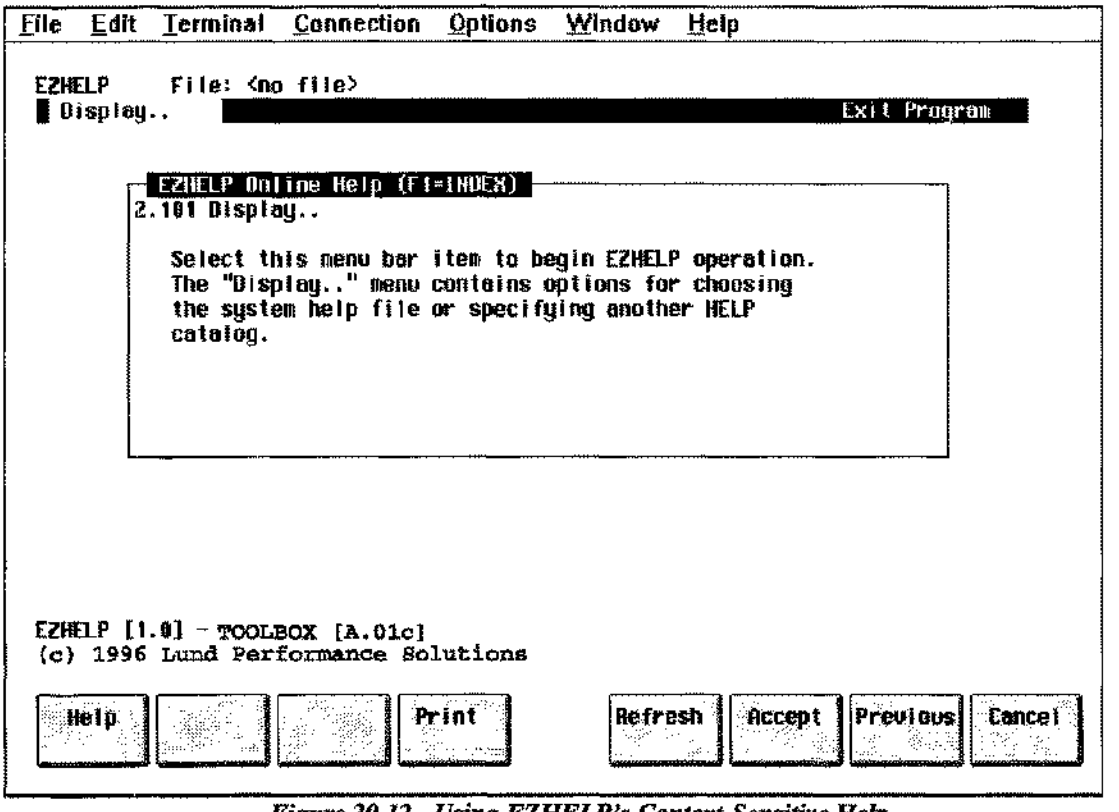


Figure 20.12 - Using EZHELP's Context-Sensitive Help

Chapter 21

The FASTLIB Tool

FASTLIB is a library of fast replacements for the standard Hewlett-Packard intrinsics: ASCII, DASCII, BINARY, DBINARY and CTRANSLATE. Given that many applications call these intrinsics hundreds of thousands of times, using FASTLIB provides significant savings in CPU time. FASTLIB intrinsics are provided in libraries for both the Classic machine and the Spectrum machine.

Operation

The five FASTLIB intrinsics are plug compatible with these standard HP intrinsics, which means no code changes are required to realize performance gains. For more detailed information on the use of these intrinsics, see the *MPE/iX Intrinsics Reference Manual*.

In addition to offering plug compatible replacements for these intrinsics, we also provide another choice. If your application does not use the return condition codes, you may choose to use the FASTRLIB library of intrinsics. Functionally equivalent to the standard FASTLIB intrinsics, these intrinsics differ only in that they omit the step of setting the return condition code which yields even greater performance.

The FASTLIB intrinsics differ from their HP equivalents in only two ways:

- They are much faster (see the timing discussion below).
- If a standard intrinsic wants to abort your process, it will do so with an intrinsic abort message. The FASTLIB intrinsics will abort in the same circumstances as the standard intrinsics, but without the same abort message (either a "FASTLIB abort" message will be displayed or an "invalid virtual address" message will appear.)

The FASTLIB libraries that you have received are:

FASTLIB.XL	Native mode executable libraries, sets condition codes
FASTRLIB.XL	Native mode executable libraries, no condition codes set
FASTLIB.O	Native mode object file, sets condition codes
FASTRLIB.O	Native mode object file, no condition codes set
FASTLIB.USL	USL (classic) object file, sets condition codes

Capabilities

No special capabilities are required to run FASTLIB.

Usage

Usage information is provided in two sections: one section for native mode usage, and a second section for compatibility mode.

Native Mode Usage

There are two ways to use FASTLIB from your native mode application.

Easiest

Simply run your program as usual except specify the FASTLIB library using the "XL=" option in the RUN statement. For example:

RUN MYNMPROG; XL="FASTLIB.XL.LPSTOOLS"

Optionally, you can specify the library at LINK time. For example:

LINK FROM=MYNMPROG.O; TO=MYNMPROG; XL="FASTLIB.XL.LPSTOOLS"

Fastest

LINK the FASTLIB intrinsics directly into your program for maximum performance. Doing so saves about 20 instructions per call to each FASTLIB intrinsic. For example:

LINK FROM=MYNMPROG.O, FASTLIB.O.LPSTOOLS; TO=MYNMPROG

Without Condition Codes

Many users do not use the condition codes set by ASCII, BINARY, DASCII, DBINARY and CTRANSLATE. *Note*: CTRANSLATE always sets the condition code to CCE. CPU time can be saved if FASTLIB does not set condition codes.

FASTRLIB is a version of FASTLIB that does not set condition codes for any of the procedures. It is used exactly like the FASTLIB (see above); the only difference is that you specify FASTRLIB instead of FASTLIB.

Compatibility Mode Usage

For compatibility mode users the only option that Lund Performance Solutions provides for the use of the FASTLIB intrinsics is to PREP them into your program. For example:

SEGMENTER
- USL MYUSL
- AUX FASTLIB.USL.LPSTOOLS
- COPY SEGMENT,FASTLIB

What's Next

The following section illustrates the calling sequence for each intrinsic (see CSEQ in the ***Developers Toolbox*** to see how this was generated), plus a brief description of each intrinsic. For a more detailed discussion, please refer to the *MPE/iX Intrinsic Reference Manual.*

ASCII

Purpose: Convert a 16-bit number into the equivalent ASCII string.

```
Function ASCII (
    word              :       UInt16          ;       {R26, #bits = 16}
    base              :       int16           ;       {R25, #bits = 16}
    string            : anyvar record         )       {R24, #bits = 65536}
      := #chars       : int16          {R28}
      {  Bases:   10, 8, -10                   }
      {           and (MPE XL) : 16            }
      {  Note:    bases 8 & 16 return          }
      {           "wrong" #chars!              }
      {  Note:    -10 moves backwards.         }
      uncheckable_anyvar
```

Figure 21.1 - Convert to ASCII

BINARY

Purpose: Convert an ASCII string into a 16-bit number.

```
Function BINARY (
    string                : anyvar record       ;       {R26, #bits = 65536}
    length                :       int16         )       {R25, #bits = 16}
          := binary#      : UInt16         {R28}
          uncheckable_anyvar
```

Figure 21.2 - Convert from ASCII

DASCII

Purpose: Convert a 32-bit number into the equivalent ASCII string.

```
Function DASCII (
    dword                 :       int32           ;       {R26, #bits = 32}
    base                  :       int16           ;       {R25, #bits = 16}
    string                : anyvar record         )       {R24, #bits = 65536}
          := #chars       : int 16         {R28}
          uncheckable_anyvar
```

Figure 21.3 - Convert to Equivalent ASCII String

DBINARY

Purpose: Convert an ASCII string into a 32-bit number.

```
Function DBINARY (
    string                : anyvar record  ;      {R26, #bits = 65536}
    length                :       int16    )      {R25, #bits = 16}
          := binary#      : int32      {R28}
          uncheckable_anyvar
```

Figure 21.4 - Converted ASCII String

CTRANSLATE

Purpose: Converts a string of EBCDIC or ASCII characters from one to the other, or between EBCDIK and KANA8. Or, translate via a user-supplied translation table.

```
Procedure CTRANSLATE (
    code            :          int16    ;    {R26, #bits = 16}
    instring        :   anyvar record   ;    {R25, #bits = 65536}
    outstring       :   anyvar record   ;    {R24, #bits = 65536} := nil
    stringlength    :          int16    ;    {R23, #bits = 16}
    table           :   anyvar record   )    {SP - $0034, #bits = 65536} := nil
        uncheckable_anyvar
```

Figure 21.5 - Conversion

Timing

How much faster are the FASTLIB intrinsics? When they were originally written, the FASTLIB intrinsics were up to 20 times faster than the system intrinsics. Although the system intrinsics have been optimized since FASTLIB first became available, FASTLIB intrinsics should still be considered as a high-performance alternative.

Two test programs are provided so that you can measure the performance gains provided by the FASTLIB intrinsics.

TIMEMPE.TIMING	test program uses HP intrinsics
TIMEFAST.TIMING	same program, uses FASTLIB intrinsics

To run these programs just use one of the RUN statements below:

RUN TIMEMPE.TIMING.LPSTOOLS
RUN TIMEFAST.TIMING.LPSTOOLS

FASTLIB Examples

Following are some examples of the FASTLIB tool:

```
:run timempe.timing.lpstools

TIMEMPE : times MPE XL intrinsics , #loops = 10000.  HPCPUNAME = SERIES 917LX
FRI, DEC 15, 1995,  2:37 PM

Loop Overhead:              3 milliseconds CPU, avg =       0 (MPE XL)
ascii (12345, 10)         236 milliseconds CPU, avg =      23 (MPE XL)
ascii (12345, -10)        216 milliseconds CPU, avg =      21 (MPE XL)
ascii (12345, 8)          167 milliseconds CPU, avg =      16 (MPE XL)
ascii (12345, 16)         149 milliseconds CPU, avg =      14 (MPE XL)
binary (12345)            305 milliseconds CPU, avg =      30 (MPE XL)
binary (%123456)          351 milliseconds CPU, avg =      35 (MPE XL)
binary ($abcd)            433 milliseconds CPU, avg =      43 (MPE XL)
ctranslate (80 byte)     5460 milliseconds CPU, avg =     546 (MPE XL)
dascii (123456, 10)       220 milliseconds CPU, avg =      22 (MPE XL)
dascii (123456, -10)      235 milliseconds CPU, avg =      23 (MPE XL)
dascii (123456, 8)        223 milliseconds CPU, avg =      22 (MPE XL)
dascii (123456, 16)       188 milliseconds CPU, avg =      18 (MPE XL)
dbinary(123456)           318 milliseconds CPU, avg =      31 (MPE XL)
dbinary(%123456)          333 milliseconds CPU, avg =      33 (MPE XL)
dbinary($abcdef)          377 milliseconds CPU, avg =      37 (MPE XL)

Note: loop overhead is NOT subtracted from any timings.

Total CPU time = 9309, elapsed = 9594 milliseconds.

END OF PROGRAM
:
```

Figure 21.6 - Running the TIMEMPE Program

```
:run timefast.timing.lpstools

TIMEFAST: times FASTLIB routines  , #loops = 10000.  HPCPUNAME = SERIES 917LX
FRI, DEC 15, 1995,  2:38 PM

Loop Overhead:              3 milliseconds CPU, avg =        0 (FASTLIB)
ascii (12345, 10)         152 milliseconds CPU, avg =       15 (FASTLIB)
ascii (12345, -10)        155 milliseconds CPU, avg =       15 (FASTLIB)
ascii (12345, 8)           91 milliseconds CPU, avg =        9 (FASTLIB)
binary (12345)            218 milliseconds CPU, avg =       21 (FASTLIB)
binary (%123456)          207 milliseconds CPU, avg =       20 (FASTLIB)
binary ($abcd)            209 milliseconds CPU, avg =       20 (FASTLIB)
ctranslate (80 byte)      299 milliseconds CPU, avg =       29 (FASTLIB)
dascii (123456, 10)       192 milliseconds CPU, avg =       19 (FASTLIB)
dascii (123456, -10)      193 milliseconds CPU, avg =       19 (FASTLIB)
dascii (123456, 8)        120 milliseconds CPU, avg =       12 (FASTLIB)
dbinary(123456)           254 milliseconds CPU, avg =       25 (FASTLIB)
dbinary(%123456)          210 milliseconds CPU, avg =       21 (FASTLIB)
dbinary($abcdef)          250 milliseconds CPU, avg =       25 (FASTLIB)

Note: loop overhead is NOT subtracted from any timings.

Total CPU time = 2638, elapsed = 2737 milliseconds.

ascii/dascii with base 16...

ascii (12345, 16)          87 milliseconds CPU, avg =        8 (FASTLIB)
dascii (123456, 16)       109 milliseconds CPU, avg =       10 (FASTLIB)

END OF PROGRAM
:
```

Figure 21.7 - Running the TIMEFAST Program

FASTLIB Error Messages

Errors generated by FASTLIB are the same as those generated by their HP equivalents. See the HP *Intrinsic Reference Manual* for possible error conditions.

Chapter 22

The WILDCARD Tool

The WILDCARD tool is a library of procedures that provide functionality not inherent in any programming language or environment. Functionally, the WILDCARD library provides solutions for two common programming tasks. First, it offers the ability to build a fileset from a complex fileset specification. This ability expands on LISTF-style operations so that you can add, subtract, or otherwise qualify groups of files for use in your programs. Second, WILDCARD provides a way to match patterns in string expressions (e.g., filename expressions).

The WILDCARD tool, then, is actually two groups of callable procedures: FILESET procedures and PATTERN procedures.

The FILESET procedures include:

getfileset
buildfileset
buildfilename
fileseterrmsg
fs_version

The PATTERN procedures include:

pattern_build
pattern_match
pattern_fga_match
check_fga_wildcard

Note: POSIX (HFS) file structures are not currently supported.

FILESET Procedures

In order to provide maximum flexibility, the FILESET building tasks have been broken into five separate procedure calls. The generated fileset is stored in an ASCII flat file so that you can access it as best suits your needs. The section called "Output Format" provides details on the layout of the file. The complete syntax that can be used to specify a file is described in the section called "*Fileset Syntax.*" If you are familiar with the MAGNET or BLAZE tools (included in the ***System Managers Toolbox***), then you may be familiar with this syntax already.

The **getfileset** procedure allows you to build a fileset with a single procedure call. If this call is not flexible enough for your needs, you may want to use the procedures **buildfilename** and **buildfileset**. These procedures provide more latitude for building the fileset the way you need it.

The remaining procedures, **fs_version** and **fileseterrmsg**, are used to provide the version string of the FILESET procedures and the error text for a specified error code.

FILESET Syntax

This section outlines the syntax used in the various fileset procedures.

```
<fileset>
       = <file set descriptor>
         [ [ <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. Or, an indirect file. }
<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>
            | "CCTL"    <onoroff>
            | "RIO"     <onoroff>
            | "MSG"     <onoroff>
            | "CIR"     <onoroff>
            | "REC"     <relop> <numeric value>
            | "TEMP"
            | "ASCII"
            | "BINARY"
            | "FIXED"
            | "VARIABLE"
            | "UNDEFINED"
<onoroff>
         = "=" { "ON" | "OFF"}
<relop>
         = "="| "< >"| "<"| "<="| ">="| ">"
<date>
         = {a date in the format yy/mm/dd or yymmdd}
           | "TODAY"
```

Figure 22.1 - WILDCARD Extended Fileset Syntax

Note: All literals are case-insensitive.

For further information, you may wish to refer to Appendix B, which features a list of the more common file codes, and Appendix C, which provides a convenient reference for LISTF WILDCARD syntax.

Output Format

This section presents the output format of the FILESET procedures.

File structure: 80 byte, fixed, ASCII

Bytes	Item
0 .. 7	Account name
8 .. 15	Group name
16 .. 23	File name
24 .. 28	File code
29 .. 37	Record size
38 .. 41	File type
42 .. 53	End-of-file
54 .. 64	File limit
65 .. 68	Blocking factor
69 .. 79	Sectors

Operation

All of the FILESET procedures are callable from either Native Mode or Compatibility Mode.

The Native Mode version follows the Procedure Calling Convention established by Hewlett-Packard and is therefore callable from any language following these conventions.

For Compatibility Mode, follow the rules established by Hewlett-Packard for parameter passing and segmentation (i.e., not callable from CCS/C CREL format programs).

Two levels of integration are provided so that you can choose the method that best suits your needs. The first level is simply to call the procedures as you would call an intrinsic. FILESET procedures can be accessed in much the same manner as intrinsics are accessed. The second method may be a better choice if a greater level of control is desired. In this case, you would merge the declaration files into your source, and then recompile and link the program.

GETFILESET

The purpose of this procedure is to build a fileset based on the fileset specification string that is passed to this procedure.

Syntax:

short int getfileset(expression)

Return Value

getfileset returns a 16-bit integer encoded as follows:

Code	Definition	Description
< 0	**: Error**	where the absolute value is the error number. This can be passed to **fileseterrmsg** to retrieve the error text.
0	**: No error**	where the resulting fileset is in the temporary file FILES.
> 0	**: Warning**	where the value is the number of characters processed from the provided fileset specification string.

