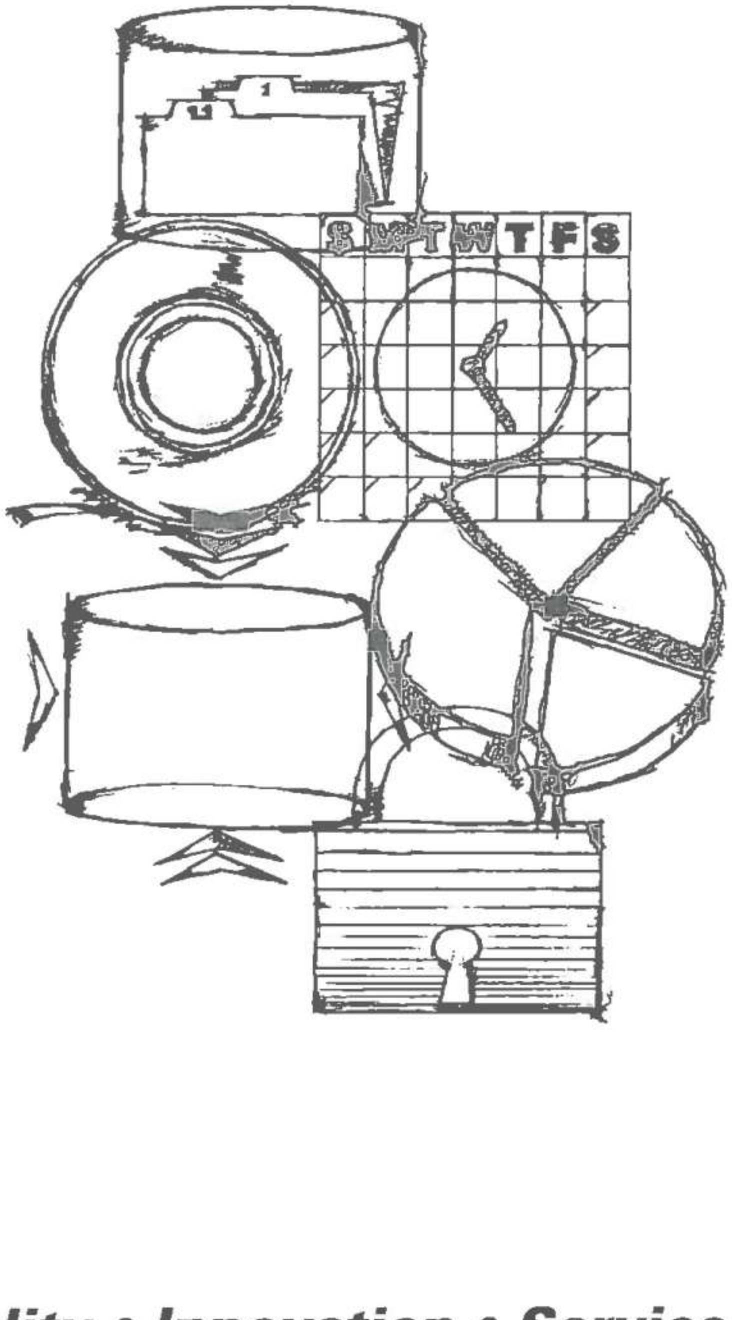
LIBRARIAN/iX™

Reference Guide

Version 4.00
January 1998



OCS

Quality • Innovation • Service

LIBRARIAN/iX Reference Guide
Version 4.00

Table of Contents

List of Figures

List of Tables

Preface

Purpose of this Manual

The *LIBRARIAN/iX Reference Guide* provides detailed information on LIBRARIAN functions, including complete command syntax, and contains reference material for all LIBRARIAN features.

Information in this guide includes:

- Command Syntax and Examples
- Data-entry screens
- Reports
- Datasets

Audience

The *LIBRARIAN/iX Reference Guide* is for operators, programmers, and other users who have a basic knowledge of LIBRARIAN concepts. Knowledge of basic MPE and/or UNIX concepts and terminology is required.

How This Guide Is Organized

The *LIBRARIAN/iX Reference Guide* contains the following chapters:

Chapter 1	"Commands": presents LIBRARIAN commands and their syntax in detail, together with examples.
Chapter 2	"User Fileset Commands": describes the user fileset (FMAINT) commands.
Chapter 3	"Listfile Maintenance Commands": discusses the listfile management (LMAINT) tools available with LIBRARIAN.
Chapter 4	"SHOWLOG Commands": gives the syntax of SHOWLOG commands and examples.
Chapter 5	"Screens": shows LIBRARIAN data entry screens, and explains how to use them.
Chapter 6	"Reports": describes reports available from LIBRARIAN and shows samples.
Chapter 7	"Macro Control Language": discusses the commands you can use in XEQ files to implement macros.
Chapter 8	"MAKE": explains the MAKE facility used to automatically rebuild or recompile changed components of an application.

Chapter 9	"Menu Hierarchy": shows LIBRARIAN's hierarchy of menus and describes menu options.
Chapter 10	"Utility Program": explains how to use the LIBRARIAN Utility program, **LIBUTILP**.
Chapter 11	"Configuration Program": describes the configuration program, **CONFIGP**.
Chapter 12	"Datasets": describes datasets in the LIBDB and LIBLOG databases.

Conventions

We use the following conventions throughout this guide.

COMMANDS All commands appear in bold capital letters. If a command can be abbreviated, the optional portion of the command is enclosed in brackets ([ ]). A blank space *must* separate the command from the parameter list.

KEYWORDS Keywords and parameters (shown in bold capital letters) must be entered exactly as specified.

italics Words or characters in italics represent variables or arguments that you must replace with an actual value. In the following example, you must replace *fileset* with the name of the file you want to copy.

>COPY *fileset*

Italics are also used to introduce new terminology or for emphasis.

punctuation Enter punctuation exactly as shown. (Refer to specific instructions for brackets and braces, below.)

{ } Braces enclose required elements. When there are several elements within braces, you must select one element. In the following example, you must select one of **PROCEDURES**, **PROJECTS**, or **STEPS**.

$$\text{>HELP} \left\{ \begin{array}{l} \text{PROCEDURES} \\ \text{PROJECTS} \\ \text{STEPS} \end{array} \right\}$$

[ ] Brackets enclose optional elements. In the following example, brackets around the letters UPDATE indicate that you do not have to type the entire word.

>AUTO[UPDATE]

If there are several elements, you can select any one or none of them. In the following example you can select **BATCH, CONFIRM** or **MEMO**, or none.

```
>COMPRESS [ filelist ]
       [ ;BATCH ]
       [ ;CONFIRM ]
       [ ;MEMO ]
```

When brackets are used, you cannot enter a value in the inner brackets unless you enter a value (wildcard or literal) in the outer brackets.

...	An ellipsis indicates that the previous bracketed element can be repeated or that elements have been omitted.
&	An ampersand indicates that the command continues on the next line.
MPE	The white flag symbol indicates that the text pertains to LIBRARIAN running under the MPE operating system.
UNIX	The gray flag symbol indicates that the text pertains to LIBRARIAN running under the UNIX operating system.
iX+	The striped flag symbol indicates that the feature being described is only available with LIBRARIAN/iX–Plus.
MPE only	This symbol identifies LIBRARIAN commands that have no equivalent under the UNIX operating system.

File Naming Conventions

In specifying files, LIBRARIAN commands use the following Wildwildcard conventions:

MPE	@	Zero or more alphabetic and/or numeric characters. Used alone, denotes all members of a set.
UNIX	*	Zero or more alphabetic and/or numeric characters. Used alone, denotes all members of a set.
MPE	#	Single numeric character.
	?	Single alphabetic or numeric character.

In addition, a slash /, a single period and slash ./, a double period and slash ../, or a tilde and a slash ~/ immediately preceding a filename indicate a UNIX file.

Related Documentation

Along with this manual, you can refer to the following documentation by OCS.

The *LIBRARIAN/iX Administrator's Guide* contains instructions on how to set up LIBRARIAN for your environment.

The *LIBRARIAN/iX User's Guide* contains instructions on how to perform LIBRARIAN functions. It also includes detailed descriptions of the MAKE facility and the XEQ macro language.

Online help contains the contents of all LIBRARIAN manuals. You can access online help with the **HELP** command or pressing **F1 (Help)** in menu mode.

Client Services

LIBRARIAN is supported by OCS Client Services, which is dedicated to providing timely and accurate information and solutions. For fast, accurate answers, we maintain a telephone hotline that includes emergency after-hours service. You can count on OCS to isolate any problems quickly and provide conscientious support and a fast response.

Operations Control Systems hotline numbers:
Phone (415) 493-4122
FAX (415) 493-3393

Your Comments

We value your comments. As we write, revise, and evaluate our documentation, your opinions are the most important input we receive. Please use the Reader's Comment Form at the end of this guide to tell us what you like and dislike about and of the OCS manuals.

Commands 1

This chapter describes LIBRARIAN operations in detail. Topics in this chapter include:

- Accessing LIBRARIAN from MPE
- Accessing LIBRARIAN from UNIX
- Options and Environment Variables for UNIX
- Menu Mode
- Command Mode
- Online Help
- How to Refer to Files
- Edit Masks
- Batch Operations
- Memos
- LIBRARIAN Commands

Accessing LIBRARIAN from MPE

To access LIBRARIAN from MPE, type:

```
:LIB
```

Note You can save the original user function keys by setting an MPE system variable called LIBSAVEFKEYS to TRUE.

Accessing LIBRARIAN from UNIX

To access LIBRARIAN from UNIX, type:

UNIX(1) ocslib *if path is set,*

otherwise

UNIX(1) $OCSLIBDIR/ocslib

where $OCSLIBDIR is the name of the directory in which the LIBRARIAN client software is installed.

Options and Environment Variables for UNIX

Table 1–1 describes UNIX command line options. Table 1–2 lists the environment variables currently recognized by LIBRARIAN for UNIX.

Table 1-1 Command Line Options

Option	Arguments	Effect
−xterm	None	Translates line drawing characters (between ^N and ^O) to nearest ASCII equivalents. This is automatically detected except when the xterm/hpterm and the actual login shell are running on different machines. This occurs if you are on a workstation and run xterm on another machine.
−hpterm	None	Spawn an hpterm and run ocslib there. Uses fonts hp.8x16 and line.8x16, if available.
−config	None	Modify the ocslib configuration file. This option is re-stricted to **root** users.
−noxlate	None	Overrides the default mode, which tests whether **ocslib** is run from an X-terminal. This is primarily used for the spawned mode to avoid infinitely recursive hpterms.

Table 1-2 Environment Variables

Variable	Value	Effect
XTERM	Any value	Indicates xterm mode, same effect as the *−xterm* command-line option.
DISPLAY	Name of X-Window display server (see X(1))	Used to verify that **−hpterm** mode is running on an X server capable of supporting hpterm.
OCSLIB-SERVER	Name of an MPE LIBRARIAN server	When *ocslib* is invoked, it will use the server name in the config file. This environment variable can be used to override the name of the LIBRARIAN MPE server.
SHELL	Path to user's default shell	Invoked when user selects a shell *escape* or issues shell commands from the LIBRARIAN client.
OCSLIB-DIR	Path to LIBRARIAN client software	Used to determine where the client software is installed.
LIB−PRINT	Options for **lp**	Overrides default for printed output from LIBRARIAN.
LIB-PROMPT	Prompt string	Overrides default LIBRARIAN prompt.
LIB-BATCH	Options for **at**	Overrides default for jobs created by LIBRARIAN and launched with **at**.

Menu Mode

By default LIBRARIAN operates in menu mode to let you select options from a set of menus. The main menu consists of a horizontal menu bar that appears under the "OCS/LIBRARIAN for MPE/iX" (or "UNIX") title bar. This title is displayed at the top of the screen whenever LIBRARIAN is running in menu mode.

To use the menus to issue the commands described in this chapter:

1. Use the right and left arrow keys to highlight the appropriate main menu bar option.
2. Press **ENTER**.
3. Use the up and down arrow keys to highlight the desired pull-down menu option.
4. Press **ENTER** to select the option.
5. If required, enter the appropriate information in the dialog window. You can also specify file revision criteria and options using the appropriate function keys. Press **Go** (**F7**) to proceed with an operation, or **Cancel** to close the dialog (**F8**).
6. Use the **F8** function key to return to the preceding menu.

Refer to Chapter 2, "Getting Started" in the *LIBRARIAN User's Guide* for detailed information about the menus.

Command Mode

As an alternative to menu mode and for batch mode operation, a command line interface is also provided. Available commands are described in this and subsequent chapters.

You can switch between menu and command modes by pressing the **F2** function key. You can also put the command **MENU OFF** in an **AUTOXEQ** file to bypass the menus automatically when running LIBRARIAN.

In addition to LIBRARIAN commands, you can run any of the following:

- MPE or UNIX commands
- UNIX scripts
- MPE UDCs
- MPE user programs

There are two ways to run any of the above, without exiting LIBRARIAN:

1. Type a colon (:) at the LIBRARIAN prompt >, and then press **RETURN** to break to the MPE/UNIX shell.

   For MPE, you can only issue commands that are available in BREAK mode. Type **RESUME** to return to LIBRARIAN from MPE.

   Type **exit** to return to LIBRARIAN from the UNIX shell.

2. Issue a UNIX or MPE command preceded by a colon (:) at the LIBRARIAN prompt >. The colon is optional if it will not be confused with a LIBRARIAN command.

You can configure the LIBRARIAN prompt by setting a system variable called LIBPROMPT to the string you want to use as a prompt. For example:

```
:SETVAR LIBPROMPT "LIB>"

:export LIBPROMPT = "LIB>"
```

You can put the SETVAR statement in the LIBRARIAN UDC file for the **LIB** and **LIBSERV** commands on the MPE/iX server. Then, each time you run LIBRARIAN, the prompt is set automatically.

Background Process on UNIX Clients

UNIX users can run a background process to issue LIBRARIAN commands from a UNIX shell prompt or within a script.

Since the background process maintains its connection to the server, LIBRARIAN is ready to accept requests at any time without the overhead of reconnecting. This capability greatly improves performance.

One use of the background feature is to check out files from MAKE files.

Start a background process by entering the following:

```
$ocslib  -bg
```

To issue LIBRARIAN commands to the background process, use the following syntax:

```
$ocslib  -fg  command
```

where command can be any of the LIBRARIAN commands, except screens and utilities. For security reasons, all requests must be made from the same terminal (or terminal window).

To terminate the background process, enter the exit command, as shown below:

```
$ocslib  -fg  exit
```

If your command includes delimiters or special characters that the shell might interpret, you must use a prefix of "\" with these characters, or enclose the entire command portion in quotations.

Note

The background process inherits its environment from the process you started from, including the working directory and environment variables. However, you can change the current directory for the background process, as shown below:

```
$ocslib  –fg  cd  directory name
```

Online Help

Comprehensive online help is available for all LIBRARIAN commands and their parameters. Use the **F1** function key or the **HELP** command described in this chapter.

You can also select **Help** from the main menu bar to open the help index, review the glossary, or get information about the current version of LIBRARIAN that you are running.

How to Refer to Files

Most LIBRARIAN commands operate on files. In the syntax description for these commands, a *filelist* parameter is included.

The following methods of referring to files are allowed alone or in combination, unless otherwise noted:

Direct reference

- Filename
- Logical fileset
- Listfile (indirect file of filenames)
- Files from the last transaction

Indirect reference

- Revision
- Version and version count
- Generation count
- Secondary location

Implied fileset by project

Implied fileset by step

Multiple file references

Exclusions

Subset selection

- Project
- Tag
- Modification status
- User confirmation
- Tracking status

The following sections describe each of these methods in detail.

Direct References

Filename

You can directly refer to files by name and location. The syntax for MPE is:

MPE

[system:] file [.group [.acct]]

where *file, group,* and *acct* identify the MPE filename. You can use wildcards consistent with MPE **LISTF** conventions. The syntax for UNIX is:

UNIX

[system:] /[path.../] file

where *file* identifies filename, including path preceding the filename. Use wildcards consistent with UNIX conventions (see "Filenaming Conventions" in the preface of this guide).

For both MPE and UNIX, *system* is the name of the system where the file is located. Your current login values are used for omitted elements, except when performing steps, in which case configured values are used.

UNIX

By default, LIBRARIAN treats all path references recursively; That is, all files in subdirectories of any directory specified are included when LIBRARIAN authorizes files. Recursion can be disabled by adding a suffix of a plus sign followed by a zero (+0) to the file reference.

For example, "/usr/devel/d*+0" finds all files starting with the letter "d" in the devel directory without including files that are in subdirectories starting with the letter "d."

You can also control the number of levels of recursion, by adding a suffix of "+n", where "n" is the maximum number of directory levels to traverse.

Logical Fileset

You can directly refer to files in a logical fileset by specifying the fileset name preceded by a percent sign (%). A logical fileset can be a master fileset, user fileset, or project fileset.

When a project fileset is specified on a step (see PERFORM command), LIBRARIAN authorizes the secondary or retained files in the "from" location of the step that are related to the master files in the project fileset and associated with the project. The syntax is:

%fileset

Listfile (Indirect File)

A listfile is a file that contains a list of filenames. You can use listfiles as a way to refer to files in all LIBRARIAN commands. Create listfiles using the LMAINT module of LIBRARIAN or with the editor of your choice. You can directly refer to files in a listfile by specifying the listfile name preceded by an exclamation point (!) or a caret (^). The syntax is:

!filename

Files from the Last Transaction

You can directly refer to files from the last logged transaction by specifying a star (*) or double–star (**). Destination files associated with, or the files processed in the last logged transaction, are authorized. The syntax is:

```
    * (MPE)   or   ** (UNIX)
```

The single or double asterisk refers to the destination files successfully processed in the last transaction (or frozen with the **SET** * command).

Note To use this feature, transaction logging must be enabled on the System Profile (SP) screen.

Indirect References

In a menu mode file dialog, press the F2 function key (Revision Criteria) to specify indirect criteria.

Revision

You can indirectly refer to revisions of master files by specifying the master file(s) and a revision ID. The syntax is:

MPE
UNIX

```
{ [system:] file [.group [.acct ]]  }
{ [system:] /[path.../] file        } ;REV[ISION] =revision-id | ALL
{ %fileset                          }
```

The revision ID is in the format VERSION:VCOUNT [.BRANCH.LEAF...].

You can authorize all revisions of a master file when using the **SET** and **PURGE** commands. To do this, use the **REVISION** parameter with the value of ALL. For example,

```
PURGE MYFILE.PUB.LIBRARY;REVISION=ALL
```

purges all revisions of the files associated with the master file, MYFILE.PUB.LIBRARY.

Version and Version Count

You can indirectly refer to versions of master files by specifying the master file(s) and a version and version count. The syntax is:

```
             { [system:] file [.group [.acct ]] }
versionid OF { [system:] /[path.../] file       } [ ;VCOUNT=versioncount ]
             { %fileset                         }
```

Versionid is the identifier of a version. If the application for the version is ambiguous, LIBRARIAN prompts for it.

VCOUNT identifies the files with a version count equal to **VCOUNT** (the number of times the master file has been revised since the base version was created). Default: 0 (baseline version).

A **VCOUNT** value of **LAST** causes LIBRARIAN to operate on the last revision of a file within a version.

For example, the following command copies the latest revision of each file in the 1.0 version to the V100 area:

```
>COPY 1.0 OF %FINANCE TO =.=.V100;VCOUNT=LAST;OLDNAME
```

```
>COPY 1.0 OF %FINANCE TO /apps/gl/v100/(3,*);VCOUNT=LAST;OLDNAME
```

A **VCOUNT** value of **LASTNOT0** causes LIBRARIAN to operate on the last revision of a file within a retained version that is not a base revision. (e.g. to create a patch tape.)