Parameters

expression Byte array (required). It contains the NULL (ASCII 0) terminated fileset specification string. For a complete discussion of fileset specifications see Appendix C

Operation

To use this routine, all that is required is to declare **getfileset** as an external procedure. Depending on the language used, this may occur automatically. Then, compile your application and link with either the WILDCARD object file, relocatable library or executable library. After calling **getfileset**, check the return value for errors. If no error occurred, the resultant fileset can be accessed through the temporary file called "FILES." *Note*: FILES cannot be file equated. See the sample code TESTGFS.C.LPSTOOLS for an example.

BUILDFILENAME

This procedure is used to complete the building of the filename based on the specified mode. Five different modes are available ranging from fully-qualifying the filename to generating a unique filename. No errors are possible. The filename will be constructed using the standard MPE filename format (i.e., filename.group.account).

Syntax

buildfilename (filename, mode, terminator);

The Parameter Set is listed next, where each parameter is either an integer, character array, or integer array.

Parameter	Name	Type	Comment
1	**filename**	character array	Required
2	**mode**	short int	Required
3	**terminator**	short int	Required

Return Value

There is no Return Value.

Parameters

filename Byte array (required). Modes 0, 1, 2, and 3 contain the space character (ASCII 32) terminated filename. For mode 4, this filename will contain the unique filename generated by the call.

For all modes, the array will be terminated with the character provided in the parameter terminator. The dot (.) separator should not be specified. No filename validation will occur.

modes Short Int (required). Recognized values range from 0 to 4. The definitions are as follows:
0 = Append the logon group and account to the specified filename
1 = Append the logon account to the specified filename
2 = Append the program group and account to the specified filename
3 = Append the program account to the specified filename
4 = Generate a unique filename in logon group

If an unknown mode is given, then the terminator is appended to **filename.**

terminator Short Int (required). It is used to specify the character that will be used to terminate the byte array filename.

Operation

To use this routine, declare **buildfilename** as an external procedure. Depending on the language, this may occur automatically. Compile your application and link with either the WILDCARD object file, relocatable library or executable library. Before calling **buildfilename**, determine which mode you want to use. Then, for modes 0 through 3, initialize the parameter filename. For all modes, initialize the terminator parameter before calling **buildfilename**.

The result of all operations will be in the byte array filename. The format for the filename will be in MPE format. Any values filled in by the call will be in uppercase. Groups, accounts and filenames will be separated by dots (.). The filename will be terminated with the character specified by the terminator parameter.

No errors are possible; calling **buildfilename** with an invalid mode will simply result in the filename being terminated by the terminator you provided. Also, calling **buildfilename** without a filename (for modes 0 through 3) will not cause an error, however, the resulting filename may not be very useful. For example, see the sample code TESTFS.C.LPSTOOLS or TESTFS.SPL.LPSTOOLS.

BUILDFILESET

This procedure will generate the fileset specified by the expression. The fileset will be stored in the file given by the parameter filename and the domain will be determined by the boolean value of perm. The **stat** parameter is a two element array. The 0^{th} element contains the status, the 1^{st} element contains an error code if the 0^{th} element is non-zero. The procedure return value equals the number of characters processed from the **expression** parameter.

Syntax

short int buildfileset (expression, filename, perm, stat);

The Parameter Set is listed next where each parameter is either an integer, character array, or integer array.

Parameter	Name	Type	Comment
1	**expression**	character array	Required
2	**filename**	character array	Required
3	**perm**	logical	Required
4	**stat**	short int array	Required

Return Value

buildfileset returns a 16-bit integer that represents the number of characters processed from the expression string. Nominally, this equals the length of **expression**.

Parameters

expression Byte array (required). This parameter contains the NULL (ASCII 0) terminated fileset specification string. See the Fileset Specification Syntax in Appendix C for a complete discussion of fileset specifications.

filename Byte array (required). It contains the NULL (ASCII 0) terminated string used to build a file to hold the result of the **buildfileset** call. It cannot be file equated.

perm Logical (required). It contains a value of true (even) or false (odd) used to indicate if the output file should be a permanent or temporary file.

stat Short Int array (required). It contains the status of the call to **buildfileset**. Stat(0) returns the status of the call. A nonzero value indicates an error. The nonzero code can be optionally passed to **fileseterrmsg** to retrieve the error text.

Operation

To use this routine, declare **buildfileset** as an external procedure. Depending on the language, this may occur automatically. Compile your application and link it with either the WILDCARD object file, relocatable library or executable library. After calling **buildfileset**, check the status variable **stat** to determine if the call was successful. Also, check the return value to determine if the entire expression was processed. If the variable stat equals zero, then the resultant fileset can be accessed through the file specified by the parameter **filename** (see the sample code TESTFS.C.LPSTOOLS or TESTFS.SPL.LPSTOOLS).

FILESETERRMSG

The purpose of **fileseterrmsg** is to provide and format the text describing the error returned from a **buildfileset** or **getfileset** call.

Syntax

short int fileseterrmsg (status, buffer);

The Parameter Set is listed next, where each parameter is either an integer array or character array.

Parameter	Name	Type	Comment
1	**status**	short int array	Required
2	**buffer**	character array	Required

Return Value

The integer value returned by **fileseterrmsg** is the byte length of the text that has been placed in buffer.

Parameters

status Short Int array (required). It contains the status of the call to **buildfileset.**

status(0) is the error number and it is used to look up the text of the error message.

status(1) (if non-zero) is appended to the end of the error text. The format used is: "**info: <status(1)>**". Its use is purely informational. Most of the time when "**status(1)**" is non-zero, it will represent the error number returned by the intrinsic FCHECK plus some kind of file system error.

buffer Byte array (required). The length must be at least 80 bytes.

Operation

To use this entry point, declare **fileseterrmsg** as an external. Depending on the language, this may occur automatically. Compile your application and link with either the WILDCARD object file, relocatable library or executable library (see the sample code TESTFS.C.LPSTOOLS or TESTFS.SPL.LPSTOOLS).

FS_VERSION

This procedure will obtain the FILESET version string.

Syntax

fs_version (buffer);

Return Value

There are no Return Values.

Parameters

buffer Byte array (required). The length must be at least 80 bytes.

Operation

To use this routine, declare **fs_version** as an external procedure. Depending on the language, this may occur automatically. Compile your application and link with either the WILDCARD object file, relocatable library or executable library. After calling **fs_version**, the byte array buffer will contain the ASCII version string. This can be used to test FILESET versions to ensure compatibility of applications that use FILESET (see the sample code TESTFS.C.LPSTOOLS or TESTFS.SPL.LPSTOOLS).

Fileset Error Numbers and Meanings

stat(0)	meaning	stat(1) meaning
7	error during fclose	error number from FCHECK
8	error during fcontrol	error number from FCHECK
9	error during fopen,new	error number from FCHECK
10	error during fopen,old	error number from FCHECK
11	error during fread	error number from FCHECK
12	error during file rename	error number from FCHECK
13	error saving file	error number from FCHECK
14	error during fwrite	error number from FCHECK
21	error closing listf temporary file	error number from FCHECK
22	error opening listf temporary file	error number from FCHECK
23	error reading listf temporary file	error number from FCHECK
2	error from command intrinsic	error number from COMMAND
29	error during listf command	error number from COMMAND
1	expected alphabetic or numeric	not used
3	expected date	not used
4	bad filename part	not used
5	bad groupname part	not used
6	bad accountname part	not used
15	bad 16 bit integer	not used
16	same as #1 & #15	not used
17	bad 32 bit integer	not used
18	error converting to 32 bit integer	not used
19	same as #18, except value	not used
20	unknown keyword	not used

24	expected keyword "on" or "off"	not used
25	unexpected value in expression	not used
26	unknown relational operator	not used
27	unbalanced right parenthesis	not used
28	expected keyword "today"	not used

PATTERN Procedures

The WILDCARD Pattern Matching collection contains four procedures used for building and checking for pattern matches. Three of the procedures (**pattern_build**, **pattern_match**, **pattern_fga_match**) provide a low-level approach for integration into your application. The fourth procedure (**check_fga_wildcard**) provides a higher-level approach.

The procedures that start with the string "**pattern_**" are easily callable from either Pascal or C. The other procedure can be called from any Native Mode language.

Conceptually, any of the "**pattern_**" procedures could be called from any Native Mode language. Given that the data structure passed into a "**pattern_**" procedure is fairly complex, you should be aware that calling these types of procedures from either COBOL or SPLash! can be tricky. Conceptually, the **pattern_match** procedure can be used for matching strings of any length. However, it was really designed for matching strings that contain fully-qualified filenames.

Operation

The first thing that must be done to use these procedures is to initialize the PATTERN_TYPE data structures. This is done by calling the **pattern_build** procedure with the appropriate parameters.

Once the PATTERN_TYPE data-structures have been successfully initialized, the **pattern_match** or **pattern_fga_match** procedures can be called repeatedly to check for as many matches as you need. This approach is nice since the **pattern_build** procedure is only called once to set up the pattern (the **check_fga_wildcard** procedure uses both **pattern_build** and **pattern_fga_match**). This approach also makes it possible to initialize several WILDCARD patterns up front and then use them as needed.

check_fga_wildcard Procedure

This procedure is very simple to use. Simply pass in the Wildcard string and the filename string and this procedure will return either true or false. True means the filename was represented by the Wildcard, and False means it wasn't. Additionally, if the return value is negative, it will contain an error number.

check_wildcard

This procedure will simplify the use of the WILDCARD Pattern matching procedures. This procedure is particularly useful if only one (or a few) filename(s) are being tested. Also, this procedure reduces some of the programming necessary to use the pattern matching procedures.

Syntax

int check_wildcard(wildcard, filename);

The Parameter Set is listed next where each parameter is a character array.

Parameter	Name	Type	Comment
1	**wildcard**	character array	Required
2	**filename**	character array	Required

Return Value

Check_wildcard returns a 32-bit integer encoded as follow:

Code	Definition	Error Description
> 0	**:Filename is matched by wildcard**	
= 0	**:Nomatch**	
< 0	**:Error**	
		-1 = Missing 1st "." delimiter
		-2 = Error initializing filename pattern
		-3 = Missing 2nd "." delimiter
		-4 = Error initializing groupname pattern
		-5 = Missing accountname
		-6 = Error initializing accountname pattern
		-7 = Missing filename
		-8 = Missing groupname

Parameters

wildcard Byte array (required). A fully-qualified string ASCII space terminated. This procedure expects that the components of the filename are separated by a dot (.). Also, the buffer containing the string should not contain any characters past the terminating space.

For example: "**@.@.@** ", "**@.pub.sys** ", "**@foo@.???.s#96** "

filename Byte array (required). A NULL (ASCII zero) terminated MPE fully-qualified filename.

Operation

Using this procedure can significantly reduce the amount of programming required to check fully-qualified MPE filenames. This is a stand-alone procedure and is not used in conjunction with any of the other WILDCARD Pattern procedures. See TESTCW.C.LPSTOOLS or TESTPAT.PASCAL.LPSTOOLS.

PATTERN_BUILD

This routine encodes a "pattern" into a special format to be used by the procedures **pattern_match** and **pattern_fga_match**. The "pattern" is returned in its encoded form in the variable of type PATTERN_TYPE. Both C and Pascal header files and example programs have been provided to assist in understanding how to use this procedure.

Syntax

```
int pattern_build(
                wp_pattern_string,
                wp_pattern_length,
                wp_pattern,
                wp_error,
                wp_wildcard_chars,
                wp_options,
                wp_chars_used
                );
```

The Parameter Set is listed next, where each parameter is either an integer, character array, PATTERN_OPTIONS_TYPE, or PATTERN_TYPE:

Parameter	Name	Type	Comment
1	**wp_pattern_string**	character array	Required
2	**wp_pattern_length**	integer — 32-bit signed	Required
3	**wp_pattern**	PATTERN_TYPE	Required
4	**wp_error**	integer — 32-bit signed	Required
5	**wp_wildcard_chars**	character array	Required
6	**wp_options**	PATTERN_OPTIONS_TYPE	Required
7	**wp_chars_used**	integer — 32-bit signed	Required

Return Value

Pattern_build returns a 32-bit integer encoded as follows:

< > 0 : Error See the **wp_error** parameter.

0 : No error

Parameters

wp_pattern_string Byte array by reference (required). It contains the wildcard pattern that is to be initialized. For example, "**@.pub.sys.**"

wp_pattern_length 32-bit integer (required). It contains the byte length of the wildcard pattern stored in **wp_pattern_string**.

wp_pattern PATTERN_TYPE data-structure by reference (required). This parameter is initialized within **pattern_build**, then subsequently passed to **pattern_match** or **pattern_fga_match**. The programmer is only responsible for declaring and passing this parameter.

wp_error 32-integer by reference (required). This parameter will contain an error number if the procedure return value is nonzero encoded as follows:

= 1: Too many firm (constant) characters in **wp_pattern_string** (see the following section for a discussion of firm characters).
= 2 : Negative length
= 3 : Too many parts (firm + wildcard characters) in **wp_pattern_string**
= 4 : Escape, internal error (check **wp_pattern_string**)

wp_wildcard_chars Byte array (optional). It contains the characters that will be used to represent wildcards.

byte 0 : single character wildcard, default = '**?**'
byte 1 : multiple character wildcard, default = '**@**'

byte 2 : single digit wildcard, default = '#'
byte 3 : not used must be an ASCII blank, default = ' '

See the following section for a discussion on setting this parameter.)

wp_options PATTERN_OPTIONS_TYPE data-structure by reference (optional). The parameter is used to select or deselect the following options:

- upshift before matching
- trim leading blanks
- trim trailing blanks

See the following section for a discussion on setting this parameter.

wp_chars_used 32-bit integer by reference (optional). It returns the number of characters used from **wp_pattern_string**. This normally equals the length of the pattern unless an error occurs.

Operation

FIRM CHARACTERS

A **firm character** is a character that is not a wildcard character. WILDCARD patterns are usually constructed of both wildcard and **firm characters**. For example, **"A@.PUB.W???"**. The maximum number of **firm characters** that a pattern can contain is eight (8). Therefore the longest legal pattern is: **"@1@2@3@4@5@6@7@8@"**, or 17 characters long. If the pattern is longer than this, the **wp_error** parameter will be set to three (**pb_err_many_parts**). If more than eight (8) firm characters are found, then the parameter **wp_error** will be set to one (**pb_err_many_firm**).

SETTING WP_OPTIONS

The default WILDCARD Pattern options are:

- Upshift pattern and strings before matching
- Trim (remove) leading spaces from strings before matching
- Trim (remove) trailing spaces from strings before matching

Each of these options are selected by enabling the appropriate entry in the PATTERN_OPTIONS_TYPE data structure. Examples for the PATTERN_OPTIONS_TYPE data structure:

In C:

With the declaration

```
PATTERN_OPTIONS_TYPE wp_options;

    wp_options.upshift          = 1     /* to select (default) */
    wp_options.upshift          = 0     /* to deselect */
    wp_options.trim_leading     = 1     /* to select (default) */
    wp_options.trim_leading     = 0     /* to deselect */
    wp_options.trim_trailing    = 1     /* to select (default) */
    wp_options.trim_trailing    = 0     /* to deselect */
```

In PASCAL:

With the declaration

```
wp_options : PATTERN_OPTIONS_TYPE;

    options     := options + [upshift]          { to select (default) }
    options     := options - [upshift]          { to deselect }
    options     := options + [trim_leading]     { to select (default) }
```

```
options     := options - [trim_leading]     { to deselect }
options     := options + [trim_trailing]    { to select (default) }
options     := options - [trim_trailing]    { to deselect }
```

SETTING WP_WILDCARD_CHARS

The WILDCARD Pattern matching procedures can be programmed to accept any wildcard characters. By default, the WILDCARD Pattern matching procedures use the question mark (?) for any single character wildcard. The "at" sign (@) for any sequence of wildcards, and the "pound" sign (#) for any digit wildcard. MPE and DOS examples follow. Examples for **wp_wildcard_chars** are:

In C:

With the declaration

char pchars[4];

```
strcpy(pchars,"?@#");     /* MPE style wildcards */
strcpy(pchars,"?*#");     /* DOS style wildcards */
```

In Pascal:

With the declaration

pchars : array[1..4]of char;

```
pchars :="?@#";           { MPE style wildcards }
pchars :="?*#";           { DOS style wildcards }
```

For example:

In C:

With the following declarations

```
        /*WILDCARD Pattern variables */
#include "paspat.h"

int
        wp_result,                      /*function returned*/
        wp_error,                       /*error # if wp_result <> 0*/
        wp_buffer_length,               /*strlen of wp_buffer*/
        wp_mismatches,                  /*returned by pattern_fga_match*/
        wp_chars_used;                  /*# of chars used by wp_pattern_build*/

char
        wp_buffer[256]                  /*buffer for passing strings to wp*/
        wp_pchars[4];                   /*ptr to user definable wildcard set*/

PATTERN_TYPE
        wp_pattern;                     /*internal representation*/

PATTERN_OPTIONS_TYPE
        wp_options;                     /*used to select wp options*/

                /*initialization code*/
        strcpy (wp_pchars, "?@#");      /*use default MPE wildcards*/
        wp-options.upshift = 1;         /*upshift before comparing*/
```

```
wp_options.trim_leading = 1;                /*trim leading spaces*/
wp_options.trim_trailing = 1;               /*trim trailing spaces*/
strcpy (wp_buffer, "a##@");                 /*specify a pattern*/
wp_buffer_length = strlen(wp_buffer);
wp_error = wp_chars_used = 0;               /*clear status variables - optional*/
wp_result = pattern_build(
                          wp_buffer,
                          wp_buffer_length,
                          &wp_file_pattern,
                          &wp_error,
                          wp_pchars,
                          wp_options,
                          &wp_chars_used
                          );
if (wp_result != 0)
  /* report error */;

strcpy (wp_buffer, "A69OUT");
wp_buffer_length = 6;
wp_result = pattern_match(
                          wp_buffer,
                          wp_buffer_length,
                          &wp_pattern
                          );
if (wp_result ==0)
   /*report error*/
else
   /*report no match*/;
```

In Pascal:

With the following declarations

```
                    { WILDCARD Pattern variables }
$include 'paspat.dec1.lpstools'$

var
        wp_buffer               : packed array [1. .80] of char;
        wp_error                : integer;
        wp_buffer_length        : integer;
        wp_option               : pattern_options_type;
        wp_pattern              : pattern_type;
        wp_result               : integer;
        wp_chars_used           : integer;
        wp-pchars               : packed array [1. .8] of char;

wp_pchars := "?@#";                                 { use default MPE wildcards }
wp_options := wp_options + [upshift];               { upshift before comparing }
wp_options := wp_options + [trim_leading];          { trim leading spaces }
wp_options := wp_options + [trim_trailing];         { trim trailing spaces }
wp_buffer := "a##@";                                { specify a pattern }
wp_buffer_length := 4;                              { the pattern's length }
wp_chars_used := 0;                                 { clear status variables - optional }
wp_error := 0;
wp_result := pattern_build (
```

```
                                add (wp_buffer),
                                wp_buffer_length,
                                wp_pattrn,
                                wp_error,
                                wp_pchars,
                                wp_options,
                                wp_chars_used
                                );

if wp_result < > 0 then
  {report error}

wp_buffer := "A69OUT";
wp_buffer_length := 6;
wp_result := pattern_match(
                                addr (wp_buffer),
                                wp_buffer_length,
                                wp_pattern
                                );

if wp_result = 0 then
   {report match}
else
   {report no match};
```

PATTERN_FGA_MATCH

This procedure was specifically designed to test a fully-qualified filename against a pattern. Since there are three components to an MPE fully-qualified filename, three patterns must be initialized (with **pattern_match**) before calling this procedure.

Syntax

```
int pattern_fga_match(
                                fga_string,
                                file_pattern,
                                group_pattern,
                                account_pattern,
                                mismatches
                                );
```

The Parameter Set is listed next where each parameter is either an integer, character array, or PATTERN_TYPE:

Parameter	Name	Type	Comment
1	**fga_string**	character array	Required
2	**file_pattern**	PATTERN_TYPE	Required
3	**group_pattern**	PATTERN_TYPE	Required
4	**account_pattern**	PATTERN_TYPE	Required
5	**mismatches**	integer — 32-bit signed	Required

Return Value

Pattern_fga_match returns a 32-bit integer encoded as follows:

= 0	**: MATCH**	
= 1	**: NO MATCH**	
= 2	**: Internal error**	(check input data for correctness)

(See the C and Pascal header files for defines for the return values.)

When a NO MATCH is returned, the variable mismatches can be tested to determine the components of the filename that failed to match.

Parameters

fga_string — Byte array (required). A fully-qualified string ASCII space terminated. This procedure expects that the components of the filename are separated by a dot (.). Also, the buffer containing this string should not contain any characters beyond the terminating space.

file_pattern — PATTERN_TYPE by reference (required). Initialized by a call to **pattern_build** with the desired filename wildcard pattern.

group_pattern — PATTERN_TYPE by reference (required). Initialized by a call to **pattern_build** with the desired groupname wildcard pattern.

account_pattern — PATTERN_TYPE by reference (required). Initialized by a call to **pattern_build** with the desired accountname wildcard pattern.

mismatches — 32-bit integer by reference (required). This variable is used to determine which components of the filename failed. If the return value is zero, then the value of this variable should not be used. If the return value is 1 (NO MATCH), then this variable is encoded as follows:

= 1 : Account name NO MATCH
= 2 : Group name NO MATCH
= 3 : Account and group name NO MATCH
= 4 : File name NO MATCH
= 5 : Account and file name NO MATCH
= 6 : Group and file name NO MATCH
= 7 : Account and group and file name NO MATCH

Operation

As is the case with the **pattern_match** procedure, once the **pattern_build** procedure is used to build the **filename, groupname** and **accountname** patterns it can be called as many times as needed. A typical initialization sequence for this procedure might be:

```
wp_result = pattern_build(filename,...,filename_pattern...)
   ...
wp_result = pattern_build(groupname,...,groupname_pattern...)
   ...
wp_result = pattern_build(accountname,...,accountname_pattern...)
   ...
wp_result = pattern_fga_match(fga_string,...,filename_pattern,)
```

Note: See the file PATTEST.PASCAL.LPSTOOLS.

PATTERN_MATCH

This procedure is used to "test" for pattern matches. For input, it requires an initialized variable of type PATTERN_TYPE (see procedure **pattern_build**), a string to check, and the length of the string.

Syntax

```
int pattern_match(
                  wp_buffer,
                  wp_buffer_length,
                  wp_pattern
                  );
```

The Parameter Set is listed next where each parameter is either an integer, character array, or PATTERN_TYPE:

Parameter	Name	Type	Comment
1	**wp_buffer**	character array	Required
2	**wp_buffer_length**	integer — 32-bit signed	Required
3	**wp_pattern**	PATTERN_TYPE	Required

Return Value

pattern_match returns a 32-bit integer encoded as follows:

= 0	**: MATCH**	
= 1	**: NO MATCH**	
= 2	**: Internal error**	(check input data for correctness)

(See the C and Pascal header files for defines for the return values.)

Parameters

wp_buffer — Byte array (required). It contains the string that is being tested for in the pattern (in **wp_pattern**). *Note*: Since the length is also given, the string doesn't have to be space or NULL terminated.

wp_buffer_length — 32-bit integer (required). The length of the string in **wp_buffer**.

wp_pattern — PATTERN_TYPE data-structure by reference (required). This variable should have been initialized by a call to **pattern_build**.

Operation

Before calling this procedure, call the **pattern_build** procedure to initialize a variable of type PATTERN_TYPE. Then, initialize **wp_buffer** and **wp_buffer_length** and call **pattern_match**. Since **wp_pattern** is initialized external to this procedure, **pattern_match** can be called as many times as needed without reinitializing the **wp_pattern** variable (see the example for the **pattern_build** procedure).

Chapter 23

The XDSMAP Tool

XDSMAP is a library of plug-compatible modules that intercept calls to extra data segment intrinsics and map these calls to mapped files. The result is that calls to DMOVIN and DMOVOUT can be up to 20 times faster than the original intrinsic.

Operation

XDSMAP does not use the Compatibility Mode extra data segments, but rather creates temporary files to store the data segments. All of the extra data segment intrinsics (GETDSEG, ALTDSEG, FREEDSEG, DMOVIN, and DMOVOUT) are intercepted by XDSMAP. Each intrinsic performs the same functional operation as the original. These intrinsics can be called from any native mode program that uses extra data segments. *Note:* Privileged access to extra data segments is not supported.

Each XDSMAP intrinsic functions in a manner consistent with the documented functionality in the Hewlett Packard *Intrinsic Reference Manual*. For the sake of completeness, a brief description of each intrinsic is provided in this chapter. Also, a small test program has been provided so that you may test these intrinsics on your system. Results from tests run on an *HP3000* S/925 are provided for comparison.

Lastly, the XDSMAP intrinsics return all error codes and conditions as documented in the Hewlett-Packard *Intrinsic Reference Manual*.

Capabilities

No special capabilities are required.

Usage

XDSMAP is delivered as a Native Mode object file which can be either linked directly to your program (preferred method where performance issues are of primary concern) or placed in an executable library (NMXL) for run-time binding.

Relocatable Library

```
:link from=myprog.o,xdsmap.o;to=myprog;cap=ia,ba,ds;rl=xdsmap.rl
```

Executable Library

```
:run myprog;xl="XDSMAP.XL"
```

or

```
:link from=myprog.o,xdsmap.o;to=myprog;cap=ia,ba;xl=xdsmap.xl
```

Intrinsic Summary

Listed below is a summary list of XDSMAP intrinsics.

Intrinsics	Description
ALTDSEG	Adjusts size of extra data segment
DMOVIN	Copies data into caller's data area
DMOVOUT	Copies data into extra data segment
FREEDSEG	Deallocates memory
GETDSEG	Allocates extra data segment for process use

Intrinsic Definitions

Following is a detailed definition for each of the XDSMAP intrinsics.

ALTDSEG

This intrinsic is used to adjust the size (up or down) of an extra data segment. The size cannot be increased above the original value allocated by GETDSEG. The calling sequence is as follows:

```
Procedure ALTDSEG  (
        index           :       UInt16  ;       {R26, #bits = 16}
        increment       :       int16   ;       {R25, #bits = 16}
        size            : var   int16   )       {R24, #bits = 32 -> 16}
            { CCE: ok                                           }
            { CCG: ok, but "size" not what you want             }
            { CCL: illegal "index"                              }
            { "size" returns the new size                       }
```

Figure 23.1 - ALTDSEG Intrinsic

DMOVIN

This intrinsic is used to copy data from the extra segment into the caller's data area. The calling sequence is as follows:

```
Procedure DMOVIN  (
        index           :       UInt16  ;       {R26, #bits = 16}
        displacement    :       int16   ;       {R25, #bits = 16}
        number          :       int16   ;       {R24, #bits = 16}
        location        : anyvar record  )      {R23, #bits = 32 -> 65536}
            uncheckable_anyvar
```

Figure 23.2 - DMOVIN Intrinsic

DMOVOUT

This intrinsic is used to copy from the caller's data area into an extra data segment. The calling sequence is as follows:

```
Procedure DMOVOUT  (
        index           :       UInt16  ;       {R26, #bits = 16}
        disp            :       int16   ;       {R25, #bits = 16}
        number          :       int16   ;       {R24, #bits = 16}
        location        : anyvar record  )      {R23, #bits = 32 -> 65536}
            uncheckable_anyvar
```

Figure 23.3 - DMOVOUT Intrinsic

FREEDSEG

This intrinsic is used to deallocate the memory allocated by the GETDSEG intrinsic. The calling sequence is as follows:

```
Procedure FREEDSEG (
        index           :       UInt16  :       {R26, #bits = 16}
        id              :       Uint16  )       {R25, #bits = 16}
```

Figure 23.4 - FREEDSEG Intrinsic

GETDSEG

This intrinsic is used for allocating or acquiring an extra data segment for use by a process. The calling sequence is as follows:

```
Procedure GETDSEG
        index           : var   UInt16  ;       {R26, #bits = 32 -> 16}
        length          : var   int16   ;       {R25, #bits = 32 -> 16}
        id              :       UInt16  )       {R24, #bits = 16}
```

Figure 23.5 - GETDSEG Intrinsic

XDSMAP Examples

Sample test program results. See the file TESTXDS.SPL.LPSTOOLS.

```
:run testxds.pub.lpstools

TESTXDS  (CM)

Length=       1    :  3010
Length=       2    :  2999
Length=       4    :  3003
Length=       8    :  2991
Length=      16    :  3045
Length=      32    :  2962
Length=      64    :  3110
Length=     128    :  3239
Length=     256    :  3397
Length=     512    :  3615
Length=    1024    :  4150
Length=    2048    :  5016
Length=    4096    :  7101
```

Figure 23.6 - Compatibility Mode Output

```
:run testxds.pub.lpstools

TESTXDS   (NM)   without XDSMAP

Length=      1    :  2337
Length=      2    :  2266
Length=      4    :  2353
Length=      8    :  2386
Length=     16    :  2344
Length=     32    :  2531
Length=     64    :  2687
Length=    128    :  2790
Length=    256    :  3081
Length=    512    :  3755
Length=   1024    :  4989
Length=   2048    :  7487
Length=   4096    :  12413
```

Figure 23.7 - Native Mode Output Without XDSMAP

```
:run testxds.pub.lpstools; xl="xdsmap.xl.lpstools"

TESTXDS   (NM)   with XDSMAP

Length=      1    :  110
Length=      2    :  111
Length=      4    :  105
Length=      8    :  111
Length=     16    :  117
Length=     32    :  150
Length=     64    :  151
Length=    128    :  178
Length=    256    :  212
Length=    512    :  271
Length=   1024    :  412
Length=   2048    :  750
Length=   4096    :  1837
```

Figure 23.8 - Native Mode Output With XDSMAP

The sample test program TESTXDS.SPL.LPSTOOLS:

```
TESTXDS  (SPL)  test program for XDSMAP

begin

intrinsic proctim,
      print, getdseg, dmovout, dmovin, quit, dascii, ascii;

integer array
      outbuf       (0  :  39),
      xds'buf      (0  :  4096);

integer
      xds, len, parm, ocnt, icnt, q4  =  q-4;

double
      t;

byte array
      outbuf'             (*)  =  outbuf;

len  :=  20000;
getdseg(xds, len, "aa");
if  <  then
      quit (0);

ocnt  :=  1;
while ocnt  <=  4096  do
      begin
      t  :=  proctim;
      for icnt  :=  1 until 1000 do
            begin
            dmovout (xds, 0, ocnt, xds'buf);
            dmovin  (xds, 0, ocnt, xds'buf);
            end;
      t  :=  proctime  -  t;

      len  :=  move outbuf'  :=  "Length=           ";
      ascii(ocnt,  -10,  outbuf'  (len  -  1) );
      len  :=  len  +  move outbuf'  (len)  :=  " : ";
      len  :=  len  +  dascii (t,  10,  outbuf'  (len) ) ;
      print  (outbuf,  -len,  0) ;
      ocnt  :=  ocnt  *  2 ;
      end;
end.
```

Figure 23.9 - Test Program for XDSMAP

XDSMAP Error Messages

See the *HP Intrinsic Reference Manual* for possible error conditions.

List of Figures

Appendix A

Unsupported Operating Systems

If ***System Managers Toolbox*** or ***Developers 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 MPE/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.

Appendix 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, AVATAR, or any other tool that presents filecode information.

File codes are recorded in the file label and are available to processes 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 mnemonics have particular system-defined meanings. This table defines the 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
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
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

Integer	Mnemonic	Meaning
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
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
1217	DOCDR	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
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

Integer	Mnemonic	Meaning
BASIC Programs		
1246	BKEY	Function Key Definition File for Terminal
1258	PFSTA	Pathflow STATIC File
1259	PFDYN	Pathflow Dynamic File
1270	RFDCA	Revisable Form 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	ROUT1	BRW Intermediate Report File
1434	ROUTD	BRW Dictionary Output
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
1526	MSACK	Man Server Acknowledgment
1527	MSNON	Man Server Non-Delivery Notification
1528	MSTRC	Man Server Trace File
3333		Reserved

Note: Default is file code 0.

Appendix 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

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:

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.)
?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."

Appendix D

Standard Windowing Terms & Features

This section explains the terminology used in describing windows-based tools. Standard features, like function keys, are discussed in the next appendix.

Term	Description
User Input	Monospace typeface (e.g., 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 (e.g., "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 in the menu and highlight it. *Note*: If more than one item starts with the same letter, the cursor will jump to the first menu item it finds with that letter.
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.

Appendix 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 or 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.

Appendix 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:

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-insensitive. 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.

Char	Mnemonic	Description
^E	erase	Erase to end of line. This will erase all of the text from the cursor to the end of the 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).
^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 (e.g., 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.
^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.

Char	Mnemonic	Description
^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

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 if 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).
^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:

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.

Symbol	Explanation
<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 (e.g., ^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 (e.g., 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 TypeAheadEngine (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 (e.g., 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:

Command	Explanation
^W^T	Toggles the TypeAhead Engine through three states: **disabled**, **enabled**, **ignored**.
Leftarrow	The HP26xx left arrow key will move the cursor one space to the left.
Rightarrow	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.

Command	Explanation
^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 start of 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.

Appendix G

Setting Options

The following list covers the standard settings that you would commonly use with a ***System Managers Toolbox*** or ***Developers 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.

Standard Commands

The following commands are common throughout ***LPS-Tools***:

CAPTURE [PARTIAL | FLAT]

The CAPTURE command will generate a hardcopy (or a disk 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". Use the PARTIAL option to capture a portion of screen memory. Use the FLAT option to capture to a disk file. FLAT is the formal file designator. It may be file-equated to another name. For example, if you are running SHOT and you want to perform a screen capture to the file **FOO**, you would type the following statements:

```
:file flat=foo
capture flat
```

:<COMMAND>

A colon (:) followed by an MPE command or UDC name is passed to the HPCICOMMAND intrinsic.

DO [cmd# | relative_cmd # | start_text]

The DO command causes the tool to reuse the selected saved input line without re-editing. If no options follow DO, then the most recent line is reused. If a **cmd#** (e.g., DO 5) is used, then that command is retrieved and reused. If a **relative_cmd#** (e.g. DO -3) is used, then that line is retrieved and reused. A value of -1 means most recent, -2 means second most recent, and so on. If **start_text** is specified, then the most recent command that started with the same text (regardless of case) is reused.

EXIT or "/"

Terminates ***LPS-Tools*** immediately.

HELP

The HELP command (or ?) displays 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 set of characters.

HELP STANDARDS	Displays information about the ***LPS-Tools*** standard interface.
? SE	Displays information about the SET command and any other command beginning with "SE."

Typing "**help ?**" will display the entire HELP file for an ***LPS-Tool***.

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. The "*" option is the default setting.

REDO [cmd# | relative_cmd # | start_text]

The REDO command 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. If a **cmd#** (e.g., REDO 5) is specified, then whatever happens to be on that line in the REDO stack is reused. If a **relative_cmd#** (e.g., REDO -3) is specified, then that line is retrieved and reused. A value of **-1** means most recent, -2 means second most recent, and so on. If **start_text** is specified, then the most recent command that started with the same text (regardless of case) is reused.