For example, the following command distributes only those files that have changed since the base version was distributed:

```
>COPY RE.1.0 OF %MYFILES TO =.=.RELEASE;VCOUNT=LASTNOT0
```

```
>COPY RE.1.0 OF %MYFILES TO /apps/gl/release/(3,*);VCOUNT=LASTNOT0
```

Generation Count

You can indirectly refer to generations of a master file by specifying a master file(s) and a generation count. The syntax is:

```
{ [system:] file [.group [.acct ]] }
{ [system:] /[path.../] file       }  ;GCOUNT = [ - ] gcount
{ %fileset                         }
```

The **GCOUNT** parameter directs LIBRARIAN to operate on files with the specified generation count (total number of times the master file has been replaced since its creation). This value can be either a positive or a negative value.

A negative value describes the generation relative to the current generation. For example, GCOUNT = -2 specifies files two generations prior to the current one.

Secondary Location

You can indirectly refer to secondary files by specifying the master filename(s) and the general location to search for associated secondaries. The syntax for MPE is:

$$\left\{ \begin{array}{ll} \textit{\%fileset} & \text{AT} \\ [\textit{system:}]\textit{file} \; [.\textit{group} \; [.\textit{acct}\,]\,] & \text{AT} \end{array} \right\} [\textit{system:}]\textit{file} \; [.\textit{group} \; [.\textit{acct}\,]]$$

When using this syntax, LIBRARIAN operates on secondaries of the specified master files found in the specified secondary (AT) location. For example:

```
%AP-FILES AT @:@.@.@
```

refers to all secondary copies of %AP-FILES.

Alternatively, the syntax for UNIX is:

$$\left\{ \begin{array}{ll} \textit{\%fileset} & \text{AT} \\ [\textit{system:}] \; /[\textit{path...}/] \; \textit{file} & \text{AT} \end{array} \right\} [\textit{system:}] \; /[\textit{path...}/] \; \textit{file}$$

When using this syntax, LIBRARIAN operates on secondaries of the specified master files found in the specified secondary (AT) location. For example:

```
%AP-FILES AT *:/*
```

refers to all secondary copies of %AP-FILES.

Note In a menu mode dialog, enter the "AT" syntax directly in the filelist field, as you would in command mode.

Implied Reference by Project

You can imply the files associated with a project when performing a step by specifying the project name, rather than files. The syntax is:

>step.project

Alternatively, you can omit the project name and select your project from the project menu when projects are defined. In menu mode, this is the only alternative.

Implied Reference by Step

If you do not specify any files when performing a step, the step fileset (as defined on the Step (ST) screen) is used. For example:

>step

If projects are being used, you are presented a menu of projects. By selecting a project, you imply the project fileset when no files are specified.

Multiple File References

You can refer to multiple files combining any of the methods described above. Use commas to create a list of file specifications. The syntax is:

filelist [, *filelist* [, ...]]

Each filelist is processed by the LIBRARIAN program in a single transaction.

Exclusions

This method designates files to be excluded from the operation. The syntax is:

MPE

UNIX

$$\left\{\begin{array}{l} -\ [\textit{system:}]\ \textit{file}\ [\textit{.group}\ [\textit{.acct}\ ]] \\ -\ [\textit{system:}]\ /[\textit{path...}/]\ \textit{file} \\ -\ \%\textit{fileset} \end{array}\right\}$$

When specifying multiple filelists, specify the exclusion(s) last. Exclusion(s) must be direct references, with or without wildcards. Use commas to separate exclusions.

Subset Selection

Project

Subset selection by project selects only files associated with a particular project. This parameter must follow all file references, including destination locations, if specified. **PROJECT** is valid for all commands. The syntax is:

filelist; PROJECT=*proj*

If you use a step to copy files in read–mode (e.g., move–to–production), LIBRARIAN automatically copies the appropriate revisions of the files associated with the project that you specify. However, if you do not use a step for file distribution (e.g., **COPY**), then use the project fileset as well as the **PROJECT** parameter.

Tag

Subset selection by tag selects only files that were assigned a specific tag with the **SET TAG** command. This parameter must follow all file references including destination locations, if specified. The syntax is:

*filelist;*TAG=*tagid*

Modification Status

Subset selection by modification status selects files based upon whether or not they have been modified since LIBRARIAN created them. Use the **MODIFIED** or **UNMODIFIED** parameters to select only those files modified or not modified since they were last copied or moved by LIBRARIAN. The current timestamp in the file label is compared with the timestamp in the LIBRARIAN database. For example:

```
filelist;MODIFIED
```

User Confirmation

Subset selection by user confirmation has LIBRARIAN prompt for confirmation of each authorized file prior to processing. Use the **CONFIRM** parameter to request prompting. Files not confirmed are excluded from the operation. For example:

```
filelist;CONFIRM
```

Tracking Status

Subset selection by tracking status lets you select files being tracked by LIBRARIAN, excluding those not being tracked. This applies only to ad hoc commands, such as **COPY** and **PURGE**. To include only Files, untracked files, prefix these commands with "X". For example:

```
filelist;TRACKED
```

Edit Masks

Edit masks are used to determine the correct destination given a specific source name. The masks are either defined in the destination of a step, or specified when performing the step or other file movement command.

Edit masks are also used to specify refinements for step destinations, and to translate pending production secondary filenames into pending master filenames. This enables LIBRARIAN to create pending master records automatically.

There is a one–to–one correspondence between elements of a fully qualified filename. (For MPE, elements are system, file, group, and account. For UNIX, elements are system, path components, and filename.) For each element, the mask can result in carrying forward the element, replacing the element, or editing the element:

- Elements are carried forward using the equal sign (=), or the at sign (@) in a step definition, if the user can override the element.
- Elements are replaced by using a string literal.
- Elements can be edited using a combination of equal (=), at (@), question (?), minus (–) sign, and literals, as described below.

Table 1–3 describes the valid edit mask characters for any element of an MPE or UNIX filename, along with their descriptions and examples.

Table 1-3. Edit Mask Symbols and Descriptions

Edit Mask Character	Description
At sign/Star @ *	Copies original value into edited version. Typically preceded and/or followed by other characters. For example, when edited with the edit mask of ABC@XYZ, the value of FRED results in a value of ABCFREDXYZ. For MPE filenames, the result is truncated to eight characters (ABCFREDX).
Question mark ?	Copies the character at this position into the resulting string. For example, the mask ?? applied to the string FRED results in the string FR. The question mark can be combined with literal characters such as ??X, which would result in the string, FRX, or X??, resulting in XRE. It can also be combined with the minus sign (–).
Minus sign -	Indicates that the original character in that position should not be included in the edited result. For example, the mask –? applied to the string FRED results in R. Alternatively, the mask -=- results in the string RE. An additional feature is the use of "–" in conjunction with "@", which strips characters from the beginning and/or end of the original element before adding other characters to the beginning or end. For example, PRTA100 edited with – – –@S results in A100S, deleting the first three characters before adding the S:. Note that - - –=S would result in A10S, replacing the last character.
Equal sign =	Copies all remaining characters after the minus sign, question mark, and literal characters have been evaluated. For example, a mask of =X with the original string FRED results in the string FREX. The mask =Q? with the initial string FRED results in FRQD.

Edit Masks for UNIX Pathnames

UNIX

To carry forward, edit, or replace an element that is at the same level in both the source and destination filenames, follow the rules described above.

Because UNIX pathnames can have varying numbers of path elements (directories), you can edit (or skip) components at varying levels in the source filename using the following construct:

/(x [– y]) [*edit–mask*]

where x and y represent the desired range of components from the source pathname. x and y are numbers from 1 to *, where * is the last directory element of the pathname. If you want a specific element, omit y which is optional.

The optional edit mask is applied to each element in the range (do not include the brackets).

For example, the mask /(1 –2)/devel/!USERID/(4– *)/= applied to the filename /usr/usr2/master/screens/abc results in the filename /usr/usr2/devel/milind/screens/abc.

You can also use the following wildcards in place of x or y:

- ~ number of levels in home directory path
- . number of levels in the current working directory path
- .. one less than the number of levels in the current working directory path

You can use curly braces, i.e., { x [– y] }, to indicate mapping from the master file name rather than the current secondary file name. For examples, see Chapter 3, "File Transactions" in the *LIBRARIAN/iX User's Guide*, and Chapter 4, "File Movement Rules" in the *LIBRARIAN/iX Administrator's Guide.*

Edit Masks for Group and Accounts

You can specify an edit mask that refers to a different element (i.e., file, group or account). To do this, use the following syntax in your edit mask:

$$\left(\left\{\begin{array}{c}\mathrm{F}\\ \mathrm{G}\\ \mathrm{A}\end{array}\right\}\textit{start:length}\right)$$

where F is for filename, G is for group name, and A is for account name. *Start* is the starting position, and *length* is the number of characters to be used.

The example below shows an edit mask that creates a group name using the first three characters of the filename:

Source:	ABC100S.PSOURCE
Edit:	PRG???.P(F:1:3)OBJ
Destination:	PRG100.PABCOBJ

Batch Operations

When you use the **BATCH** parameter under MPE, LIBRARIAN prompts you for a **!JOB** command and MPE **:STREAM** options. All job parameters, such as **INPRI**, **PRI**, **OUTCLASS**, **STREAM**, **AT**, and **DATE** are supported.

If the OCS-ENABLED flag is set to Y on the System Profile (SP) screen, the EXPRESS SUBMIT facility is invoked enabling you to schedule the transaction. If the flag is set to N, you are prompted to supply the login values and MPE **:STREAM** options for the transaction job before it is streamed.

When you use the **BATCH** parameter under UNIX, LIBRARIAN launches jobs using the UNIX **at** command. LIBRARIAN prompts you for **at** options, or you can set the environment variable, **LIBBATCH**, to provide these options.