Option Syntax

SET [OPTIONS]
RESET [OPTIONS]

In addition to the various SET/RESET options provided by each tool, every tool supports the following options:

BATCH	CAPTUREOK	COPYLP
CRON	CRONOK	CRONPROMPT
EATEMPTY	EATPROMPT	ECHO
MPEOK	PAGING	PSCREENOK
REDO	REDOALL	REDOOK
QEDITOK	UPSHIFT	USEOK

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 provide 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, MEPOK, REDOOK, and USEOK.

Note: Changing the default values for these options is only effective for the duration of the session of the tool you are currently using. Consider using the USE standard command to automatically run a file that will execute a given series of [RE]SET option specifications to help "automate" the process of enabling and disabling the options you want to use most often.

Setting Option Descriptions

Listed below is a detailed description for each of the setting options.

BATCH — 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.

CAPTUREOK — If CAPTUREOK is true, then the "pscreen" (a hard copy of the screen's current contents) can be obtained by entering the CAPTURE command at most prompts. CAPTUREOK can be turned on by entering: SET CAPTUREOK, or turned off by entering: RESET CAPTUREOK.

COPYLP — When COPYLP is true, then a copy of the terminal output (except for input prompts) is sent to LPSLP. COPYLP can be turned on by entering: SET COPYLP, or turned off by entering: RESET COPYLP.

CRON [CROFF] — When CRON is true, pressing **Return** with no other input on the line will cause a tool to reuse 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**

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).

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, or off by entering: RESET CRONPROMPT.

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, or off by entering: RESET EATEMPTY.

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, or turned off by entering: RESET EATPROMPT.

ECHO If ECHO is true, then all input read by the tool input routine is automatically echoed to LPSTOOLSLIST. ECHO is SET/RESET automatically at the start of each tool, and is normally not changed by users.

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, or turned off by entering: RESET MPEOK.

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 approximately every 22 lines). The HELP subsystem *always* temporarily sets paging to true for sessions. PAGING can be turned on by entering: SET PAGING, or turned off by entering: RESET PAGING.

PSCREENOK Synonym for CAPTUREOK.

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 labeled "LISTREDO."

REDO
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.

REDOOK If REDOOK is true, then most tools support the DO, LISTREDO, and REDO commands. REDOOK can be turned on by entering: SET REDOOK, or turned off by entering: RESET REDOOK.

UPSHIFT If UPSHIFT is true, then input will be automatically shifted to uppercase. UPSHIFT can be turned on by entering: SET UPSHIFT, or turned off by entering: RESET UPSHIFT.

USEOK If USEOK is true, then most tools will allow the USE command. USEOK can be turned on by entering: SET USEOK, or turned off by entering: RESET USEOK.

Appendix H

CHRONOS Modes

All mode numbers are in hexadecimal format. Refer to the key provided for an expanded description of format codes used in this list.

Key
MDY = Month/Day/Year
DMY = Day/Month/Year
YMD = Year/Month/Day
WD = Weekday

Mode	Source	Destination	Source Format	Destination Format	Increment Flag	Increment Type
0008	SYSTEM LOCAL	CHRONOS-STAMP	Month/Day/Year	Month/Day/Year	FALSE	
0808	SYSTEM LOCAL	CHRONOS-STAMP	Month/Day/Year	Month/Day/Year	TRUE	DATE
1808	SYSTEM LOCAL	CHRONOS-STAMP	Month/Day/Year	Month/Day/Year	TRUE	TIME
0108	SYSTEM LOCAL	CHRONOS-STAMP	Month/Day/Year	Day/Month/Year	FALSE	
0908	SYSTEM LOCAL	CHRONOS-STAMP	Month/Day/Year	Day/Month/Year	TRUE	DATE
1908	SYSTEM LOCAL	CHRONOS-STAMP	Month/Day/Year	Day/Month/Year	TRUE	TIME
0208	SYSTEM LOCAL	CHRONOS-STAMP	Month/Day/Year	Year/Month/Day	FALSE	
0A08	SYSTEM LOCAL	CHRONOS-STAMP	Month/Day/Year	Year/Month/Day	TRUE	DATE
1A08	SYSTEM LOCAL	CHRONOS-STAMP	Month/Day/Year	Year/Month/Day	TRUE	TIME
0048	SYSTEM LOCAL	CHRONOS-STAMP	Day/Month/Year	Month/Day/Year	FALSE	
0848	SYSTEM LOCAL	CHRONOS-STAMP	Day/Month/Year	Month/Day/Year	TRUE	DATE
1848	SYSTEM LOCAL	CHRONOS-STAMP	Day/Month/Year	Month/Day/Year	TRUE	TIME
0148	SYSTEM LOCAL	CHRONOS-STAMP	Day/Month/Year	Day/Month/Year	FALSE	
0948	SYSTEM LOCAL	CHRONOS-STAMP	Day/Month/Year	Day/Month/Year	TRUE	DATE
1948	SYSTEM LOCAL	CHRONOS-STAMP	Day/Month/Year	Day/Month/Year	TRUE	TIME
0248	SYSTEM LOCAL	CHRONOS-STAMP	Day/Month/Year	Year/Month/Day	FALSE	
0A48	SYSTEM LOCAL	CHRONOS-STAMP	Day/Month/Year	Year/Month/Day	TRUE	DATE
1A48	SYSTEM LOCAL	CHRONOS-STAMP	Day/Month/Year	Year/Month/Day	TRUE	TIME
0088	SYSTEM LOCAL	CHRONOS-STAMP	Year/Month/Day	Month/Day/Year	FALSE	
0888	SYSTEM LOCAL	CHRONOS-STAMP	Year/Month/Day	Month/Day/Year	TRUE	DATE
1888	SYSTEM LOCAL	CHRONOS-STAMP	Year/Month/Day	Month/Day/Year	TRUE	TIME
0188	SYSTEM LOCAL	CHRONOS-STAMP	Year/Month/Day	Day/Month/Year	FALSE	
0988	SYSTEM LOCAL	CHRONOS-STAMP	Year/Month/Day	Day/Month/Year	TRUE	DATE
1988	SYSTEM LOCAL	CHRONOS-STAMP	Year/Month/Day	Day/Month/Year	TRUE	TIME
0288	SYSTEM LOCAL	CHRONOS-STAMP	Year/Month/Day	Year/Month/Day	FALSE	
0A88	SYSTEM LOCAL	CHRONOS-STAMP	Year/Month/Day	Year/Month/Day	TRUE	DATE
1A88	SYSTEM LOCAL	CHRONOS-STAMP	Year/Month/Day	Year/Month/Day	TRUE	TIME
0010	SYSTEM LOCAL	FORMATTED	Month/Day/Year	Month/Day/Year	FALSE	
0810	SYSTEM LOCAL	FORMAT TED	Month/Day/Year	Month/Day/Year	TRUE	DATE
1810	SYSTEM LOCAL	FORMATTED	Month/Day/Year	Month/Day/Year	TRUE	TIME
0110	SYSTEM LOCAL	FORMATTED	Month/Day/Year	Day/Month/Year	FALSE	
0910	SYSTEM LOCAL	FORMATTED	Month/Day/Year	Day/Month/Year	TRUE	DATE
1910	SYSTEM LOCAL	FORMATTED	Month/Day/Year	Day/Month/Year	TRUE	TIME
0210	SYSTEM LOCAL	FORMATTED	Month/Day/Year	Year/Month/Day	FALSE	
0A10	SYSTEM LOCAL	FORMATTED	Month/Day/Year	Year/Month/Day	TRUE	DATE
1A10	SYSTEM LOCAL	FORMATTED	Month/Day/Year	Year/Month/Day	TRUE	TIME
0050	SYSTEM LOCAL	FORMATTED	Day/Month/Year	Month/Day/Year	FALSE	
0850	SYSTEM LOCAL	FORMATTED	Day/Month/Year	Month/Day/Year	TRUE	DATE

Mode	Source	Destination	Source Format	Destination Format	Increment Flag	Increment Type
1850	SYSTEM LOCAL	FORMATTED	Day/Month/Year	Month/Day/Year	TRUE	TIME
0150	SYSTEM LOCAL	FORMATTED	Day/Month/Year	Day/Month/Year	FALSE	
0950	SYSTEM LOCAL	FORMATTED	Day/Month/Year	Day/Month/Year	TRUE	DATE
1950	SYSTEM LOCAL	FORMATTED	Day/Month/Year	Day/Month/Year	TRUE	TIME
0250	SYSTEM LOCAL	FORMATTED	Day/Month/Year	Year/Month/Day	FALSE	
0A50	SYSTEM LOCAL	FORMATTED	Day/Month/Year	Year/Month/Day	TRUE	DATE
1A50	SYSTEM LOCAL	FORMATTED	Day/Month/Year	Year/Month/Day	TRUE	TIME
0090	SYSTEM LOCAL	FORMATTED	Year/Month/Day	Month/Day/Year	FALSE	
0890	SYSTEM LOCAL	FORMATTED	Year/Month/Day	Month/Day/Year	TRUE	DATE
1890	SYSTEM LOCAL	FORMATTED	Year/Month/Day	Month/Day/Year	TRUE	TIME
0190	SYSTEM LOCAL	FORMATTED	Year/Month/Day	Day/Month/Year	FALSE	
0990	SYSTEM LOCAL	FORMATTED	Year/Month/Day	Day/Month/Year	TRUE	DATE
1990	SYSTEM LOCAL	FORMATTED	Year/Month/Day	Day/Month/Year	TRUE	TIME
0290	SYSTEM LOCAL	FORMATTED	Year/Month/Day	Year/Month/Day	FALSE	
0A90	SYSTEM LOCAL	FORMATTED	Year/Month/Day	Year/Month/Day	TRUE	DATE
1A90	SYSTEM LOCAL	FORMATTED	Year/Month/Day	Year/Month/Day	TRUE	TIME
0018	SYSTEM LOCAL	UNFORMATTED	Month/Day/Year	Month/Day/Year	FALSE	
0818	SYSTEM LOCAL	UNFORMATTED	Month/Day/Year	Month/Day/Year	TRUE	DATE
1818	SYSTEM LOCAL	UNFORMATTED	Month/Day/Year	Month/Day/Year	TRUE	TIME
0118	SYSTEM LOCAL	UNFORMATTED	Month/Day/Year	Day/Month/Year	FALSE	
0918	SYSTEM LOCAL	UNFORMATTED	Month/Day/Year	Day/Month/Year	TRUE	DATE
1918	SYSTEM LOCAL	UNFORMATTED	Month/Day/Year	Day/Month/Year	TRUE	TIME
0218	SYSTEM LOCAL	UNFORMATTED	Month/Day/Year	Year/Month/Day	FALSE	
0A18	SYSTEM LOCAL	UNFORMATTED	Month/Day/Year	Year/Month/Day	TRUE	DATE
1A18	SYSTEM LOCAL	UNFORMATTED	Month/Day/Year	Year/Month/Day	TRUE	TIME
0058	SYSTEM LOCAL	UNFORMATTED	Day/Month/Year	Month/Day/Year	FALSE	
0858	SYSTEM LOCAL	UNFORMATTED	Day/Month/Year	Month/Day/Year	TRUE	DATE
1858	SYSTEM LOCAL	UNFORMATTED	Day/Month/Year	Month/Day/Year	TRUE	TIME
0158	SYSTEM LOCAL	UNFORMATTED	Day/Month/Year	Day/Month/Year	FALSE	
0958	SYSTEM LOCAL	UNFORMATTED	Day/Month/Year	Day/Month/Year	TRUE	DATE
1958	SYSTEM LOCAL	UNFORMATTED	Day/Month/Year	Day/Month/Year	TRUE	TIME
0258	SYSTEM LOCAL	UNFORMATTED	Day/Month/Year	Year/Month/Day	FALSE	
0A58	SYSTEM LOCAL	UNFORMATTED	Day/Month/Year	Year/Month/Day	TRUE	DATE
1A58	SYSTEM LOCAL	UNFORMATTED	Day/Month/Year	Year/Month/Day	TRUE	TIME
0098	SYSTEM LOCAL	UNFORMATTED	Year/Month/Day	Month/Day/Year	FALSE	
0898	SYSTEM LOCAL	UNFORMATTED	Year/Month/Day	Month/Day/Year	TRUE	DATE
1898	SYSTEM LOCAL	UNFORMATTED	Year/Month/Day	Month/Day/Year	TRUE	TIME
0198	SYSTEM LOCAL	UNFORMATTED	Year/Month/Day	Day/Month/Year	FALSE	
0998	SYSTEM LOCAL	UNFORMATTED	Year/Month/Day	Day/Month/Year	TRUE	DATE
1998	SYSTEM LOCAL	UNFORMATTED	Year/Month/Day	Day/Month/Year	TRUE	TIME
0298	SYSTEM LOCAL	UNFORMATTED	Year/Month/Day	Year/Month/Day	FALSE	
0A98	SYSTEM LOCAL	UNFORMATTED	Year/Month/Day	Year/Month/Day	TRUE	DATE
1A98	SYSTEM LOCAL	UNFORMATTED	Year/Month/Day	Year/Month/Day	TRUE	TIME
0020	SYSTEM LOCAL	JULIAN	Month/Day/Year	Month/Day/Year	FALSE	
0820	SYSTEM LOCAL	JULIAN	Month/Day/Year	Month/Day/Year	TRUE	DATE
1820	SYSTEM LOCAL	JULIAN	Month/Day/Year	Month/Day/Year	TRUE	TIME
0120	SYSTEM LOCAL	JULIAN	Month/Day/Year	Day/Month/Year	FALSE	
0920	SYSTEM LOCAL	JULIAN	Month/Day/Year	Day/Month/Year	TRUE	DATE
1920	SYSTEM LOCAL	JULIAN	Month/Day/Year	Day/Month/Year	TRUE	TIME

Mode	Source	Destination	Source Format	Destination Format	Increment Flag	Increment Type
0220	SYSTEM LOCAL	JULIAN	Month/Day/Year	Year/Month/Day	FALSE	
0A20	SYSTEM LOCAL	JULIAN	Month/Day/Year	Year/Month/Day	TRUE	DATE
1A20	SYSTEM LOCAL	JULIAN	Month/Day/Year	Year/Month/Day	TRUE	TIME
0060	SYSTEM LOCAL	JULIAN	Day/Month/Year	Month/Day/Year	FALSE	
0860	SYSTEM LOCAL	JULIAN	Day/Month/Year	Month/Day/Year	TRUE	DATE
1860	SYSTEM LOCAL	JULIAN	Day/Month/Year	Month/Day/Year	TRUE	TIME
0160	SYSTEM LOCAL	JULIAN	Day/Month/Year	Day/Month/Year	FALSE	
0960	SYSTEM LOCAL	JULIAN	Day/Month/Year	Day/Month/Year	TRUE	DATE
1960	SYSTEM LOCAL	JULIAN	Day/Month/Year	Day/Month/Year	TRUE	TIME
0260	SYSTEM LOCAL	JULIAN	Day/Month/Year	Year/Month/Day	FALSE	
0A60	SYSTEM LOCAL	JULIAN	Day/Month/Year	Year/Month/Day	TRUE	DATE
1A60	SYSTEM LOCAL	JULIAN	Day/Month/Year	Year/Month/Day	TRUE	TIME
00A0	SYSTEM LOCAL	JULIAN	Year/Month/Day	Month/Day/Year	FALSE	
08A0	SYSTEM LOCAL	JULIAN	Year/Month/Day	Month/Day/Year	TRUE	DATE
18A0	SYSTEM LOCAL	JULIAN	Year/Month/Day	Month/Day/Year	TRUE	TIME
01A0	SYSTEM LOCAL	JULIAN	Year/Month/Day	Day/Month/Year	FALSE	
09A0	SYSTEM LOCAL	JULIAN	Year/Month/Day	Day/Month/Year	TRUE	DATE
19A0	SYSTEM LOCAL	JULIAN	Year/Month/Day	Day/Month/Year	TRUE	TIME
02A0	SYSTEM LOCAL	JULIAN	Year/Month/Day	Year/Month/Day	FALSE	
0AA0	SYSTEM LOCAL	JULIAN	Year/Month/Day	Year/Month/Day	TRUE	DATE
1AA0	SYSTEM LOCAL	JULIAN	Year/Month/Day	Year/Month/Day	TRUE	TIME
0028	SYSTEM LOCAL	CHRONOS-STRING	Month/Day/Year	STRING-WD-MDY	FALSE	
0828	SYSTEM LOCAL	CHRONOS-STRING	Month/Day/Year	STRING-WD-MDY	TRUE	DATE
1828	SYSTEM LOCAL	CHRONOS-STRING	Month/Day/Year	STRING-WD-MDY	TRUE	TIME
0128	SYSTEM LOCAL	CHRONOS-STRING	Month/Day/Year	STRING-WD-DMY	FALSE	
0928	SYSTEM LOCAL	CHRONOS-STRING	Month/Day/Year	STRING-WD-DMY	TRUE	DATE
1928	SYSTEM LOCAL	CHRONOS-STRING	Month/Day/Year	STRING-WD-DMY	TRUE	TIME
0228	SYSTEM LOCAL	CHRONOS-STRING	Month/Day/Year	STRING-MDY	FALSE	
0A28	SYSTEM LOCAL	CHRONOS-STRING	Month/Day/Year	STRING-MDY	TRUE	DATE
1A28	SYSTEM LOCAL	CHRONOS-STRING	Month/Day/Year	STRING-MDY	TRUE	TIME
0328	SYSTEM LOCAL	CHRONOS-STRING	Month/Day/Year	STRING-DMY	FALSE	
0B28	SYSTEM LOCAL	CHRONOS-STRING	Month/Day/Year	STRING-DMY	TRUE	DATE
1B28	SYSTEM LOCAL	CHRONOS-STRING	Month/Day/Year	STRING-DMY	TRUE	TIME
0068	SYSTEM LOCAL	CHRONOS-STRING	Day/Month/Year	STRING-WD-MDY	FALSE	
0868	SYSTEM LOCAL	CHRONOS-STRING	Day/Month/Year	STRING-WD-MDY	TRUE	DATE
1868	SYSTEM LOCAL	CHRONOS-STRING	Day/Month/Year	STRING-WD-MDY	TRUE	TIME
0168	SYSTEM LOCAL	CHRONOS-STRING	Day/Month/Year	STRING-WD-DMY	FALSE	
0968	SYSTEM LOCAL	CHRONOS-STRING	Day/Month/Year	STRING-WD-DMY	TRUE	DATE
1968	SYSTEM LOCAL	CHRONOS-STRING	Day/Month/Year	STRING-WD-DMY	TRUE	TIME
0268	SYSTEM LOCAL	CHRONOS-STRING	Day/Month/Year	STRING-MDY	FALSE	
0A68	SYSTEM LOCAL	CHRONOS-STRING	Day/Month/Year	STRING-MDY	TRUE	DATE
1A68	SYSTEM LOCAL	CHRONOS-STRING	Day/Month/Year	STRING-MDY	TRUE	TIME
0368	SYSTEM LOCAL	CHRONOS-STRING	Day/Month/Year	STRING-DMY	FALSE	
0B68	SYSTEM LOCAL	CHRONOS-STRING	Day/Month/Year	STRING-DMY	TRUE	DATE
1B68	SYSTEM LOCAL	CHRONOS-STRING	Day/Month/Year	STRING-DMY	TRUE	TIME
00A8	SYSTEM LOCAL	CHRONOS-STRING	Year/Month/Day	STRING-WD-MDY	FALSE	
08A8	SYSTEM LOCAL	CHRONOS-STRING	Year/Month/Day	STRING-WD-MDY	TRUE	DATE
18A8	SYSTEM LOCAL	CHRONOS-STRING	Year/Month/Day	STRING-WD-MDY	TRUE	TIME
01A8	SYSTEM LOCAL	CHRONOS-STRING	Year/Month/Day	STRING-WD-DMY	FALSE	
09A8	SYSTEM LOCAL	CHRONOS-STRING	Year/Month/Day	STRING-WD-DMY	TRUE	DATE