Memos

A memo is text that describes the current transaction in the audit trail. Use the **MEMO** parameter to include a memo. To create one–line memos up to 72 characters, enter the memo text on the command line as a value for the **MEMO** parameter (i.e., **MEMO** = memo–text). For multi–line memos, do not specify the text on the command line and LIBRARIAN will invoke the configured editor; by default, **EDITOR/3000** on MPE, or **vi** on UNIX. You can review and modify the text through SHOWLOG.

LIBRARIAN Commands

There are five types of LIBRARIAN operations:

Steps	Execute defined file movements or approvals. Step operation is described under the **PERFORM** command.
Ad hoc file operations	Execute a full range of file transactions. File system security is enforced for files not tracked by LIBRARIAN. (X capability can be assigned to override this security).
Information displays	Show file data, source code comparisons, and transaction logs.
Other activities	User identification, utilities, and work environment setup.
X commands	Execute ad hoc file operations only on files not tracked by LIBRARIAN, by prefixing the command with X in command mode.

Note In all cases, **XSCAN, XPRINT, XLCOMPARE,** and **XSCOMPARE** operate on both tracked and untracked files.

LIBRARIAN permits access to the five types of commands at three levels, as summarized in Table 1–4.

Table 1–4. Command Access Restrictions

Command Type	LIBRARIAN Manager	Application Manager	General Users
PERFORM (step)	Files covered by a defined step	Files covered by a defined step for own applications	Must have specific authorization
Ad hoc file operations	Any file	Any file in own applications	Files they own
Operations on untracked files	Any file	MPE security enforced	MPE security enforced
Information displays	Any file	Any file	Any file
Other activities	All activities	Restricted	Restricted

Table 1–5 below lists the LIBRARIAN commands, their functions and page references to help you locate detailed descriptions.

Commands can be abbreviated to the least number of characters that make it unambiguous. A menu is provided if the command is ambiguous. Brackets indicate the shortest allowed abbreviations for each command. The part of the command in brackets may be omitted.

Table 1-5. LIBRARIAN Commands Summary

Command Name	Function	Page
Step Operation		
[PE[RFORM]]	Executes predefined steps.	1-66
File Operations		
COMPR[ESS] *	Packs files into a smaller space.	1-22
COP[Y] *	Copies files to a new location.	1-26
DE[COMPRESS] *	Decompresses files.	1-31
LCOMPA[RE] *	Shows differences between text files.	1-46
LOCK	Places files on hold.	1-52
MO[VE] *	Moves files to a new location.	1-53
OR[PHAN]	Disables tracking of read-mode secondary files.	1-61
OV[ERLAY]	Replaces a file's contents with another file.	1-62
PCR[ECEIVE]	Transfers files to a PC.	1-64
PCS[END]	Transfers files from a PC.	1-65
PR[INT] *	Prints files.	1-76
PU[RGE] *	Deletes files from the system.	1-81
REL[EASE] †	Removes MPE security from a file.	1-88
REN[AME] *	Renames files.	1-90
RESET * (or **)	Releases asterisk (*) last transaction reference.	1-94
RESET (EXCEPTION)	Removes exception flag on a file.	1-96
RESET (TIMESTAMP)	Replaces file timestamp in database with timestamp from file label.	1-99
REST[ORE]	Restores retained files.	1-101
SCA[N] *	Scans/replaces text in a file.	1-105
SCO[MPARE] † *	Accesses the S/COMPARE utility.	1-109
SEC[URE] †	Restores MPE security to a file.	1-113
SET * (or **)	Freezes the transaction fileset that asterisk (*) will refer to in future transactions.	1-115
SET (EXPDATE)	Changes a file's expiration date.	1-118
SET (LANGUAGE)	Allows you to assign a programming language, used when annotating source code.	1-120
SET (LOCKWORD) †	Assigns a new lockword to a file.	1-121
SET (MODE)	Changes access mode of a secondary file.	1-122
SET (OWNER)	Changes the file owner.	1-124
SET (TAG)	Assigns a user-defined tag to a file or group of files.	1-128

Continued

Table 1-5. LIBRARIAN Commands Summary (continued)

Command Name	Function	Page
File Operations (continued)		
T[OUCH] † *	Updates the MPE file modification timestamp with current date and time.	1–131
UN[LOCK]	Releases locked files.	1–133
UP[DATE]	Updates read-mode secondaries with current master.	1–134
VERSION	Defines, makes obsolete, and deletes versions.	1–159
Information		
H[ELP] or ?	Provides comprehensive online help.	1–43
ME[MO] or MAIL	Sends a message to a particular user or to audit trail.	1–55
SH[OWLOG]	Invokes SHOWLOG utility from LIBRARIAN.	1–130
V[ERIFY]	Shows information about files and versions.	1–138
Other Activities		
AC[TIVATE] †	Activates a suspended process.	1–19
AL[LOW]	Temporarily gives current user capabilities of another user.	1–20
AU[TOUPDATE]	Runs Auto Fileset Update (AUTOUPDP) utility.	1–21
CLE[ANDB]	Purges records from LIBDB tracking database.	1–23
CLO[SE]	Terminates a remote link.	1–24
CONNECT †	Opens connection to a remote system.	1–27
DO	Repeats execution of the last command.	1–37
ED[IT] †	Accesses configured editor to edit a file.	1–38
EX[IT] or Q[UIT]	Terminates an active LIBRARIAN session.	1–39
FLUSH	Runs the FLUSH utility.	1–40
FLUSHL[OG]	Runs the FLUSHLOG utility.	1–41
FM[AINT]	Accesses the FMAINT user fileset module.	1–42
K[ILL] †	Terminates a suspended process.	1–45
[LIBSCR[EEN]]	Accesses LIBRARIAN screens.	1–49
LIS[TREDO]	Lists the contents of LIBRARIAN command stack.	1–50
LM[AINT]	Accesses the listfile maintenance module.	1–51
MA[KE] †	Runs the MAKE utility.	1–53
PRO[JECT] †	Maintains the project definitions, status, and user authorization information for the project.	1–79
QUIET	Sets prompt and display level for the current LIBRARIAN session.	1–83
RED[O]	Edits previous command entry.	1–86

Continued

Table 1-5. LIBRARIAN Commands Summary (continued)

Command Name	Function	Page
Other Activities (continued)		
R1 or R7	Executes Reflection commands within LIBRARIAN.	1-85
RED[O]	Edits previous command entry.	1-86
RESET (APPLICATION)	Resets an application set with SET APPLICATION.	1-95
RESET (PROJECT)	Resets a project set with SET PROJECT.	1-88
RESET (ROUTE)	Resets a route set with SET ROUTE.	1-98
RET[RY] †	Sets number of times to retry remote systems that are not accessible.	1-104
SET (APPLICATION)	Sets the application a user is currently working on.	1-117
SET (PROJECT)	Sets the project a user is currently working on.	1-126
SET (PROCEDURE)	Instructs LIBRARIAN to catalog all procedures in a macro or procedure file.	1-125
SET (ROUTE)	Sets the route a user is currently working on.	1-127
US[ER]	Establishes the current user.	1-136
X[EQ]	Executes a macro.	1-161
† MPE only		
* An X version of this command is available.		

The following pages contain descriptions of all the commands, in alphabetical order. Each command description includes the following information:

Restrictions	Minimum user capability necessary to execute the command.
Menu Mode	How to perform the operation from menu mode (if applicable).
Command Mode Syntax	How the command is entered from the command line.
Parameters	Detailed descriptions of each command parameter
Operation	Basic function and detailed descriptions of the command
Examples	Examples of how to use the command
Related Information	Locations of related information

The following commands will also be covered in other chapters in more detail:

FMAINT commands are used to define and maintain user filesets, as described separately in Chapter 2, "User Fileset Commands".

LMAINT commands are used to create and maintain indirect files, as covered in Chapter 3, "Listfile Maintenance Commands".

SHOWLOG commands, used to generate custom log reports, as explained in Chapter 4, "SHOWLOG Commands".

ACTIVATE

Activates a suspended process.

Restrictions

None

Menu Mode

Not available in menu mode. Press F2 for command mode and then use **ACTIVATE** as described below.

Command Mode Syntax

```
>AC[TIVATE] [ PIN ]
```

Parameters

PIN	A process ID number.

Operation

ACTIVATE allows you to activate any suspended process created with the **RUN** command. To select a process to be activated, you must know its process ID number. If you enter **ACTIVATE** without the parameter, LIBRARIAN shows you the current processes and their corresponding process ID numbers.

Examples

Activate process number 93 by typing:

```
>ACTIVATE 93
```

If you do not know what processes are running or the ID number of the process, type:

```
>ACTIVATE
```

Related Information

See **KILL**

ALLOW

Temporarily gives the current user the capabilities of another user. Used in secure macro files.

Restrictions

None

Menu Mode

Not available in menu mode. Press F2 for command mode and then use **ALLOW** as described below.

Command Mode Syntax

>AL[LOW] [*userid:password*]

Parameters

userid	Name of user whose capabilities are allowed.
password	Password for user whose capabilities are allowed.

Note All LIBRARIAN user IDs and passwords are case-sensitive.

Operation

ALLOW lets you exercise another user's capabilities. These capabilities will remain in effect until you turn them off by invoking **ALLOW** without parameters.

Typically, **ALLOW** is used in a macro, where NOHELP and NOBREAK are set so that the user's password cannot be seen.

Examples

In the following secure macro, the first **ALLOW** command temporarily grants the user Paul the capabilities of Derek. The second **ALLOW** (without parameters) restores Paul's capabilities.

```
OPTION FILES, NOHELP, NOBREAK
USER PAUL
ALLOW DEREK:PASS
SET %%[] MODE=WRITE
ALLOW
```

Related Information

See **XEQ**
Chapter 9, "Macros" in the *LIBRARIAN/iX User's Guide*

AUTOUPDATE

Runs the Auto Fileset Update (**AUTOUPDP**) utility which adds files to filesets based on auto fileset descriptors.

Restrictions

LIBRARIAN Manager

Menu Mode

Select the **Autoupdate** option from the **Admin** menu.