Mode	Source	Destination	Source Format	Destination Format	Increment Flag	Increment Type
19A8	SYSTEM LOCAL	CHRONOS-STRING	Year/Month/Day	STRING-WD-DMY	TRUE	TIME
02A8	SYSTEM LOCAL	CHRONOS-STRING	Year/Month/Day	STRING-MDY	FALSE	
0AA8	SYSTEM LOCAL	CHRONOS-STRING	Year/Month/Day	STRING-MDY	TRUE	DATE
1AA8	SYSTEM LOCAL	CHRONOS-STRING	Year/Month/Day	STRING-MDY	TRUE	TIME
03A8	SYSTEM LOCAL	CHRONOS-STRING	Year/Month/Day	STRING-DMY	FALSE	
0BA8	SYSTEM LOCAL	CHRONOS-STRING	Year/Month/Day	STRING-DMY	TRUE	DATE
1BA8	SYSTEM LOCAL	CHRONOS-STRING	Year/Month/Day	STRING-DMY	TRUE	TIME
0009	CHRONOS-STAMP	CHRONOS-STAMP	Month/Day/Year	Month/Day/Year	FALSE	
0809	CHRONOS-STAMP	CHRONOS-STAMP	Month/Day/Year	Month/Day/Year	TRUE	DATE
1809	CHRONOS-STAMP	CHRONOS-STAMP	Month/Day/Year	Month/Day/Year	TRUE	TIME
0109	CHRONOS-STAMP	CHRONOS-STAMP	Month/Day/Year	Day/Month/Year	FALSE	
0909	CHRONOS-STAMP	CHRONOS-STAMP	Month/Day/Year	Day/Month/Year	TRUE	DATE
1909	CHRONOS-STAMP	CHRONOS-STAMP	Month/Day/Year	Day/Month/Year	TRUE	TIME
0209	CHRONOS-STAMP	CHRONOS-STAMP	Month/Day/Year	Year/Month/Day	FALSE	
0A09	CHRONOS-STAMP	CHRONOS-STAMP	Month/Day/Year	Year/Month/Day	TRUE	DATE
1A09	CHRONOS-STAMP	CHRONOS-STAMP	Month/Day/Year	Year/Month/Day	TRUE	TIME
0049	CHRONOS-STAMP	CHRONOS-STAMP	Day/Month/Year	Month/Day/Year	FALSE	
0849	CHRONOS-STAMP	CHRONOS-STAMP	Day/Month/Year	Month/Day/Year	TRUE	DATE
1849	CHRONOS-STAMP	CHRONOS-STAMP	Day/Month/Year	Month/Day/Year	TRUE	TIME
0149	CHRONOS-STAMP	CHRONOS-STAMP	Day/Month/Year	Day/Month/Year	FALSE	
0949	CHRONOS-STAMP	CHRONOS-STAMP	Day/Month/Year	Day/Month/Year	TRUE	DATE
1949	CHRONOS-STAMP	CHRONOS-STAMP	Day/Month/Year	Day/Month/Year	TRUE	TIME
0249	CHRONOS-STAMP	CHRONOS-STAMP	Day/Month/Year	Year/Month/Day	FALSE	
0A49	CHRONOS-STAMP	CHRONOS-STAMP	Day/Month/Year	Year/Month/Day	TRUE	DATE
1A49	CHRONOS-STAMP	CHRONOS-STAMP	Day/Month/Year	Year/Month/Day	TRUE	TIME
0089	CHRONOS-STAMP	CHRONOS-STAMP	Year/Month/Day	Month/Day/Year	FALSE	
0889	CHRONOS-STAMP	CHRONOS-STAMP	Year/Month/Day	Month/Day/Year	TRUE	DATE
1889	CHRONOS-STAMP	CHRONOS-STAMP	Year/Month/Day	Month/Day/Year	TRUE	TIME
0189	CHRONOS-STAMP	CHRONOS-STAMP	Year/Month/Day	Day/Month/Year	FALSE	
0989	CHRONOS-STAMP	CHRONOS-STAMP	Year/Month/Day	Day/Month/Year	TRUE	DATE
1989	CHRONOS-STAMP	CHRONOS-STAMP	Year/Month/Day	Day/Month/Year	TRUE	TIME
0289	CHRONOS-STAMP	CHRONOS-STAMP	Year/Month/Day	Year/Month/Day	FALSE	
0A89	CHRONOS-STAMP	CHRONOS-STAMP	Year/Month/Day	Year/Month/Day	TRUE	DATE
1A89	CHRONOS-STAMP	CHRONOS-STAMP	Year/Month/Day	Year/Month/Day	TRUE	TIME
0011	CHRONOS-STAMP	FORMATTED	Month/Day/Year	Month/Day/Year	FALSE	
0811	CHRONOS-STAMP	FORMATTED	Month/Day/Year	Month/Day/Year	TRUE	DATE
1811	CHRONOS-STAMP	FORMATTED	Month/Day/Year	Month/Day/Year	TRUE	TIME
0111	CHRONOS-STAMP	FORMATTED	Month/Day/Year	Day/Month/Year	FALSE	
0911	CHRONOS-STAMP	FORMATTED	Month/Day/Year	Day/Month/Year	TRUE	DATE
1911	CHRONOS-STAMP	FORMATTED	Month/Day/Year	Day/Month/Year	TRUE	TIME
0211	CHRONOS-STAMP	FORMATTED	Month/Day/Year	Year/Month/Day	FALSE	
0A11	CHRONOS-STAMP	FORMATTED	Month/Day/Year	Year/Month/Day	TRUE	DATE
1A11	CHRONOS-STAMP	FORMATTED	Month/Day/Year	Year/Month/Day	TRUE	TIME
0051	CHRONOS-STAMP	FORMATTED	Day/Month/Year	Month/Day/Year	FALSE	
0851	CHRONOS-STAMP	FORMATTED	Day/Month/Year	Month/Day/Year	TRUE	DATE
1851	CHRONOS-STAMP	FORMATTED	Day/Month/Year	Month/Day/Year	TRUE	TIME
0151	CHRONOS-STAMP	FORMATTED	Day/Month/Year	Day/Month/Year	FALSE	
0951	CHRONOS-STAMP	FORMATTED	Day/Month/Year	Day/Month/Year	TRUE	DATE
1951	CHRONOS-STAMP	FORMATTED	Day/Month/Year	Day/Month/Year	TRUE	TIME

Mode	Source	Destination	Source Format	Destination Format	Increment Flag	Increment Type
0251	CHRONOS-STAMP	FORMATTED	Day/Month/Year	Year/Month/Day	FALSE	
0A51	CHRONOS-STAMP	FORMATTED	Day/Month/Year	Year/Month/Day	TRUE	DATE
1A51	CHRONOS-STAMP	FORMATTED	Day/Month/Year	Year/Month/Day	TRUE	TIME
0091	CHRONOS-STAMP	FORMATTED	Year/Month/Day	Month/Day/Year	FALSE	
0891	CHRONOS-STAMP	FORMATTED	Year/Month/Day	Month/Day/Year	TRUE	DATE
1891	CHRONOS-STAMP	FORMATTED	Year/Month/Day	Month/Day/Year	TRUE	TIME
0191	CHRONOS-STAMP	FORMATTED	Year/Month/Day	Day/Month/Year	FALSE	
0991	CHRONOS-STAMP	FORMATTED	Year/Month/Day	Day/Month/Year	TRUE	DATE
1991	CHRONOS-STAMP	FORMATTED	Year/Month/Day	Day/Month/Year	TRUE	TIME
0291	CHRONOS-STAMP	FORMATTED	Year/Month/Day	Year/Month/Day	FALSE	
0A91	CHRONOS-STAMP	FORMATTED	Year/Month/Day	Year/Month/Day	TRUE	DATE
1A91	CHRONOS-STAMP	FORMATTED	Year/Month/Day	Year/Month/Day	TRUE	TIME
0019	CHRONOS-STAMP	UNFORMATTED	Month/Day/Year	Month/Day/Year	FALSE	
0819	CHRONOS-STAMP	UNFORMATTED	Month/Day/Year	Month/Day/Year	TRUE	DATE
1819	CHRONOS-STAMP	UNFORMATTED	Month/Day/Year	Month/Day/Year	TRUE	TIME
0119	CHRONOS-STAMP	UNFORMATTED	Month/Day/Year	Day/Month/Year	FALSE	
0919	CHRONOS-STAMP	UNFORMATTED	Month/Day/Year	Day/Month/Year	TRUE	DATE
1919	CHRONOS-STAMP	UNFORMATTED	Month/Day/Year	Day/Month/Year	TRUE	TIME
0219	CHRONOS-STAMP	UNFORMATTED	Month/Day/Year	Year/Month/Day	FALSE	
0A19	CHRONOS-STAMP	UNFORMATTED	Month/Day/Year	Year/Month/Day	TRUE	DATE
1A19	CHRONOS-STAMP	UNFORMATTED	Month/Day/Year	Year/Month/Day	TRUE	TIME
0059	CHRONOS-STAMP	UNFORMATTED	Day/Month/Year	Month/Day/Year	FALSE	
0859	CHRONOS-STAMP	UNFORMATTED	Day/Month/Year	Month/Day/Year	TRUE	DATE
1859	CHRONOS-STAMP	UNFORMATTED	Day/Month/Year	Month/Day/Year	TRUE	TIME
0159	CHRONOS-STAMP	UNFORMATTED	Day/Month/Year	Day/Month/Year	FALSE	
0959	CHRONOS-STAMP	UNFORMATTED	Day/Month/Year	Day/Month/Year	TRUE	DATE
1959	CHRONOS-STAMP	UNFORMATTED	Day/Month/Year	Day/Month/Year	TRUE	TIME
0259	CHRONOS-STAMP	UNFORMATTED	Day/Month/Year	Year/Month/Day	FALSE	
0A59	CHRONOS-STAMP	UNFORMATTED	Day/Month/Year	Year/Month/Day	TRUE	DATE
1A59	CHRONOS-STAMP	UNFORMATTED	Day/Month/Year	Year/Month/Day	TRUE	TIME
0099	CHRONOS-STAMP	UNFORMATTED	Year/Month/Day	Month/Day/Year	FALSE	
0899	CHRONOS-STAMP	UNFORMATTED	Year/Month/Day	Month/Day/Year	TRUE	DATE
1899	CHRONOS-STAMP	UNFORMATTED	Year/Month/Day	Month/Day/Year	TRUE	TIME
0199	CHRONOS-STAMP	UNFORMATTED	Year/Month/Day	Day/Month/Year	FALSE	
0999	CHRONOS-STAMP	UNFORMATTED	Year/Month/Day	Day/Month/Year	TRUE	DATE
1999	CHRONOS-STAMP	UNFORMATTED	Year/Month/Day	Day/Month/Year	TRUE	TIME
0299	CHRONOS-STAMP	UNFORMATTED	Year/Month/Day	Year/Month/Day	FALSE	
0A99	CHRONOS-STAMP	UNFORMATTED	Year/Month/Day	Year/Month/Day	TRUE	DATE
1A99	CHRONOS-STAMP	UNFORMATTED	Year/Month/Day	Year/Month/Day	TRUE	TIME
0021	CHRONOS-STAMP	JULIAN	Month/Day/Year	Month/Day/Year	FALSE	
0821	CHRONOS-STAMP	JULIAN	Month/Day/Year	Month/Day/Year	TRUE	DATE
1821	CHRONOS-STAMP	JULIAN	Month/Day/Year	Month/Day/Year	TRUE	TIME
0121	CHRONOS-STAMP	JULIAN	Month/Day/Year	Day/Month/Year	FALSE	
0921	CHRONOS-STAMP	JULIAN	Month/Day/Year	Day/Month/Year	TRUE	DATE
1921	CHRONOS-STAMP	JULIAN	Month/Day/Year	Day/Month/Year	TRUE	TIME
0221	CHRONOS-STAMP	JULIAN	Month/Day/Year	Year/Month/Day	FALSE	
0A21	CHRONOS-STAMP	JULIAN	Month/Day/Year	Year/Month/Day	TRUE	DATE
1A21	CHRONOS-STAMP	JULIAN	Month/Day/Year	Year/Month/Day	TRUE	TIME
0061	CHRONOS-STAMP	JULIAN	Day/Month/Year	Month/Day/Year	FALSE	

Mode	Source	Destination	Source Format	Destination Format	Increment Flag	Increment Type
0861	CHRONOS-STAMP	JULIAN	Day/Month/Year	Month/Day/Year	TRUE	DATE
1861	CHRONOS-STAMP	JULIAN	Day/Month/Year	Month/Day/Year	TRUE	TIME
0161	CHRONOS-STAMP	JULIAN	Day/Month/Year	Day/Month/Year	FALSE	
0961	CHRONOS-STAMP	JULIAN	Day/Month/Year	Day/Month/Year	TRUE	DATE
1961	CHRONOS-STAMP	JULIAN	Day/Month/Year	Day/Month/Year	TRUE	TIME
0261	CHRONOS-STAMP	JULIAN	Day/Month/Year	Year/Month/Day	FALSE	
0A61	CHRONOS-STAMP	JULIAN	Day/Month/Year	Year/Month/Day	TRUE	DATE
1A61	CHRONOS-STAMP	JULIAN	Day/Month/Year	Year/Month/Day	TRUE	TIME
00A1	CHRONOS-STAMP	JULIAN	Year/Month/Day	Month/Day/Year	FALSE	
08A1	CHRONOS-STAMP	JULIAN	Year/Month/Day	Month/Day/Year	TRUE	DATE
18A1	CHRONOS-STAMP	JULIAN	Year/Month/Day	Month/Day/Year	TRUE	TIME
01A1	CHRONOS-STAMP	JULIAN	Year/Month/Day	Day/Month/Year	FALSE	
09A1	CHRONOS-STAMP	JULIAN	Year/ Month/Day	Day/Month/Year	TRUE	DATE
19A1	CHRONOS-STAMP	JULIAN	Year/Month/Day	Day/Month/Year	TRUE	TIME
02A1	CHRONOS-STAMP	JULIAN	Year/Month/Day	Year/Month/Day	FALSE	
0AA1	CHRONOS-STAMP	JULIAN	Year/Month/Day	Year/Month/Day	TRUE	DATE
1AA1	CHRONOS-STAMP	JULIAN	Year/Month/Day	Year/Month/Day	TRUE	TIME
0029	CHRONOS-STAMP	CHRONOS-STRING	Month/Day/Year	STRING-WD-MDY	FALSE	
0829	CHRONOS-STAMP	CHRONOS-STRING	Month/Day/Year	STRING-WD-MDY	TRUE	DATE
1829	CHRONOS-STAMP	CHRONOS-STRING	Month/Day/Year	STRING-WD-MDY	TRUE	TIME
0129	CHRONOS-STAMP	CHRONOS-STRING	Month/Day/Year	STRING-WD-DMY	FALSE	
0929	CHRONOS-STAMP	CHRONOS-STRING	Month/Day/Year	STRING-WD-DMY	TRUE	DATE
1929	CHRONOS-STAMP	CHRONOS-STRING	Month/Day/Year	STRING-WD-DMY	TRUE	TIME
0229	CHRONOS-STAMP	CHRONOS-STRING	Month/Day/Year	STRING-MDY	FALSE	
0A29	CHRONOS-STAMP	CHRONOS-STRING	Month/Day/Year	STRING-MDY	TRUE	DATE
1A29	CHRONOS-STAMP	CHRONOS-STRING	Month/Day/Year	STRING-MDY	TRUE	TIME
0329	CHRONOS-STAMP	CHRONOS-STRING	Month/Day/Year	STRING-DMY	FALSE	
0B29	CHRONOS-STAMP	CHRONOS-STRING	Month/Day/Year	STRING-DMY	TRUE	DATE
1B29	CHRONOS-STAMP	CHRONOS-STRING	Month/Day/Year	STRING-DMY	TRUE	TIME
0069	CHRONOS-STAMP	CHRONOS-STRING	Day/Month/Year	STRING-WD-MDY	FALSE	
0869	CHRONOS-STAMP	CHRONOS-STRING	Day/Month/Year	STRING-WD-MDY	TRUE	DATE
1869	CHRONOS-STAMP	CHRONOS-STRING	Day/Month/Year	STRING-WD-MDY	TRUE	TIME
0169	CHRONOS-STAMP	CHRONOS-STRING	Day/Month/Year	STRING-WD-DMY	FALSE	
0969	CHRONOS-STAMP	CHRONOS-STRING	Day/Month/Year	STRING-WD-DMY	TRUE	DATE
1969	CHRONOS-STAMP	CHRONOS-STRING	Day/Month/Year	STRING-WD-DMY	TRUE	TIME
0269	CHRONOS-STAMP	CHRONOS-STRING	Day/Month/Year	STRING-MDY	FALSE	
0A69	CHRONOS-STAMP	CHRONOS-STRING	Day/Month/Year	STRING-MDY	TRUE	DATE
1A69	CHRONOS-STAMP	CHRONOS-STRING	Day/Month/Year	STRING-MDY	TRUE	TIME
0369	CHRONOS-STAMP	CHRONOS-STRING	Day/Month/Year	STRING-DMY	FALSE	
0B69	CHRONOS-STAMP	CHRONOS-STRING	Day/Month/Year	STRING-DMY	TRUE	DATE
1B69	CHRONOS-STAMP	CHRONOS-STRING	Day/Month/Year	STRING-DMY	TRUE	TIME
00A9	CHRONOS-STAMP	CHRONOS-STRING	Year/Month/Day	STRING-WD-MDY	FALSE	
08A9	CHRONOS-STAMP	CHRONOS-STRING	Year/Month/Day	STRING-WD-MDY	TRUE	DATE
18A9	CHRONOS-STAMP	CHRONOS-STRING	Year/Month/Day	STRING-WD-MDY	TRUE	TIME
01A9	CHRONOS-STAMP	CHRONOS-STRING	Year/Month/Day	STRING-WD-DMY	FALSE	
09A9	CHRONOS-STAMP	CHRONOS-STRING	Year/Month/Day	STRING-WD-DMY	TRUE	DATE
19A9	CHRONOS-STAMP	CHRONOS-STRING	Year/Month/Day	STRING-WD-DMY	TRUE	TIME
02A9	CHRONOS-STAMP	CHRONOS-STRING	Year/Month/Day	STRING-MDY	FALSE	
0AA9	CHRONOS-STAMP	CHRONOS-STRING	Year/Month/Day	STRING-MDY	TRUE	DATE
1AA9	CHRONOS-STAMP	CHRONOS-STRING	Year/Month/Day	STRING-MDY	TRUE	TIME

Mode	Source	Destination	Source Format	Destination Format	Increment Flag	Increment Type
03A9	CHRONOS-STAMP	CHRONOS-STRING	Year/Month/Day	STRING-DMY	FALSE	
0BA9	CHRONOS-STAMP	CHRONOS-STRING	Year/Month/Day	STRING-DMY	TRUE	DATE
1BA9	CHRONOS-STAMP	CHRONOS-STRING	Year/Month/Day	STRING-DMY	TRUE	TIME
000A	FORMATTED	CHRONOS-STAMP	Month/Day/Year	Month/Day/Year	FALSE	
080A	FORMATTED	CHRONOS-STAMP	Month/Day/Year	Month/Day/Year	TRUE	DATE
180A	FORMATTED	CHRONOS-STAMP	Month/Day/Year	Month/Day/Year	TRUE	TIME
010A	FORMATTED	CHRONOS-STAMP	Month/Day/Year	Day/Month/Year	FALSE	
090A	FORMATTED	CHRONOS-STAMP	Month/Day/Year	Day/Month/Year	TRUE	DATE
190A	FORMATTED	CHRONOS-STAMP	Month/Day/Year	Day/Month/Year	TRUE	TIME
020A	FORMATTED	CHRONOS-STAMP	Month/Day/Year	Year/Month/Day	FALSE	
0A0A	FORMATTED	CHRONOS-STAMP	Month/Day/Year	Year/Month/Day	TRUE	DATE
1A0A	FORMATTED	CHRONOS-STAMP	Month/Day/Year	Year/Month/Day	TRUE	TIME
004A	FORMATTED	CHRONOS-STAMP	Day/Month/Year	Month/Day/Year	FALSE	
084A	FORMATTED	CHRONOS-STAMP	Day/Month/Year	Month/Day/Year	TRUE	DATE
184A	FORMATTED	CHRONOS-STAMP	Day/Month/Year	Month/Day/Year	TRUE	TIME
014A	FORMATTED	CHRONOS-STAMP	Day/Month/Year	Day/Month/Year	FALSE	
094A	FORMATTED	CHRONOS-STAMP	Day/Month/Year	Day/Month/Year	TRUE	DATE
194A	FORMATTED	CHRONOS-STAMP	Day/Month/Year	Day/Month/Year	TRUE	TIME
024A	FORMATTED	CHRONOS-STAMP	Day/Month/Year	Year/Month/Day	FALSE	
0A4A	FORMATTED	CHRONOS-STAMP	Day/Month/Year	Year/Month/Day	TRUE	DATE
1A4A	FORMATTED	CHRONOS-STAMP	Day/Month/Year	Year/Month/Day	TRUE	TIME
008A	FORMATTED	CHRONOS-STAMP	Year/Month/Day	Month/Day/Year	FALSE	
088A	FORMATTED	CHRONOS-STAMP	Year/Month/Day	Month/Day/Year	TRUE	DATE
188A	FORMATTED	CHRONOS-STAMP	Year/Month/Day	Month/Day/Year	TRUE	TIME
018A	FORMATTED	CHRONOS-STAMP	Year/Month/Day	Day/Month/Year	FALSE	
098A	FORMATTED	CHRONOS-STAMP	Year/Month/Day	Day/Month/Year	TRUE	DATE
198A	FORMATTED	CHRONOS-STAMP	Year/Month/Day	Day/Month/Year	TRUE	TIME
028A	FORMATTED	CHRONOS-STAMP	Year/Month/Day	Year/Month/Day	FALSE	
0A8A	FORMATTED	CHRONOS-STAMP	Year/Month/Day	Year/Month/Day	TRUE	DATE
1A8A	FORMATTED	CHRONOS-STAMP	Year/Month/Day	Year/Month/Day	TRUE	TIME
0012	FORMATTED	FORMATTED	Month/Day/Year	Month/Day/Year	FALSE	
0812	FORMATTED	FORMATTED	Month/Day/Year	Month/Day/Year	TRUE	DATE
1812	FORMATTED	FORMATTED	Month/Day/Year	Month/Day/Year	TRUE	TIME
0112	FORMATTED	FORMATTED	Month/Day/Year	Day/Month/Year	FALSE	
0912	FORMATTED	FORMATTED	Month/Day/Year	Day/Month/Year	TRUE	DATE
1912	FORMATTED	FORMATTED	Month/Day/Year	Day/Month/Year	TRUE	TIME
0212	FORMATTED	FORMATTED	Month/Day/Year	Year/Month/Day	FALSE	
0A12	FORMATTED	FORMATTED	Month/Day/Year	Year/Month/Day	TRUE	DATE
1A12	FORMATTED	FORMATTED	Month/Day/Year	Year/Month/Day	TRUE	TIME
0052	FORMATTED	FORMATTED	Day/Month/Year	Month/Day/Year	FALSE	
0852	FORMATTED	FORMATTED	Day/Month/Year	Month/Day/Year	TRUE	DATE
1852	FORMATTED	FORMATTED	Day/Month/Year	Month/Day/Year	TRUE	TIME
0152	FORMATTED	FORMATTED	Day/Month/Year	Day/Month/Year	FALSE	
0952	FORMATTED	FORMATTED	Day/Month/Year	Day/Month/Year	TRUE	DATE
1952	FORMATTED	FORMATTED	Day/Month/Year	Day/Month/Year	TRUE	TIME
0252	FORMATTED	FORMATTED	Day/Month/Year	Year/Month/Day	FALSE	
0A52	FORMATTED	FORMATTED	Day/Month/Year	Year/Month/Day	TRUE	DATE
1A52	FORMATTED	FORMATTED	Day/Month/Year	Year/Month/Day	TRUE	TIME
0092	FORMATTED	FORMATTED	Year/Month/Day	Month/Day/Year	FALSE	

Mode	Source	Destination	Source Format	Destination Format	Increment Flag	Increment Type
0892	FORMATTED	FORMATTED	Year/Month/Day	Month/Day/Year	TRUE	DATE
1892	FORMATTED	FORMATTED	Year/Month/Day	Month/Day/Year	TRUE	TIME
0192	FORMATTED	FORMATTED	Year/Month/Day	Day/Month/Year	FALSE	
0992	FORMATTED	FORMATTED	Year/Month/Day	Day/Month/Year	TRUE	DATE
1992	FORMATTED	FORMATTED	Year/Month/Day	Day/Month/Year	TRUE	TIME
0292	FORMATTED	FORMATTED	Year/Month/Day	Year/Month/Day	FALSE	
0A92	FORMATTED	FORMATTED	Year/Month/Day	Year/Month/Day	TRUE	DATE
1A92	FORMATTED	FORMATTED	Year/Month/Day	Year/Month/Day	TRUE	TIME
001A	FORMATTED	UNFORMATTED	Month/Day/Year	Month/Day/Year	FALSE	
081A	FORMATTED	UNFORMATTED	Month/Day/Year	Month/Day/Year	TRUE	DATE
181A	FORMATTED	UNFORMATTED	Month/Day/Year	Month/Day/Year	TRUE	TIME
011A	FORMATTED	UNFORMATTED	Month/Day/Year	Day/Month/Year	FALSE	
091A	FORMATTED	UNFORMATTED	Month/Day/Year	Day/Month/Year	TRUE	DATE
191A	FORMATTED	UNFORMATTED	Month/Day/Year	Day/Month/Year	TRUE	TIME
021A	FORMATTED	UNFORMATTED	Month/Day/Year	Year/Month/Day	FALSE	
0A1A	FORMATTED	UNFORMATTED	Month/Day/Year	Year/Month/Day	TRUE	DATE
1A1A	FORMATTED	UNFORMATTED	Month/Day/Year	Year/Month/Day	TRUE	TIME
005A	FORMATTED	UNFORMATTED	Day/Month/Year	Month/Day/Year	FALSE	
085A	FORMATTED	UNFORMATTED	Day/Month/Year	Month/Day/Year	TRUE	DATE
185A	FORMATTED	UNFORMATTED	Day/Month/Year	Month/Day/Year	TRUE	TIME
015A	FORMATTED	UNFORMATTED	Day/Month/Year	Day/Month/Year	FALSE	
095A	FORMATTED	UNFORMATTED	Day/Month/Year	Day/Month/Year	TRUE	DATE
195A	FORMATTED	UNFORMATTED	Day/Month/Year	Day/Month/Year	TRUE	TIME
025A	FORMATTED	UNFORMATTED	Day/Month/Year	Year/Month/Day	FALSE	
0A5A	FORMATTED	UNFORMATTED	Day/Month/Year	Year/Month/Day	TRUE	DATE
1A5A	FORMATTED	UNFORMATTED	Day/Month/Year	Year/Month/Day	TRUE	TIME
009A	FORMATTED	UNFORMATTED	Year/Month/Day	Month/Day/Year	FALSE	
089A	FORMATTED	UNFORMATTED	Year/Month/Day	Month/Day/Year	TRUE	DATE
189A	FORMATTED	UNFORMATTED	Year/Month/Day	Month/Day/Year	TRUE	TIME
019A	FORMATTED	UNFORMATTED	Year/Month/Day	Day/Month/Year	FALSE	
099A	FORMATTED	UNFORMATTED	Year/Month/Day	Day/Month/Year	TRUE	DATE
199A	FORMATTED	UNFORMATTED	Year/Month/Day	Day/Month /Year	TRUE	TIME
029A	FORMATTED	UNFORMATTED	Year/Month/Day	Year/Month/Day	FALSE	
0A9A	FORMATTED	UNFORMATTED	Year/Month/Day	Year/Month/Day	TRUE	DATE
1A9A	FORMATTED	UNFORMATTED	Year/Month/Day	Year/Month/Day	TRUE	TIME
0022	FORMATTED	JULIAN	Month/Day/Year	Month/Day/Year	FALSE	
0822	FORMATTED	JULIAN	Month/Day/Year	Month/Day/Year	TRUE	DATE
1822	FORMATTED	JULIAN	Month/Day/Year	Month/Day/Year	TRUE	TIME
0122	FORMATTED	JULIAN	Month/Day/Year	Day/Month/Year	FALSE	
0922	FORMATTED	JULIAN	Month/Day/Year	Day/Month/Year	TRUE	DATE
1922	FORMATTED	JULIAN	Month/Day/Year	Day/Month/Year	TRUE	TIME
0222	FORMATTED	JULIAN	Month/Day/Year	Year/Month/Day	FALSE	
0A22	FORMATTED	JULIAN	Month/Day/Year	Year/Month/Day	TRUE	DATE
1A22	FORMATTED	JULIAN	Month/Day/Year	Year/Month/Day	TRUE	TIME
0062	FORMATTED	JULIAN	Day/Month/Year	Month/Day/Year	FALSE	
0862	FORMATTED	JULIAN	Day/Month/Year	Month/Day/Year	TRUE	DATE
1862	FORMATTED	JULIAN	Day/Month/Year	Month/Day/Year	TRUE	TIME
0162	FORMATTED	JULIAN	Day/Month/Year	Day/Month/Year	FALSE	
0962	FORMATTED	JULIAN	Day/Month/Year	Day/Month/Year	TRUE	DATE

Mode	Source	Destination	Source Format	Destination Format	Increment Flag	Increment Type
1962	FORMATTED	JULIAN	Day/Month/Year	Day/Month/Year	TRUE	TIME
0262	FORMATTED	JULIAN	Day/Month/Year	Year/Month/Day	FALSE	
0A62	FORMATTED	JULIAN	Day/Month/Year	Year/Month/Day	TRUE	DATE
1A62	FORMATTED	JULIAN	Day/Month/Year	Year/Month/Day	TRUE	TIME
00A2	FORMATTED	JULIAN	Year/Month/Day	Month/Day/Year	FALSE	
08A2	FORMATTED	JULIAN	Year/Month/Day	Month/Day/Year	TRUE	DATE
18A2	FORMATTED	JULIAN	Year/Month/Day	Month/Day/Year	TRUE	TIME
01A2	FORMATTED	JULIAN	Year/Month/Day	Day/Month/Year	FALSE	
09A2	FORMATTED	JULIAN	Year/Month/Day	Day/Month/Year	TRUE	DATE
19A2	FORMATTED	JULIAN	Year/Month/Day	Day/Month/Year	TRUE	TIME
02A2	FORMATTED	JULIAN	Year/Month/Day	Year/Month/Day	FALSE	
0AA2	FORMATTED	JULIAN	Year/Month/Day	Year/Month/Day	TRUE	DATE
1AA2	FORMATTED	JULIAN	Year/Month/Day	Year/Month/Day	TRUE	TIME
002A	FORMATTED	CHRONOS-STRING	Month/Day/Year	STRING-WD-MDY	FALSE	
082A	FORMATTED	CHRONOS-STRING	Month/Day/Year	STRING-WD-MDY	TRUE	DATE
182A	FORMATTED	CHRONOS-STRING	Month/Day/Year	STRING-WD-MDY	TRUE	TIME
012A	FORMATTED	CHRONOS-STRING	Month/Day/Year	STRING-WD-DMY	FALSE	
092A	FORMATTED	CHRONOS-STRING	Month/Day/Year	STRING-WD-DMY	TRUE	DATE
192A	FORMATTED	CHRONOS-STRING	Month/Day/Year	STRING-WD-DMY	TRUE	TIME
022A	FORMATTED	CHRONOS-STRING	Month/Day/Year	STRING-MDY	FALSE	
0A2A	FORMATTED	CHRONOS-STRING	Month/Day/Year	STRING-MDY	TRUE	DATE
1A2A	FORMATTED	CHRONOS-STRING	Month/Day/Year	STRING-MDY	TRUE	TIME
032A	FORMATTED	CHRONOS-STRING	Month/Day/Year	STRING-DMY	FALSE	
0B2A	FORMATTED	CHRONOS-STRING	Month/Day/Year	STRING-DMY	TRUE	DATE
1B2A	FORMATTED	CHRONOS-STRING	Month/Day/Year	STRING-DMY	TRUE	TIME
006A	FORMATTED	CHRONOS-STRING	Day/Month/Year	STRING-WD-MDY	FALSE	
086A	FORMATTED	CHRONOS-STRING	Day/Month/Year	STRING-WD-MDY	TRUE	DATE
186A	FORMATTED	CHRONOS-STRING	Day/Month/Year	STRING-WD-MDY	TRUE	TIME
016A	FORMATTED	CHRONOS-STRING	Day/Month/Year	STRING-WD-DMY	FALSE	
096A	FORMATTED	CHRONOS-STRING	Day/Month/Year	STRING-WD-DMY	TRUE	DATE
196A	FORMATTED	CHRONOS-STRING	Day/Month/Year	STRING-WD-DMY	TRUE	TIME
026A	FORMATTED	CHRONOS-STRING	Day/Month/Year	STRING-MDY	FALSE	
0A6A	FORMATTED	CHRONOS-STRING	Day/Month/Year	STRING-MDY	TRUE	DATE
1A6A	FORMATTED	CHRONOS-STRING	Day/Month/Year	STRING-MDY	TRUE	TIME
036A	FORMATTED	CHRONOS-STRING	Day/Month/Year	STRING-DMY	FALSE	
0B6A	FORMATTED	CHRONOS-STRING	Day/Month/Year	STRING-DMY	TRUE	DATE
1B6A	FORMATTED	CHRONOS-STRING	Day/Month/Year	STRING-DMY	TRUE	TIME
00AA	FORMATTED	CHRONOS-STRING	Year/Month/Day	STRING-WD-MDY	FALSE	
08AA	FORMATTED	CHRONOS-STRING	Year/Month/Day	STRING-WD-MDY	TRUE	DATE
18AA	FORMATTED	CHRONOS-STRING	Year/Month/Day	STRING-WD-MDY	TRUE	TIME
01AA	FORMATTED	CHRONOS-STRING	Year/Month/Day	STRING-WD-DMY	FALSE	
09AA	FORMATTED	CHRONOS-STRING	Year/Month/Day	STRING-WD-DMY	TRUE	DATE
19AA	FORMATTED	CHRONOS-STRING	Year/Month/Day	STRING-WD-DMY	TRUE	TIME
02AA	FORMATTED	CHRONOS-STRING	Year/Month/Day	STRING-MDY	FALSE	
0AAA	FORMATTED	CHRONOS-STRING	Year/Month/Day	STRING-MDY	TRUE	DATE
1AAA	FORMATTED	CHRONOS-STRING	Year/Month/Day	STRING-MDY	TRUE	TIME
03AA	FORMATTED	CHRONOS-STRING	Year/Month/Day	STRING-DMY	FALSE	
0BAA	FORMATTED	CHRONOS-STRING	Year/Month/Day	STRING-DMY	TRUE	DATE
1BAA	FORMATTED	CHRONOS-STRING	Year/Month/Day	STRING-DMY	TRUE	TIME