Command Mode Syntax

```
>AU[TOUPDATE]
```

Parameters

None

Operation

AUTOUPDATE invokes the Auto Fileset Update (**AUTOUPDP**) utility. LIBRARIAN prompts you to choose an automatic update of one of the following:

1. Application (program prompts for an application)
2. Fileset (program prompts for a fileset name)
3. All filesets in all applications

The Auto Fileset Update utility determines which files belong to which filesets based on automatic fileset descriptors that you defined. It then adds any files found which are not presently members of the appropriate filesets. You can review the automatic fileset descriptors on the Auto Fileset Explosion Report (RAF10) and the Automatic Filesets (AF) screen.

You can add files individually to filesets using the Files in Filesets (FF) screen.

Examples

Invoke the Auto Fileset Update utility by typing:

```
>AUTO
```

Related Information

See Auto Filesets (AF) Screens in Chapter 5, "Screens"
Auto Fileset Explosion Report (RAF10) in Chapter 6, "Reports"
Chapter 3, "Master Library" in the *LIBRARIAN/iX Administrator's Guide*

CHECKDB

Monitors LIBRARIAN database capacities.

Restrictions

None

Menu Mode

None

Command Mode Syntax

```
>CHECKDB threshold
```

Parameters

threshold	An optional threshold for checking percentage full, expressed as a whole number between 0 and 100. The default is 90.

Operation

CHECKDB allows you to check LIBDB and LIBLOG dataset capacities from within LIBRARIAN. **CHECKDB** is available as a command from the LIBRARIAN prompt.

You can send the output of **CHECKDB** as mail to a user using the LIBRARIAN **MAIL** command with the **CHECKDB** parameter.

Example

Display all datasets that are at least percent full:

```
>CHECKDB 80

M-USER of dataset LIBDB.PUB.LIBDOC is 85% full.
D-USER-CAPS of dataset LIBDB.PUB.LIBDOC is 82% full.
```

Related Information

See the **MAIL** command.

CLEANDB

Purges records from the LIBRARIAN database for files that no longer exist on disk.

Restrictions

LIBRARIAN Manager

Menu Mode

Select the **Cleandb** option from the **Admin** menu. A dialog appears allowing you to specify files, revision criteria, and options.

Command Mode Syntax

```
>CLE[ANDB] filelist
    [ ;EXTERNAL ]
```

Parameters

filelist	A list of files, as described in *How to Refer to Files* at the beginning of this chapter. You are not allowed to refer to files that contain wildcards (e.g., @, #, *, or ?). Input to **CLEANDB** can also be the listfile, LISTFX10, created by the RFX10 exception report program.
EXTERNAL	Allows you to delete tracking for files on systems not running LIBRARIAN.

Operation

CLEANDB allows you to delete tracking records for files that were purged from disk by some means other than through LIBRARIAN; for example, via MPE's **PURGE** command or UNIX **rm** command.

Note When you use **CLEANDB** to remove the last master, related secondary, or retained file, the master filename will automatically be removed from the project fileset.

Examples

Clean up nonexistent files reported by the RFX10 exception report by typing:

```
>CLEANDB !LISTFX10
```

Related Information

See File Exceptions Report (RFX10) in Chapter 6, "Reports"

CLOSE

Terminates a remote link that was opened during the current LIBRARIAN session.

Restrictions

None

Menu Mode

Not available in menu mode. Press F2 for command mode and then use **CLOSE** as described below.

Command Mode Syntax

>CLO[SE] *system*

Parameters

system	The link to be terminated. Use the same system name as was used to establish the link.

Operation

CLOSE terminates a remote link that was opened during the current session by LIBRARIAN. The remote session is not terminated if it was established by the user prior to the current LIBRARIAN session.

Since there is a limit on the number of links that can be established during one LIBRARIAN session (currently set at 22), or if you are using a dial up, you can use the **CLOSE** command to provide space for new links (or free up the telephone line) by terminating those that are no longer required.

Examples

Terminate a remote link with the system name of DST068 by typing:

```
>CLOSE DST068
```

COMPRESS (XCOMPRESS)

Packs files into a smaller space.

Restrictions

File Owner for tracked files; read/write access for untracked files.

Menu Mode

Select the **Compress** option from the **Tools** menu. A dialog appears allowing you to specify files, revision criteria, and options.

Command Mode Syntax

```
>COMPR[ESS] filelist

     [ ;BATCH ]
     [ ;MEMO [ = memo-text ] ]
```

Parameters

filelist	A list of files, as described in *How to Refer to Files* at the beginning of this chapter.
BATCH	Processes the transaction in batch mode. Refer to *Batch Operations* at the beginning of this chapter.
MEMO = *memo-text*	Lets you include comments describing the current transaction. If you do not specify memo text, LIBRARIAN will invoke the configured editor so that you can enter the memo text.

Operation

COMPRESS packs files to save disk space. Savings vary according to file type, but generally range from 60 to 90 percent for source or data files. Compressed MPE files are assigned a special filecode of 1012 (or -1012 if the file is privileged). UNIX uses adaptive Lempel-Ziv coding, via the UNIX **compress** command. Transactions are logged.

XCOMPRESS operates on untracked files only and enforces normal file system security, unless the user has X capability.

Examples

Compress all source files in the library by typing:

```
>COMPRESS @.SOURCE.MFG
>COMPRESS /library/mfg/source/*
```

Compress version REL2.0 of the MFG-FILES fileset by typing:

```
>COMPRESS REL2.0 OF %MFG-FILES
```

COMPRESS (XCOMPRESS) (continued)

Examples, continued

Compress all of the files in batch mode with names ending in DB or names that use DB followed by two digits by typing:

MPE
```
>COMPRESS @DB.@.@, @DB##.@.@ ;BATCH
```
UNIX
```
>COMPRESS /*/db*, /*/ db[0-9][0-9] ;BATCH
```

Related Information

See Chapter 3, "File Transactions" in the *LIBRARIAN/iX User's Guide.*

CONNECT

Connects to a remote system as defined in the Network Configuration (NC), Systems (SY), and System-to-System (SS) screens.

Restrictions

LIBRARIAN Manager or Operator

Menu Mode

Not available in menu mode. Press F2 for command mode and then use **CONNECT** as described below.

Command Mode Syntax

```
>CONNECT system   [ ;NOLOGON ]
                  [ ;NORPM ]
                  [ ;NOQUIET ]
```

Parameters

system	The LIBRARIAN system name.
NOLOGON	Opens the DSLINE only. Do not log on to the remote system.
NORPM	Performs the remote login as configured, but does not initiate remote process management (interprocess communication).
NOQUIET	Instructs LIBRARIAN not to use the QUIET option when opening the DSLINE.

Operation

CONNECT is typically used in a macro to connect to a remote system using the **NORPM** option. This option allows network activities such as **DSCOPY** (used by the **INSTMPEC.INSTALL** macro to connect to remote systems to install the LIBRARIAN product). The benefit from using **CONNECT** is that it uses the LIBRARIAN network configuration and overrides information. Additionally, the **RETRY** command can be used in conjunction with **CONNECT**.

Examples

The following example connects to a remote system called JUMBLE without logging in and omitting the QUIET option when opening the DSLINE:

```
>CONNECT JUMBLE ;NOLOGON ;NOQUIET

CONNECTING TO REMOTE SYSTEM JUMBLE...OK
```

CONNECT (continued)

Related Information

See **XEQ**
RETRY
Network Configuration (NC), Systems (SY), and System-to-System (SS) screens in Chapter 5, "Screens"

UNIX If you are running UNIX, see the **rlogin** manual page for equivalent functionality.

COPY (XCOPY)

Copies files to a new location.

Restrictions

File owner for tracked files; read access for untracked files.

Menu Mode

Select the **Copy** option from the **File** menu. A dialog appears allowing you to specify files, revision criteria, and options.

Command Mode Syntax

```
>COP[Y] filelist [ TO tolocation ] [ ,filelist [ TO tolocation ] ... ]

†    [ ;ANNOTATE [ = NODELETE ] ]
     [ ;APPEND ]
     [ ;BATCH ]
     [ ;COMPRESS ]
     [ ;CREATE [ = GROUP ][ ,ACCOUNT ][ ,CREATOR ] ]
     [ ;CREATOR = creator ]
     [ ;DECOMPRESS ]
     [ ;GROUP = unixgroup ]
     [ ;KEEP ]
     [ ;LOCKWORD = lockword ]
     [ ;MEMO [ = memo-text ] ]
†    [ ;OLDDATE ]
†    [ ;ORPHAN ]
     [ ;PERMISSIONS = unixpermissions ]
†    [ ;RENUMBER ]
†    [ ;RESET ]
†    [ ;VERIFY ]
```

† Not available with XCOPY

Parameters

filelist	A list of files, as described in *How to Refer to Files* at the beginning of this chapter.
TO *tolocation*	The destination location for the filelist. The syntax is: TO [*system:*] *file* [*.group* [*.acct*]] TO [*system:*] /[*path.../*] *file* The equal sign (=) wildcard can be entered in any filename element. It takes the same value as the corresponding element in the filelist (source file). Also, a single at sign (@) can be used. It has the same meaning as the equal sign (=) wildcard. The *tolocation* can also be an Edit Mask, as described earlier in this chapter. If *tolocation* is omitted, your current login location is assumed.

COPY (continued)

Parameters, continued

ANNOTATE	Creates an annotated copy of source code based on the delta file, showing lines that were inserted and deleted for each revision, including date/time, user, and project. Annotation appears as commented code appropriate for the programming language set for the file (see **SET LANGUAGE** command). If you attempt to **ANNOTATE** files for which deltas are not being stored, LIBRARIAN will notify you that a violation has occurred. **NODELETE** Shows only insertions. Deleted lines normally appear as commented text but are not shown when the **NODELETE** option is specified.
APPEND	Appends data from the source file to the end of the destination file, if it exists. In MPE, an error occurs if the EOF plus new data exceeds the file LIMIT.
BATCH	Processes the transaction in batch mode. Refer to *Batch Operations* at the beginning of this chapter.
COMPRESS	Compresses the new destination file(s).
CREATE	Makes any directories on the destination path that do not already exist.
CREATE=GROUP, ACCOUNT, and/or **CREATOR**	Creates MPE group, account, and/or creator if it does not exist. If none of **GROUP, ACCOUNT,** or **CREATOR** is specified, all will be created if necessary.