Mode	Source	Destination	Source Format	Destination Format	Increment Flag	Increment Type
000B	UNFORMATTED	CHRONOS-STAMP	Month/Day/Year	Month/Day/Year	FALSE	
080B	UNFORMATTED	CHRONOS-STAMP	Month/Day/Year	Month/Day/Year	TRUE	DATE
180B	UNFORMATTED	CHRONOS-STAMP	Month/Day/Year	Month/Day/Year	TRUE	TIME
010B	UNFORMATTED	CHRONOS-STAMP	Month/Day/Year	Day/Month/Year	FALSE	
090B	UNFORMATTED	CHRONOS-STAMP	Month/Day/Year	Day/Month/Year	TRUE	DATE
190B	UNFORMATTED	CHRONOS-STAMP	Month/Day/Year	Day/Month/Year	TRUE	TIME
020B	UNFORMATTED	CHRONOS-STAMP	Month/Day/Year	Year/Month/Day	FALSE	
0A0B	UNFORMATTED	CHRONOS-STAMP	Month/Day/Year	Year/Month/Day	TRUE	DATE
1A0B	UNFORMATTED	CHRONOS-STAMP	Month/Day/Year	Year/Month/Day	TRUE	TIME
004B	UNFORMATTED	CHRONOS-STAMP		Month/Day/Year	FALSE	
084B	UNFORMATTED	CHRONOS-STAMP	Day/Month/Year	Month/Day/Year	TRUE	DATE
184B	UNFORMATTED	CHRONOS-STAMP	Day/Month/Year	Month/Day/Year	TRUE	TIME
014B	UNFORMATTED	CHRONOS-STAMP	Day/Month/Year	Day/Month/Year	FALSE	
094B	UNFORMATTED	CHRONOS-STAMP	Day/Month/Year	Day/Month/Year	TRUE	DATE
194B	UNFORMATTED	CHRONOS-STAMP	Day/Month/Year	Day/Month/Year	TRUE	TIME
024B	UNFORMATTED	CHRONOS-STAMP	Day/Month/Year	Year/Month/Day	FALSE	
0A4B	UNFORMATTED	CHRONOS-STAMP	Day/Month/Year	Year/Month/Day	TRUE	DATE
1A4B	UNFORMATTED	CHRONOS-STAMP	Day/Month/Year	Year/Month/Day	TRUE	TIME
008B	UNFORMATTED	CHRONOS-STAMP	Year/Month/Day	Month/Day/Year	FALSE	
088B	UNFORMATTED	CHRONOS-STAMP	Year/Month/Day	Month/Day/Year	TRUE	DATE
188B	UNFORMATTED	CHRONOS-STAMP	Year/Month/Day	Month/Day/Year	TRUE	TIME
018B	UNFORMATTED	CHRONOS-STAMP	Year/Month/Day	Day/Month/Year	FALSE	
098B	UNFORMATTED	CHRONOS-STAMP	Year/Month/Day	Day/Month/Year	TRUE	DATE
198B	UNFORMATTED	CHRONOS-STAMP	Year/Month/Day	Day/Month/Year	TRUE	TIME
028B	UNFORMATTED	CHRONOS-STAMP	Year/Month/Day	Year/Month/Day	FALSE	
0A8B	UNFORMATTED	CHRONOS-STAMP	Year/Month/Day	Year/Month/Day	TRUE	DATE
1A8B	UNFORMATTED	CHRONOS-STAMP	Year/Month/Day	Year/Month/Day	TRUE	TIME
0013	UNFORMATTED	FORMATTED	Month/Day/Year	Month/Day/Year	FALSE	
0813	UNFORMATTED	FORMATTED	Month/Day/Year	Month/Day/Year	TRUE	DATE
1813	UNFORMATTED	FORMATTED	Month/Day/Year	Month/Day/Year	TRUE	TIME
0113	UNFORMATTED	FORMATTED	Month/Day/Year	Day/Month/Year	FALSE	
0913	UNFORMATTED	FORMATTED	Month/Day/Year	Day/Month/Year	TRUE	DATE
1913	UNFORMATTED	FORMATTED	Month/Day/Year	Day/Month/Year	TRUE	TIME
0213	UNFORMATTED	FORMATTED	Month/Day/Year	Year/Month/Day	FALSE	
0A13	UNFORMATTED	FORMATTED	Month/Day/Year	Year/Month/Day	TRUE	DATE
1A13	UNFORMATTED	FORMATTED	Month/Day/Year	Year/Month/Day	TRUE	TIME
0053	UNFORMATTED	FORMATTED	Day/Month/Year	Month/Day/Year	FALSE	
0853	UNFORMATTED	FORMATTED	Day/Month/Year	Month/Day/Year	TRUE	DATE
1853	UNFORMATTED	FORMATTED	Day/Month/Year	Month/Day/Year	TRUE	TIME
0153	UNFORMATTED	FORMATTED	Day/Month/Year	Day/Month/Year	FALSE	
0953	UNFORMATTED	FORMATTED	Day/Month/Year	Day/Month/Year	TRUE	DATE
1953	UNFORMATTED	FORMATTED	Day/Month/Year	Day/Month/Year	TRUE	TIME
0253	UNFORMATTED	FORMATTED	Day/Month/Year	Year/Month/Day	FALSE	
0A53	UNFORMATTED	FORMATTED	Day/Month/Year	Year/Month/Day	TRUE	DATE
1A53	UNFORMATTED	FORMATTED	Day/Month/Year	Year/Month/Day	TRUE	TIME
0093	UNFORMATTED	FORMATTED	Year/Month/Day	Month/Day/Year	FALSE	
0893	UNFORMATTED	FORMATTED	Year/Month/Day	Month/Day/Year	TRUE	DATE
1893	UNFORMATTED	FORMATTED	Year/Month/Day	Month/Day/Year	TRUE	TIME
0193	UNFORMATTED	FORMATTED	Year/Month/Day	Day/Month/Year	FALSE	
0993	UNFORMATTED	FORMATTED	Year/Month/Day	Day/Month/Year	TRUE	DATE

Mode	Source	Destination	Source Format	Destination Format	Increment Flag	Increment Type
1993	UNFORMATTED	FORMATTED	Year/Month/Day	Day/Month/Year	TRUE	TIME
0293	UNFORMATTED	FORMATTED	Year/Month/Day	Year/Month/Day	FALSE	
0A93	UNFORMATTED	FORMATTED	Year/Month/Day	Year/Month/Day	TRUE	DATE
1A93	UNFORMATTED	FORMATTED	Year/Month/Day	Year/Month/Day	TRUE	TIME
001B	UNFORMATTED	UNFORMATTED	Month/Day/Year	Month/Day/Year	FALSE	
081B	UNFORMATTED	UNFORMATTED	Month/Day/Year	Month/Day/Year	TRUE	DATE
181B	UNFORMATTED	UNFORMATTED	Month/Day/Year	Month/Day/Year	TRUE	TIME
011B	UNFORMATTED	UNFORMATTED	Month/Day/Year	Day/Month/Year	FALSE	
091B	UNFORMATTED	UNFORMATTED	Month/Day/Year	Day/Month/Year	TRUE	DATE
191B	UNFORMATTED	UNFORMATTED	Month/Day/Year	Day/Month/Year	TRUE	TIME
021B	UNFORMATTED	UNFORMATTED	Month/Day/Year	Year/Month/Day	FALSE	
0A1B	UNFORMATTED	UNFORMATTED	Month/Day/Year	Year/Month/Day	TRUE	DATE
1A1B	UNFORMATTED	UNFORMATTED	Month/Day/Year	Year/Month/Day	TRUE	TIME
005B	UNFORMATTED	UNFORMATTED	Day/Month/Year	Month/Day/Year	FALSE	
085B	UNFORMATTED	UNFORMATTED	Day/Month/Year	Month/Day/Year	TRUE	DATE
185B	UNFORMATTED	UNFORMATTED	Day/Month/Year	Month/Day/Year	TRUE	TIME
015B	UNFORMATTED	UNFORMATTED	Day/Month/Year	Day/Month/Year	FALSE	
095B	UNFORMATTED	UNFORMATTED	Day/Month/Year	Day/Month/Year	TRUE	DATE
195B	UNFORMATTED	UNFORMATTED	Day/Month/Year	Day/Month/Year	TRUE	TIME
025B	UNFORMATTED	UNFORMATTED	Day/Month/Year	Year/Month/Day	FALSE	
0A5B	UNFORMATTED	UNFORMATTED	Day/Month/Year	Year/Month/Day	TRUE	DATE
1A5B	UNFORMATTED	UNFORMATTED	Day/Month/Year	Year/Month/Day	TRUE	TIME
009B	UNFORMATTED	UNFORMATTED	Year/Month/Day	Month/Day/Year	FALSE	
089B	UNFORMATTED	UNFORMATTED	Year/Month/Day	Month/Day/Year	TRUE	DATE
189B	UNFORMATTED	UNFORMATTED	Year/Month/Day	Month/Day/Year	TRUE	TIME
019B	UNFORMATTED	UNFORMATTED	Year/Month/Day	Day/Month/Year	FALSE	
099B	UNFORMATTED	UNFORMATTED	Year/Month/Day	Day/Month/Year	TRUE	DATE
199B	UNFORMATTED	UNFORMATTED	Year/Month/Day	Day/Month/Year	TRUE	TIME
029B	UNFORMATTED	UNFORMATTED	Year/Month/Day	Year/Month/Day	FALSE	
0A9B	UNFORMATTED	UNFORMATTED	Year/Month/Day	Year/Month/Day	TRUE	DATE
1A9B	UNFORMATTED	UNFORMATTED	Year/Month/Day	Year/Month/Day	TRUE	TIME
0023	UNFORMATTED	JULIAN	Month/Day/Year	Month/Day/Year	FALSE	
0823	UNFORMATTED	JULIAN	Month/Day/Year	Month/Day/Year	TRUE	DATE
1823	UNFORMATTED	JULIAN	Month/Day/Year	Month/Day/Year	TRUE	TIME
0123	UNFORMATTED	JULIAN	Month/Day/Year	Day/Month/Year	FALSE	
0923	UNFORMATTED	JULIAN	Month/Day/Year	Day/Month/Year	TRUE	DATE
1923	UNFORMATTED	JULIAN	Month/Day/Year	Day/Month/Year	TRUE	TIME
0223	UNFORMATTED	JULIAN	Month/Day/Year	Year/Month/Day	FALSE	
0A23	UNFORMATTED	JULIAN	Month/Day/Year	Year/Month/Day	TRUE	DATE
1A23	UNFORMATTED	JULIAN	Month/Day/Year	Year/Month/Day	TRUE	TIME
0063	UNFORMATTED	JULIAN	Day/Month/Year	Month/Day/Year	FALSE	
0863	UNFORMATTED	JULIAN	Day/Month/Year	Month/Day/Year	TRUE	DATE
1863	UNFORMATTED	JULIAN	Day/Month/Year	Month/Day/Year	TRUE	TIME
0163	UNFORMATTED	JULIAN	Day/Month/Year	Day/Month/Year	FALSE	
0963	UNFORMATTED	JULIAN	Day/Month/Year	Day/Month/Year	TRUE	DATE
1963	UNFORMATTED	JULIAN	Day/Month/Year	Day/Month/Year	TRUE	TIME
0263	UNFORMATTED	JULIAN	Day/Month/Year	Year/Month/Day	FALSE	
0A63	UNFORMATTED	JULIAN	Day/Month/Year	Year/Month/Day	TRUE	DATE
1A63	UNFORMATTED	JULIAN	Day/Month/Year	Year/Month/Day	TRUE	TIME

Mode	Source	Destination	Source Format	Destination Format	Increment Flag	Increment Type
00A3	UNFORMATTED	JULIAN	Year/Month/Day	Month/Day/Year	FALSE	
08A3	UNFORMATTED	JULIAN	Year/Month/Day	Month/Day/Year	TRUE	DATE
18A3	UNFORMATTED	JULIAN	Year/Month/Day	Month/Day/Year	TRUE	TIME
01A3	UNFORMATTED	JULIAN	Year/Month/Day	Day/Month/Year	FALSE	
09A3	UNFORMATTED	JULIAN	Year/Month/Day	Day/Month/Year	TRUE	DATE
19A3	UNFORMATTED	JULIAN	Year/Month/Day	Day/Month/Year	TRUE	TIME
02A3	UNFORMATTED	JULIAN	Year/Month/Day	Year/Month/Day	FALSE	
0AA3	UNFORMATTED	JULIAN	Year/Month/Day	Year/Month/Day	TRUE	DATE
1AA3	UNFORMATTED	JULIAN	Year/Month/Day	Year/Month/Day	TRUE	TIME
002B	UNFORMATTED	CHRONOS-STRING	Month/Day/Year	STRING-WD-MDY	FALSE	
082B	UNFORMATTED	CHRONOS-STRING	Month/Day/Year	STRING-WD-MDY	TRUE	DATE
182B	UNFORMATTED	CHRONOS-STRING	Month/Day/Year	STRING-WD-MDY	TRUE	TIME
012B	UNFORMATTED	CHRONOS-STRING	Month/Day/Year	STRING-WD-DMY	FALSE	
092B	UNFORMATTED	CHRONOS-STRING	Month/Day/Year	STRING-WD-DMY	TRUE	DATE
192B	UNFORMATTED	CHRONOS-STRING	Month/Day/Year	STRING-WD-DMY	TRUE	TIME
022B	UNFORMATTED	CHRONOS-STRING	Month/Day/Year	STRING-MDY	FALSE	
0A2BD	UNFORMATTED	CHRONOS-STRING	Month/Day/Year	STRING-MDY	TRUE	DATE
1A2B	UNFORMATTED	CHRONOS-STRING	Month/Day/Year	STRING-MDY	TRUE	TIME
032B	UNFORMATTED	CHRONOS-STRING	Month/Day/Year	STRING-DMY	FALSE	
0B2B	UNFORMATTED	CHRONOS-STRING	Month/Day/Year	STRING-DMY	TRUE	DATE
1B2B	UNFORMATTED	CHRONOS-STRING	Month/Day/Year	STRING-DMY	TRUE	TIME
006B	UNFORMATTED	CHRONOS-STRING	Day/Month/Year	STRING-WD-MDY	FALSE	
086B	UNFORMATTED	CHRONOS-STRING	Day/Month/Year	STRING-WD-MDY	TRUE	DATE
186B	UNFORMATTED	CHRONOS-STRING	Day/Month/Year	STRING-WD-MDY	TRUE	TIME
016B	UNFORMATTED	CHRONOS-STRING	Day/Month/Year	STRING-WD-DMY	FALSE	
096B	UNFORMATTED	CHRONOS-STRING	Day/Month/Year	STRING-WD-DMY	TRUE	DATE
196B	UNFORMATTED	CHRONOS-STRING	Day/Month/Year	STRING-WD-DMY	TRUE	TIME
026B	UNFORMATTED	CHRONOS-STRING	Day/Month/Year	STRING-MDY	FALSE	
0A6B	UNFORMATTED	CHRONOS-STRING	Day/Month/Year	STRING-MDY	TRUE	DATE
1A6B	UNFORMATTED	CHRONOS-STRING	Day/Month/Year	STRING-MDY	TRUE	TIME
036B	UNFORMATTED	CHRONOS-STRING	Day/Month/Year	STRING-DMY	FALSE	
0B6B	UNFORMATTED	CHRONOS-STRING	Day/Month/Year	STRING-DMY	TRUE	DATE
1B6B	UNFORMATTED	CHRONOS-STRING	Day/Month/Year	STRING-DMY	TRUE	TIME
00AB	UNFORMATTED	CHRONOS-STRING	Year/Month/Day	STRING-WD-MDY	FALSE	
08AB	UNFORMATTED	CHRONOS-STRING	Year/Month/Day	STRING-WD-MDY	TRUE	DATE
18AB	UNFORMATTED	CHRONOS-STRING	Year/Month/Day	STRING-WD-MDY	TRUE	TIME
01AB	UNFORMATTED	CHRONOS-STRING	Year/Month/Day	STRING-WD-DMY	FALSE	
09AB	UNFORMATTED	CHRONOS-STRING	Year/Month/Day	STRING-WD-DMY	TRUE	DATE
19AB	UNFORMATTED	CHRONOS-STRING	Year/Month/Day	STRING-WD-DMY	TRUE	TIME
02AB	UNFORMATTED	CHRONOS-STRING	Year/Month/Day	STRING-MDY	FALSE	
0AAB	UNFORMATTED	CHRONOS-STRING	Year/Month/Day	STRING-MDY	TRUE	DATE
1AAB	UNFORMATTED	CHRONOS-STRING	Year/Month/Day	STRING-MDY	TRUE	TIME
03AB	UNFORMATTED	CHRONOS-STRING	Year/Month/Day	STRING-DMY	FALSE	
0BAB	UNFORMATTED	CHRONOS-STRING	Year/Month/Day	STRING-DMY	TRUE	DATE
1BAB	UNFORMATTED	CHRONOS-STRING	Year/Month/Day	STRING-DMY	TRUE	TIME
000C	JULIAN	CHRONOS-STAMP	Month/Day/Year	Month/Day/Year	FALSE	
080C	JULIAN	CHRONOS-STAMP	Month/Day/Year	Month/Day/Year	TRUE	DATE
180C	JULIAN	CHRONOS-STAMP	Month/Day/Year	Month/Day/Year	TRUE	TIME
010C	JULIAN	CHRONOS-STAMP	Month/Day/Year	Day/Month/Year	FALSE	