Note H-P's Security Monitor product enables system managers to prevent the creation of users, groups, or accounts without passwords. If **CREATE** is specified and this Security Monitor feature is in use, users will receive the error message "Password required for user".

CREATOR = *creator* Overrides the MPE creator/UNIX owner of the new file. For MPE, it can be a specific MPE user. For UNIX, it can be a specific UNIX user login, or decimal user ID found in /etc/passwd. Alternatively, you can use one of the following:

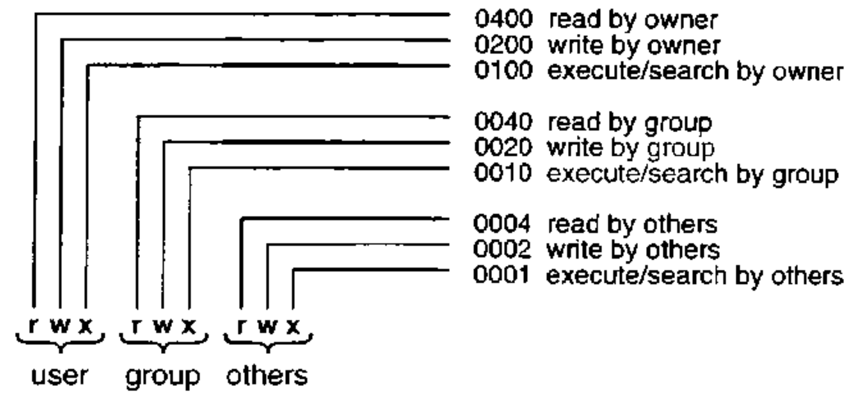
!USERID	Uses current LIBRARIAN user.
!LOGON	Uses MPE current login user.
!KEEP	Uses MPE creator/UNIX owner of the file being replaced.

Note All LIBRARIAN user IDs are case-sensitive.

COPY (continued)

Parameters, continued

DECOMPRESS	Decompresses the new destination file if the source file is compressed.
UNIX **GROUP** = *unixgroup*	Changes the group ID of the new destination file. It can be a specific UNIX group name or decimal group ID found in */etc/group*.
KEEP	Indicates not to copy a file if the destination file already exists. When using batch mode or **QUIET ON**, the default is to purge existing destination files. Otherwise, the default is to prompt. **KEEP** overrides both of these defaults.
MPE **LOCKWORD**=*lockword*	Assigns the given lockword to the destination file. You can specify !KEEP as the lockword, which instructs LIBRARIAN to use the lockword from the old destination file.
MEMO = *memo–text*	Lets you include comments describing the current transaction. If you do not specify memo text, LIBRARIAN will invoke the configured editor so that you can enter the memo text.
OLDDATE	Leaves timestamp in database as is and sets the create and modify dates of the destination file to be the same as the from file.
ORPHAN	Creates files that are not tracked by LIBRARIAN.
UNIX **PERMISSIONS** = *unixpermissions*	Changes the permissions of the new destination file according to the *unixpermissions* mode, which is a number constructed from the following mode bits:



MPE **RENUMBER**	Renumbers a numbered file when reconstructed from a delta file.
RESET	Resets the modification timestamp in the database after the copy has been completed. Generally used with the **MODIFIED** parameter.
VERIFY	Identifies files that have been modified by comparing the modification timestamp in the file label with the timestamp stored when it was last created by LIBRARIAN. If the file has been modified since it was moved into its current location, the file is not copied (violation).

COPY (continued)

Operation

COPY copies any type of file currently tracked by LIBRARIAN, including masters, retained masters, secondaries, and retained secondaries. The new destination files are owned by the user who copied them and have read-mode access.

For files not tracked by LIBRARIAN, normal file security is applied.

The new destination files are always secondary files unless they were copied from retained masters/secondaries or were untracked files to begin with. LIBRARIAN normally tracks secondary files, unless you use the **ORPHAN** parameter. For security reasons, **COPY** does not replace a file currently tracked by LIBRARIAN; instead, use the **OVERLAY** command or a defined step.

If a version or revision is specified, the original name of the retained file is used in the new destination location.

XCOPY operates on untracked files only and enforces normal file system security, unless the user has X capability.

Examples

Copy all secondary files of the FINANCE fileset that currently reside in a particular development location:

MPE
```
>COPY %FINANCE AT @.@.FINDEVEL
```

UNIX
```
>COPY %FINANCE AT /usr/findevel/bin/*
```

Copy all of the current and retained master files in version REL1.0 of the FINANCE fileset to another location with the same filename:

MPE
```
>COPY REL1.0 OF %FINANCE TO =.=.=.PAYR
```

UNIX
```
>COPY REL1.0 OF %finance TO /apps/payroll
```

Copy the finance source files on system SYS68 to files of the same name in your current logon:

MPE
```
>COPY SYS68:@.SOURCE.FINANCE TO =.=
```

UNIX
```
>COPY sys68:/usr/findevel/src/* TO ./=
```

From your login, copy all files ending in E to files of the same name in one location, and then copy all files ending in J or S to files of the same name to another location:

MPE
```
>COPY @E TO =.EXE, @J, @S TO =.SOURCE
```

UNIX
```
>COPY *e TO ../bin/= *j,*s TO ../src/=
```

COPY (continued)

Examples, continued

Copy master files to your login. Exclude all retained files and all files modified since they were moved to their current location. If a destination file of that name exists, abort the copy.

MPE

```
>COPY @.SOURCE.MASTER TO =,-G#######.SOURCE;VERIFY;KEEP
```

UNIX

```
>COPY master/src/* TO src/=, −master/src/.g???????
```

Related Information

See **OVERLAY**
UPDATE

DECOMPRESS (XDECOMPRESS)

Decompresses files that were compressed using the **COMPRESS** command, the **COMPRESS** parameter, or a step.

Restrictions

File Owner; read/write access for untracked files.

Menu Mode

Select the **Decompress** option from the **Tools** menu. A dialog appears allowing you to specify files, revision criteria, and options.

Command Mode Syntax

```
>DE[COMPRESS] filelist

    [ ;BATCH ]
    [ ;MEMO [ = memo-text ] ]
```

Parameters

filelist	A list of files, as described in *How to Refer to Files* at the beginning of this chapter.
BATCH	Processes the transaction in batch mode. Refer to *Batch Operations* at the beginning of this chapter.
MEMO = *memo-text*	Lets you include comments describing the current transaction. If you do not specify memo text, LIBRARIAN will invoke the configured editor so that you can enter the memo text.

Operation

DECOMPRESS restores files to the state they were in before being compressed.

XDECOMPRESS operates on untracked files only and enforces normal file system security, unless the user has X capability.

Examples

Decompress all files in the SOURCE group of your login account by typing:

```
>DECOMPRESS @.SOURCE
```

Decompress retained version REL2.0 of the MFG-FILES fileset by typing:

```
>DECOMPRESS REL2.0 OF %MFG-FILES
```

DECOMPRESS (XDECOMPRESS) (continued)

Examples , continued

Decompress the files on the current system with names ending in DB or names that use DB followed by two digits in batch by typing:

MPE
```
>DECOMPRESS @DB.@.@, @DB##.@.@;BATCH
```

UNIX
```
>DECOMPRESS /*/*db/*,/*db[0-9][0-9] ;BATCH
```

Related Information

See **COMPRESS**
Chapter 3, "File Transactions" in the *LIBRARIAN/iX User's Guide*

DELETE

Deletes an application and all associated steps and file tracking records.

Restrictions

LIBRARIAN Manager

Menu Mode

Not available in menu mode. Press F2 for command mode and then use **DELETE** as described below.

Command Mode Syntax

```
>DELETE appl-id
```

Parameters

appl-id	Application ID of application to be deleted.

Operation

DELETE deletes all application-related routes, steps, and file tracking. You are prompted to confirm the deletion.

Examples

```
    >DELETE MFG
```

Related Information

See Applications (AP) screen

DO

Repeats the execution of a command saved in the LIBRARIAN command stack.

Restrictions

None

Menu Mode

Not available in menu mode. Press **F2** for command mode and then use **DO** as described below.

Command Mode Syntax

```
>DO [ cmdid ]
```

Parameters

cmdid Command to re-execute. You can specify one of the following.

-n	The *n*th command before the most current one, where *n* is a number in the command line stack relative to most recent command, which is -1.
m	Number *m* in the command line stack. The number *m* is absolute (not relative).
string	Most recent command beginning with the text string.

If you do not specify a value after the command, LIBRARIAN re-executes the previous command (i.e., the default is -1).

Operation

Use **DO** to execute a command again.

Examples

Repeat the previous command by typing:

```
>DO
```

Repeat command 20 by typing:

```
>DO 20
```

Related Information

See **LISTREDO**
REDO

EDIT

Accesses the editor of your choice to edit a file.

Restrictions

None

Menu Mode

Not available in menu mode. Press F2 for command mode and then use **EDIT** as described below.

Command Mode Syntax

```
>ED[IT] [ filename ]
```

Parameters

filename	Name of the file you want to edit. This is only valid if your editor can accept a filename passed via the INFO string.

Operation

EDIT invokes the editor specified in the LIBRARIAN configuration file. You can override the default editor by issuing a file equation or setting the system variable LIBEDITOR. For example,

```
:SETVAR LIBEDITOR "QEDIT.PUB.ROBELLE"
```

Examples

Edit the file RPT100S using QEDIT, where QEDIT is the default editor by typing:

```
>EDIT RPT100S
```

Related Information

See Chapter 3, "File Transactions" in the *LIBRARIAN/iX User's Guide*

UNIX If you are running UNIX, see the **vi** and **ex** manual entries for equivalent functionality.