Mode	Source	Destination	Source Format	Destination Format	Increment Flag	Increment Type
090C	JULIAN	CHRONOS-STAMP	Month/Day/Year	Day/Month/Year	TRUE	DATE
190C	JULIAN	CHRONOS-STAMP	Month/Day/Year	Day/Month/Year	TRUE	TIME
020C	JULIAN	CHRONOS-STAMP	Month/Day/Year	Year/Month/Day	FALSE	
0A0C	JULIAN	CHRONOS-STAMP	Month/Day/Year	Year/Month/Day	TRUE	DATE
1A0C	JULIAN	CHRONOS-STAMP	Month/Day/Year	Year/Month/Day	TRUE	TIME
004C	JULIAN	CHRONOS-STAMP	Day/Month/Year	Month/Day/Year	FALSE	
084C	JULIAN	CHRONOS-STAMP	Day/Month/Year	Month/Day/Year	TRUE	DATE
184C	JULIAN	CHRONOS-STAMP	Day/Month/Year	Month/Day/Year	TRUE	TIME
014C	JULIAN	CHRONOS-STAMP	Day/Month/Year	Day/Month/Year	FALSE	
094C	JULIAN	CHRONOS-STAMP	Day/Month/Year	Day/Month/Year	TRUE	DATE
194C	JULIAN	CHRONOS-STAMP	Day/Month/Year	Day/Month/Year	TRUE	TIME
024C	JULIAN	CHRONOS-STAMP	Day/Month/Year	Year/Month/Day	FALSE	
0A4C	JULIAN	CHRONOS-STAMP	Day/Month/Year	Year/Month/Day	TRUE	DATE
1A4C	JULIAN	CHRONOS-STAMP	Day/Month/Year	Year/Month/Day	TRUE	TIME
008C	JULIAN	CHRONOS-STAMP	Year/Month/Day	Month/Day/Year	FALSE	
088C	JULIAN	CHRONOS-STAMP	Year/Month/Day	Month/Day/Year	TRUE	DATE
188C	JULIAN	CHRONOS-STAMP	Year/Month/Day	Month/Day/Year	TRUE	TIME
018C	JULIAN	CHRONOS-STAMP	Year/Month/Day	Day/Month/Year	FALSE	
098C	JULIAN	CHRONOS-STAMP	Year/Month/Day	Day/Month/Year	TRUE	DATE
198C	JULIAN	CHRONOS-STAMP	Year/Month/Day	Day/Month/Year	TRUE	TIME
028C	JULIAN	CHRONOS-STAMP	Year/Month/Day	Year/Month/Day	FALSE	
0A8C	JULIAN	CHRONOS-STAMP	Year/Month/Day	Year/Month/Day	TRUE	DATE
1A8C	JULIAN	CHRONOS-STAMP	Year/Month/Day	Year/Month/Day	TRUE	TIME
0014	JULIAN	FORMATTED	Month/Day/Year	Month/Day/Year	FALSE	
0814	JULIAN	FORMATTED	Month/Day/Year	Month/Day/Year	TRUE	DATE
1814	JULIAN	FORMATTED	Month/Day/Year	Month/Day/Year	TRUE	TIME
0114	JULIAN	FORMATTED	Month/Day/Year	Day/Month/Year	FALSE	
0914	JULIAN	FORMATTED	Month/Day/Year	Day/Month/Year	TRUE	DATE
1914	JULIAN	FORMATTED	Month/Day/Year	Day/Month/Year	TRUE	TIME
0214	JULIAN	FORMATTED	Month/Day/Year	Year/Month/Day	FALSE	
0A14	JULIAN	FORMATTED	Month/Day/Year	Year/Month/Day	TRUE	DATE
1A14	JULIAN	FORMATTED	Month/Day/Year	Year/Month/Day	TRUE	TIME
0054	JULIAN	FORMATTED	Day/Month/Year	Month/Day/Year	FALSE	
0854	JULIAN	FORMATTED	Day/Month/Year	Month/Day/Year	TRUE	DATE
1854	JULIAN	FORMATTED	Day/Month/Year	Month/Day/Year	TRUE	TIME
0154	JULIAN	FORMATTED	Day/Month/Year	Day/Month/Year	FALSE	
0954	JULIAN	FORMATTED	Day/Month/Year	Day/Month/Year	TRUE	DATE
1954	JULIAN	FORMATTED	Day/Month/Year	Day/Month/Year	TRUE	TIME
0254	JULIAN	FORMATTED	Day/Month/Year	Year/Month/Day	FALSE	
0A54	JULIAN	FORMATTED	Day/Month/Year	Year/Month/Day	TRUE	DATE
1A54	JULIAN	FORMATTED	Day/Month/Year	Year/Month/Day	TRUE	TIME
0094	JULIAN	FORMATTED	Year/Month/Day	Month/Day/Year	FALSE	
0894	JULIAN	FORMATTED	Year/Month/Day	Month/Day/Year	TRUE	DATE
1894	JULIAN	FORMATTED	Year/Month/Day	Month/ Day/Year	TRUE	TIME
0194	JULIAN	FORMATTED	Year/Month/Day	Day/Month/Year	FALSE	
0994	JULIAN	FORMATTED	Year/Month/Day	Day/Month/Year	TRUE	DATE
1994	JULIAN	FORMATTED	Year/Month/Day	Day/Month/Year	TRUE	TIME
0294	JULIAN	FORMATTED	Year/Month/Day	Year/Month/Day	FALSE	
0A94	JULIAN	FORMATTED	Year/Month/Day	Year/Month/Day	TRUE	DATE
1A94	JULIAN	FORMATTED	Year/Month/Day	Year/Month/Day	TRUE	TIME

Mode	Source	Destination	Source Format	Destination Format	Increment Flag	Increment Type
001C	JULIAN	UNFORMATTED	Month/Day/Year	Month/Day/Year	FALSE	
081C	JULIAN	UNFORMATTED	Month/Day/Year	Month/Day/Year	TRUE	DATE
181C	JULIAN	UNFORMATTED	Month/Day/Year	Month/Day/Year	TRUE	TIME
011C	JULIAN	UNFORMATTED	Month/Day/Year	Day/Month/Year	FALSE	
091C	JULIAN	UNFORMATTED	Month/Day/Year	Day/Month/Year	TRUE	DATE
191C	JULIAN	UNFORMATTED	Month/Day/Year	Day/Month/Year	TRUE	TIME
021C	JULIAN	UNFORMATTED	Month/Day/Year	Year/Month/Day	FALSE	
0A1C	JULIAN	UNFORMATTED	Month/Day/Year	Year/Month/Day	TRUE	DATE
1A1C	JULIAN	UNFORMATTED	Month/Day/Year	Year/Month/Day	TRUE	TIME
005C	JULIAN	UNFORMATTED	Day/Month/Year	Month/Day/Year	FALSE	
085C	JULIAN	UNFORMATTED	Day/Month/Year	Month/Day/Year	TRUE	DATE
185C	JULIAN	UNFORMATTED	Day/Month/Year	Month/Day/Year	TRUE	TIME
015C	JULIAN	UNFORMATTED	Day/Month/Year	Day/Month/Year	FALSE	
095C	JULIAN	UNFORMATTED	Day/Month/Year	Day/Month/Year	TRUE	DATE
195C	JULIAN	UNFORMATTED	Day/Month/Year	Day/Month/Year	TRUE	TIME
025C	JULIAN	UNFORMATTED	Day/Month/Year	Year/Month/Day	FALSE	
0A5C	JULIAN	UNFORMATTED	Day/Month/Year	Year/Month/Day	TRUE	DATE
1A5C	JULIAN	UNFORMATTED	Day/Month/Year	Year/Month/Day	TRUE	TIME
009C	JULIAN	UNFORMATTED	Year/Month/Day	Month/Day/Year	FALSE	
089C	JULIAN	UNFORMATTED	Year/Month/Day	Month/Day/Year	TRUE	DATE
189C	JULIAN	UNFORMATTED	Year/Month/Day	Month/Day/Year	TRUE	TIME
019C	JULIAN	UNFORMATTED	Year/Month/Day	Day/Month/Year	FALSE	
099C	JULIAN	UNFORMATTED	Year/Month/Day	Day/Month/Year	TRUE	DATE
199C	JULIAN	UNFORMATTED	Year/Month/Day	Day/Month/Year	TRUE	TIME
029C	JULIAN	UNFORMATTED	Year/Month/Day	Year/Month/Day	FALSE	
0A9C	JULIAN	UNFORMATTED	Year/Month/Day	Year/Month/Day	TRUE	DATE
1A9C	JULIAN	UNFORMATTED	Year/Month/Day	Year/Month/Day	TRUE	TIME
0024	JULIAN	JULIAN	Month/Day/Year	Month/Day/Year	FALSE	
0824	JULIAN	JULIAN	Month/Day/Year	Month/Day/Year	TRUE	DATE
1824	JULIAN	JULIAN	Month/Day/Year	Month/Day/Year	TRUE	TIME
0124	JULIAN	JULIAN	Month/Day/Year	Day/Month/Year	FALSE	
0924	JULIAN	JULIAN	Month/Day/Year	Day/Month/Year	TRUE	DATE
1924	JULIAN	JULIAN	Month/Day/Year	Day/Month/Year	TRUE	TIME
0224	JULIAN	JULIAN	Month/Day/Year	Year/Month/Day	FALSE	
0A24	JULIAN	JULIAN	Month/Day/Year	Year/Month/Day	TRUE	DATE
1A24	JULIAN	JULIAN	Month/Day/Year	Year/Month/Day	TRUE	TIME
0064	JULIAN	JULIAN	Day/Month/Year	Month/Day/Year	FALSE	
0864	JULIAN	JULIAN	Day/Month/Year	Month/Day/Year	TRUE	DATE
1864	JULIAN	JULIAN	Day/Month/Year	Month/Day/Year	TRUE	TIME
0164	JULIAN	JULIAN	Day/Month/Year	Day/Month/Year	FALSE	
0964	JULIAN	JULIAN	Day/Month/Year	Day/Month/Year	TRUE	DATE
1964	JULIAN	JULIAN	Day/Month/Year	Day/Month/Year	TRUE	TIME
0264	JULIAN	JULIAN	Day/Month/Year	Year/Month/Day	FALSE	
0A64	JULIAN	JULIAN	Day/Month/Year	Year/Month/Day	TRUE	DATE
1A64	JULIAN	JULIAN	Day/Month/Year	Year/Month/Day	TRUE	TIME
00A4	JULIAN	JULIAN	Year/Month/Day	Month/Day/Year	FALSE	
08A4	JULIAN	JULIAN	Year/Month/Day	Month/Day/Year	TRUE	DATE
18A4	JULIAN	JULIAN	Year/Month/Day	Month/Day/Year	TRUE	TIME
01A4	JULIAN	JULIAN	Year/Month/Day	Day/Month/Year	FALSE	

Mode	Source	Destination	Source Format	Destination Format	Increment Flag	Increment Type
09A4	JULIAN	JULIAN	Year/Month/Day	Day/Month/Year	TRUE	DATE
19A4	JULIAN	JULIAN	Year/Month/Day	Day/Month/Year	TRUE	TIME
02A4	JULIAN	JULIAN	Year/Month/Day	Year/Month/Day	FALSE	
0AA4	JULIAN	JULIAN	Year/Month/Day	Year/Month/Day	TRUE	DATE
1AA4	JULIAN	JULIAN	Year/Month/Day	Year/Month/Day	TRUE	TIME
002C	JULIAN	CHRONOS-STRING	Month/Day/Year	STRING-WD-MDY	FALSE	
082C	JULIAN	CHRONOS-STRING	Month/Day/Year	STRING-WD-MDY	TRUE	DATE
182C	JULIAN	CHRONOS-STRING	Month/Day/Year	STRING-WD-MDY	TRUE	TIME
012C	JULIAN	CHRONOS-STRING	Month/Day/Year	STRING-WD-DMY	FALSE	
092C	JULIAN	CHRONOS-STRING	Month/Day/Year	STRING-WD-DMY	TRUE	DATE
192C	JULIAN	CHRONOS-STRING	Month/Day/Year	STRING-WD-DMY	TRUE	TIME
022C	JULIAN	CHRONOS-STRING	Month/Day/Year	STRING-MDY	FALSE	
0A2C	JULIAN	CHRONOS-STRING	Month/Day/Year	STRING-MDY	TRUE	DATE
1A2C	JULIAN	CHRONOS-STRING	Month/Day/Year	STRING-MDY	TRUE	TIME
032C	JULIAN	CHRONOS-STRING	Month/Day/Year	STRING-DMY	FALSE	
0B2C	JULIAN	CHRONOS-STRING	Month/Day/Year	STRING-DMY	TRUE	DATE
1B2C	JULIAN	CHRONOS-STRING	Month/Day/Year	STRING-DMY	TRUE	TIME
006C	JULIAN	CHRONOS-STRING	Day/Month/Year	STRING-WD-MDY	FALSE	
086C	JULIAN	CHRONOS-STRING	Day/Month/Year	STRING-WD-MDY	TRUE	DATE
186C	JULIAN	CHRONOS-STRING	Day/Month/Year	STRING-WD-MDY	TRUE	TIME
016C	JULIAN	CHRONOS-STRING	Day/Month/Year	STRING-WD-DMY	FALSE	
096C	JULIAN	CHRONOS-STRING	Day/Month/Year	STRING-WD-DMY	TRUE	DATE
196C	JULIAN	CHRONOS-STRING	Day/Month/Year	STRING-WD-DMY	TRUE	TIME
026C	JULIAN	CHRONOS-STRING	Day/Month/Year	STRING-MDY	FALSE	
0A6C	JULIAN	CHRONOS-STRING	Day/Month/Year	STRING-MDY	TRUE	DATE
1A6C	JULIAN	CHRONOS-STRING	Day/Month/Year	STRING-MDY	TRUE	TIME
036C	JULIAN	CHRONOS-STRING	Day/Month/Year	STRING-DMY	FALSE	
0B6C	JULIAN	CHRONOS-STRING	Day/Month/Year	STRING-DMY	TRUE	DATE
1B6C	JULIAN	CHRONOS-STRING	Day/Month/Year	STRING-DMY	TRUE	TIME
00AC	JULIAN	CHRONOS-STRING	Year/Month/Day	STRING-WD-MDY	FALSE	
08AC	JULIAN	CHRONOS-STRING	Year/Month/Day	STRING-WD-MDY	TRUE	DATE
18AC	JULIAN	CHRONOS-STRING	Year/Month/Day	STRING-WD-MDY	TRUE	TIME
01AC	JULIAN	CHRONOS-STRING	Year/Month/Day	STRING-WD-DMY	FALSE	
09AC	JULIAN	CHRONOS-STRING	Year/Month/Day	STRING-WD-DMY	TRUE	DATE
19AC	JULIAN	CHRONOS-STRING	Year/Month/Day	STRING-WD-DMY	TRUE	TIME
02AC	JULIAN	CHRONOS-STRING	Year/Month/Day	STRING-MDY	FALSE	
0AAC	JULIAN	CHRONOS-STRING	Year/Month/Day	STRING-MDY	TRUE	DATE
1AAC	JULIAN	CHRONOS-STRING	Year/Month/Day	STRING-MDY	TRUE	TIME
03AC	JULIAN	CHRONOS-STRING	Year/Month/Day	STRING-DMY	FALSE	
0BAC	JULIAN	CHRONOS-STRING	Year/Month/Day	STRING-DMY	TRUE	DATE
1BAC	JULIAN	CHRONOS-STRING	Year/Month/Day	STRING-DMY	TRUE	TIME

Index

G

H

I

J