EXIT (or QUIT)

Terminates an active LIBRARIAN session.

Restrictions

None

Menu Mode

Select the Exit option from the File menu or press the **F8** (**Exit**) function key.

Command Mode Syntax

```
>EX[IT] or Q[UIT]
```

Parameters

None

Operation

EXIT (or **QUIT**) terminates the current LIBRARIAN session. Remote sessions initiated by LIBRARIAN are terminated. Remote sessions that you initiated prior to invoking LIBRARIAN are left as they were.

Examples

Exit the LIBRARIAN program and return to the MPE or UNIX shell by typing:

```
>EXIT
```

FLUSH

Runs the FLUSH utility to purge expired retained and secondary files.

Restrictions

LIBRARIAN Manager or Operator

Menu Mode

Select the **Flush** option from the **Admin** menu.

Command Mode Syntax

```
>FLUSH
```

Parameters

None

Operation

FLUSH purges read–mode secondaries that have expired. Expiration dates for read–mode secondaries are determined by the policy of the steps that create them. You can also assign expiration dates using the **SET EXPDATE** command.

FLUSH also purges retained masters and secondaries, based on the following criteria:

- The System Profile (SP) Flush Policy allows the LIBRARIAN Manager to specify the minimum number of retained revisions to keep, even when the retained files have expired. If you are using branching, this number reflects the number of branches off the trunk revisions that you want to keep. Any related branch revisions are also kept.
- **FLUSH** will only purge retained files which have reached their expiration dates, and are not protected by the Flush Policy described above. Expiration dates for retained files are determined by the Safety Retention Policy for the steps that create these files. You can use the **SET EXPDATE** command to set the expiration date on retained revisions.
- Retained master files that have a Version Count (VCOUNT) of 0 are never purged, unless the version baseline to which the revision belongs is obsolete. Use the **VERSION** command to make a version obsolete.

To review the files that **FLUSH** is ready to purge, run the RFN10 and/or RFN20 reports.

Examples

Invoke the FLUSH utility by typing;

```
    >FLUSH
```

Related Information

See Chapter 7, "Versions" in the *LIBRARIAN/iX Administrator's Guide* and Pre–Flush Notification Reports (RFN10/RFN20) in Chapter 6, "Reports"

FLUSHLOG

Runs the FLUSHLOG utility which flushes log records older than the number of days specified in the System Profile (SP) screen.

Restrictions

LIBRARIAN Manager or Operator

Menu Mode

Select the **Flushlog** option from the **Admin** menu.

Command Mode Syntax

```
>FLUSHL[OG]
```

Parameters

None

Operation

The System Profile (SP) Log Records Aging Policy determines the transactions to be flushed from the audit trail. **FLUSHLOG** purges records for transactions older than the number of days specified in the policy. Transactions associated with projects, however, are exceptions. **FLUSHLOG** will only purge transactions associated with projects, when the project status is **FLUSH PENDING**. You can change the project status on the PS screen.

You can also use **SHOWLOG** to selectively flush transactions.

Examples

Purge all log transaction records by typing:

```
    >FLUSHLOG
```

Related Information

See System Profile (SP) screen in Chapter 5, "Screens"
Appendix D, "Customizing System Parameters" in the *LIBRARIAN/iX Administrator's Guide*

FMAINT

Accesses the user fileset maintenance (FMAINT) utility.

Restrictions

None

Menu Mode

Select **User Filesets** from the **Tools** menu. A menu appears listing FMAINT operations.

Command Mode Syntax

```
>FM[AINT]
```

Parameters

None

Operation

FMAINT accesses the user fileset module that consists of separate commands to allow users to create and maintain filesets for their own use. User filesets are convenient and easy to use because they allow you to group files according to your needs.

Once invoked, the FMAINT module signals its active status with a FM> prompt, and only FMAINT commands are processed.

Examples

Access the user fileset module by typing:

```
    >FMAINT
```

Related Information

See Chapter 2, "User Fileset Commands"
    Chapter 6, "User Filesets" in the *LIBRARIAN/iX User's Guide*

HELP

Accesses online help for information about LIBRARIAN commands, about the steps you can perform, about a specific step, or about projects you are authorized to work on.

Restrictions

None

Menu Mode

For context sensitive help, press F1. For general help, select the **Help** option from the menu bar. Another menu appears with a variety of help options.

Command Mode Syntax

```
>H[ELP]  {  command [ option ]          }
         {  PROCEDURES                  }
         {  PROJECTS                    }
         {  STEPS                       }
         {  step [ .route [ .appl ] ]   }
         {  macro-name                  }
```

Parameters

command	Any LIBRARIAN command name. If omitted, a brief description of LIBRARIAN appears with a list of commands.
option	Type of help you want to access for a command. Use one of the following option keywords.

PARMS	Describes command parameters.
OPERATION	Describes operation of the command.
EXAMPLE	Displays an example of the command.
ALL	Displays all information for the command.

If you do not specify an option, LIBRARIAN displays the command syntax.

PROCEDURES	Shows names of all procedures that were made available with the **SET PROCEDURE** command
PROJECTS	Lists authorized projects for the current LIBRARIAN user.
STEPS	Lists authorized steps for the current LIBRARIAN user.
step	The step for which you want information.
route	Route associated with the step. If LIBRARIAN cannot identify the route, it displays a list of steps from which you can choose.
appl	Application associated with the step. If LIBRARIAN cannot identify the application, it displays a list of steps from which you can choose.
macro−name	Displays corresponding macro/procedure help if it exists; otherwise, it displays the contents of the macro.

HELP (continued)

Operation

The LIBRARIAN online help module is window–based to let you scroll through the help text for a particular topic. The **F1** function key opens the help index which includes the complete LIBRARIAN Glossary, User's Guide, and Reference Guide. Some help topics have related topics that you can access. To do this, highlight a topic and then press ENTER. To exit the help module, press the **F8** function key.

Examples

Display the syntax for the **PERFORM** command by typing:

```
>HELP PERFORM
```

Display examples of the **MOVE** command by typing:

```
>HELP MOVE EXAMPLE
```

List all of the steps the current user is authorized to perform by typing:

```
>HELP STEPS
```

Display help for the CHECKOUT step in the DEVEL route of the FIN application by typing:

```
>HELP CHECKOUT.DEVEL.FIN
```

List of all the projects the current user is authorized to perform by typing:

```
>HELP PROJECTS
```

KILL

Terminates a son process created with the MPE **RUN** command.

Restrictions

None

Menu Mode

Not available in menu mode. Press F2 for command mode and then use **KILL** as described below.

Command Mode Syntax

```
>K[ILL] [ PIN ]
```

Parameters

PIN	A process ID number.

Operation

KILL terminates son processes created by the **RUN** command. To select a process to terminate, you must know its process ID number. If you invoke **KILL** without the parameter, LIBRARIAN shows you current processes and their corresponding process ID numbers.

Examples

Terminate process number 93 by typing:

```
>KILL 93
```

If you do not know what processes are running or the ID number of the process, type:

```
>KILL
```

Related Information

If you are running UNIX, see the **ps** and **kill** manual entries for equivalent functionality.

LCOMPARE (XLCOMPARE)

Shows the differences between text files.

Restrictions

Owner for tracked files; read access for untracked files.

Menu Mode

Select the **Compare** option from the **Tools** menu. A dialog appears allowing you to specify files, revision criteria, and options. The method of comparison is determined by the compare method that you can change on the **Settings** window from the **User** menu.

Command Mode Syntax

```
>LCOMPA[RE] filelist1 [TO filelist2]

     [ ;ALL ]
     [ ;BATCH ]
†    [ ;DELTA ]
†    [ ;FROMREV = revid ] or
†    [ ;FROMVCOUNT = fromvcount ] or [ ;FROMGCOUNT = fromgcount ]
†    [ ;TOREV = revid ] or
†    [ ;TOVCOUNT = tovcount ] or [ ;TOGCOUNT = togcount ]
†    [ ;MASTER]
     [ ;NOBLANKS ]
     [ ;OFFLINE ]
     [ ;UNNUMBERED ]
     [ ;UPSHIFT ]
```

† Not available with **XLCOMPARE**

Parameters

filelist1	A list of files, as described in *How to Refer to Files* at the beginning of this chapter.
TO *filelist2*	The reference files to be used for comparisons. You can omit this parameter when comparing revisions of the same file. When *filelist1* specifies multiple files, i.e. wildcard, indirect file, etc., *filelist2* must be a wildcard expression that maps each authorized file to a corresponding filename. You cannot use an indirect file as a *filelist2*.
ALL	Shows all lines of the reference file. The default is to show ten lines before and after file differences.
BATCH	Processes the transaction in batch mode. Refer to *Batch Operations* at the beginning of this chapter.
DELTA	Reconstructs the reference file from the delta file corresponding to the master to check the integrity of the current master. With this option you do not need to specify a **TO** file.
FROMREV = *revid*	Specifies which revision to retrieve as the compare file.

LCOMPARE (continued)

Parameters, continued

FROMVCOUNT *= fromvcount*	Uses compare files with a version count equal to *fromvcount* This parameter must have a positive value and be used in conjunction with the *versionid* parameter.
FROMGCOUNT *= fromcount*	Uses compare files with a generation count equal to *fromgcount*. This parameter can have either a positive or negative value. A negative value describes the generation equal to the current generation minus the **GCOUNT**. For example, GCOUNT=-2 specifies the file two generations earlier than the current generation.
TOREV *= revid*	Same as **FROMREV**, but applies to the reference file.
TOVCOUNT *= tovcount*	Same as **FROMVCOUNT** described earlier, but applies to the reference.
TOGCOUNT *= togcount*	Same as the **FROMGCOUNT** parameter described earlier, but applies to the reference files.
MASTER	Compares secondaries to their associated master. With this option, you do not need to specify a **TO** file.
NOBLANKS	Ignores spacing and blank lines when comparing files.
OFFLINE	Prints file(s) offline to device LP.
UNNUMBERED	Ignores numbering when comparing files.
UPSHIFT	Ignores case when comparing files.

Operation

For each reference file, **LCOMPARE** shows lines inserted and/or deleted to result in the compare file.

LCOMPARE's output also provides LIBRARIAN file information, including the file name, file type, versions, and generations of the files being compared. If the files are compressed, **LCOMPARE** automatically decompresses them before a compare operation, then compresses them again after the completion of the operation.

XLCOMPARE operates on both tracked and untracked files, and enforces normal file system security, unless the user has X capability.

You can specify a set of escape sequences for printer initialization and bold print in a file called PRINTESC. The first line of the file is for printer initialization. The second line is for bold print, and the third line is for normal print. You can use the ESC keyword in the file to indicate the **Escape** character. In addition, you can use the keyword UNDERLINE rather than an escape sequence for bold. You can use file equations for the PRINTESC file.

Operation, continued

The following example sets escape sequences for the HP laser jet printer.

```
ESC(s10H
ESC(s3B
ESC(s0B
```

Examples

Compare the files listed in LOGFILES to files of the same name in another location. The output device is LP. To do this, type:

```
LCOMPARE > !LOGFILES TO =.=.DEVEL;OFFLINE
LCOMPARE > !LOGFILES TO /usr/devel/=;OFFLINE
```

The following example compares the current master against the same file reconstructed from the delta file. The result should be that the files match, otherwise there is an integrity problem.

```
LCOMPARE > ABC1000S.SOURCE.ACCTG;DELTA
LCOMPARE> /usr/master/acctg/abc100s.c;DELTA
```

The following example compares a secondary file to its associated master.

```
LCOMPARE > ABC1000S.SOURCE.LIBDEVEL;MASTER
LCOMPARE> /usr/devel/acctg/abc1000s.c;MASTER
```

Related Information

See Chapter 5, "Printing, Scanning, and Comparing Files" in the *LIBRARIAN/iX User's Guide*

LIBSCREEN

Accesses the LIBRARIAN screen system.

Restrictions

Applications Manger/Project Manager

Menu Mode

Select the **Screen** option from the **Admin** menu. A number of screen categories appear. Choose a category and select a screen.

Command Mode Syntax

>[LIBSCR[EEN]] *screen code*

Parameters

screen code	Any valid LIBRARIAN screen code.

Operation

LIBSCREEN goes directly to the specified screen after validating your current LIBRARIAN user ID, if not already specified. You can omit the command **LIBSCREEN**, and access any LIBRARIAN screen by its screen code alone.

Examples

Access the File Inquiry (FI) screen directly from the LIBRARIAN prompt by typing:

```
>FI
```

Related Information

See Chapter 5, "Screens"

LISTREDO

Lists the contents of the LIBRARIAN command stack (up to 20 commands) with corresponding reference numbers.

Restrictions

None

Menu Mode

Not available in menu mode. Press **F2** for command mode and then use **LISTREDO** as described below.

Command Mode Syntax

```
>LIS[TREDO]
```

Parameters

None

Operation

Use **LISTREDO** to list a maximum of 20 previously issued commands with corresponding reference numbers. The LIBRARIAN **LISTREDO** command functions similarly to the MPE's **LISTREDO** command. For more information, refer to the *MPE Commands Reference Manual*.

Examples

List the previous commands (up to 20) by typing:

```
    >LISTREDO
```

Related Information

See **DO**
REDO

LMAINT

Accesses the listfile maintenance module.

Restrictions

None

Menu Mode

Select the **Listfiles** option from the **Tools** menu. A menu appears listing LMAINT operations.

Command Mode Syntax

```
>LM[AINT]
```

Parameters

None

Operation

The **LMAINT** command accesses LMAINT, the listfile maintenance module that consists of separate commands, allowing users to create, maintain, and view listfiles. Listfiles (indirect files) are files containing a list of filenames, with or without system names.

Once invoked, the LMAINT module signals its active status with a **LM>** prompt, and only LMAINT commands are processed. Return to the LIBRARIAN prompt by typing **EXIT**.

Examples

Access the listfile maintenance module by typing:

```
>LMAINT
```

Related Information

See Chapter 7, "Listfiles" in the *LIBRARIAN/iX User's Guide*

LOCK

Places files on hold so they cannot be moved or copied.

Restrictions

File Owner

Menu Mode

Select the **Set...Lock/Unlock** option from the **File** menu. A dialog appears allowing you to specify files, revision criteria, and options.

Command Mode Syntax

```
>LOCK filelist

     [ ;BATCH ]
     [ ;MEMO [ = memo-text ] ]
```

Parameters

filelist	A list of files, as described in *How to Refer to Files* at the beginning of this chapter.
BATCH	Processes the transaction in batch mode. Refer to *Batch Operations* at the beginning of this chapter.
MEMO = *memo-text*	Lets you include comments describing the current transaction. If you do not specify memo text, LIBRARIAN will invoke the configured editor so that you can enter the memo text.

Operation

LOCK protects a file from being copied or moved from its current location. Use the **UNLOCK** command to release files.

Examples

Lock up all tracked files in your login group:

MPE
```
>LOCK @
```
UNIX
```
>LOCK ./*
```

Lock all secondary copies of the FIN fileset by typing:

MPE
```
>LOCK %FIN AT @.@.@
```
UNIX
```
>LOCK %FINANCE AT ./*
```

Related Information

See **UNLOCK**

MAKE

Scans the specified makefile(s) for dependency rules. It determines if specified target(s) need to be rebuilt and then generates a MAKE job with commands to bring the target up-to-date.

Restrictions

None

Menu Mode

Select the **Make** option from the **Tools** menu. A dialog appears allowing you to specify a makefile and other options.

Command Mode Syntax

>MA[KE] [*makefile* [*,target* [*,listfile* [*,cmdfile*]]]]

    [;ALL]
    [;ECHO]
    [;NOMAKE]
    [;QUIET]
    [;SHOW]
    [;XEQ]

Parameters

makefile	A file specification consistent with MPE **LISTF** conventions, with or without wildcards, specifying one or more files containing MAKE rules. If omitted, the default name and formal file designator is MAKEFILE.
target	Name of the target to be (re)built. If omitted, the default is the target of the first rule found.
listfile	Name of the file into which a compile listing is generated. If omitted, the default name and formal file designator is MAKELIST.
cmdfile	Name of the file into which MAKE generates commands. If omitted, the default name and formal file designator is MAKEOUT. The file is temporary.
ALL	Builds all components, regardless of whether or not they are out of date.
ECHO	Displays echo comments in makefile(s) that begin with NOTE.
NOMAKE	Does not automatically stream the MAKEOUT job.
QUIET	Does not output summary information to terminal.
SHOW	Produces detailed progress reports to the terminal.
XEQ	Creates a MAKEOUT file, without streaming the file, that can be used used as a LIBRARIAN macro.

MAKE (continued)

Examples

Process all makefiles in the BUILD group echoing NOTE comments as they are processed. Use the default names for the compile listing file and command output file. To do this, type:

```
>MAKE MAKE@.BUILD;ECHO
```

Related Information

See Chapter 8, "Rebuilding Applications with MAKE" in the *LIBRARIAN/iX User's Guide*

If you are running UNIX, see the **make** manual page for equivalent functionality.

MEMO (or MAIL)

Sends a message to a particular user or the audit trail. **MAIL** can also alert users when datasets are getting near capacity.

Restrictions

None

Menu Mode

Select the **Memo** option from the **User** menu. A dialog appears allowing you to specify who to send mail to and a message. Press **F6** to send mail.

Command Mode Syntax

>ME[MO] (or MAIL) **[** *userid*/$LOG] [;*memotext*/;**CHECKDB** [= *threshold*]]

Parameters

userid	The user to receive the message.
Note	All LIBRARIAN user IDs are case-sensitive.
$LOG	Specifies that the memo is to be added to the audit trail.
memotext	Text consisting of a maximum of 72 characters. If omitted, an editor is invoked for a multi-line memo.
CHECKDB = *threshold*	Checks LIBDB and LIBLOG dataset capacities and notifies a user if any datasets are getting close to capacity. An optional threshold for checking percentage full, expressed as a whole number between 0 and 100. The default is 90.

Operation

With **MEMO** (or **MAIL**) you can send messages to other users. In addition, LIBRARIAN sends messages (e.g., notification of exceptions). These messages are checked before every command. Users are informed if there are new messages. You can retrieve your messages using the **MEMO** command without parameters.

If you use the **CHECKDB** parameter to monitor LIBRARIAN dataset capacities, mail will only be sent if there are datasets that exceed the threshold.

Example

Send a message to Derek by typing:

```
>MEMO DEREK ;DON'T FORGET TO CHECK THOSE XEQ FILES BACK IN.
```

Related Information

See the **CHECKDB** command.

MOVE (XMOVE)

Move files from one location to another.

Restrictions

File Owner for tracked files; read access for untracked files.

Menu Mode

Select the **Move** option from the **File** menu. A dialog appears allowing you to specify files, revision criteria, and options.

Command Mode Syntax

```
>MO[VE] filelist [ TO tolocation ] [ ,filelist [ TO tolocation ] ...]

     [ ;APPEND ]
     [ ;BATCH ]
     [ ;COMPRESS ]
     [ ;CREATE [ = GROUP ][ ,ACCOUNT ][ ,CREATOR ] ]
     [ ;CREATOR = creator ]
     [ ;DECOMPRESS ]
†    [ ;EXTERNAL ]
     [ ;GROUP = unixgroup ]
     [ ;KEEP ]
     [ ;LOCKWORD = lockword ]
     [ ;MEMO [ = memo-text ] ]
†    [ ;OLDDATE ]
†    [ ;OLDNAME ]
     [ ;PERMISSIONS = unixpermissions ]
†    [ ;VERIFY ]
```

† Not available with **XMOVE**.

Parameters

filelist	A list of files, as described in *How to Refer to Files* at the beginning of this chapter.
TO *tolocation*	The destination location for the filelist. The syntax is:
MPE	TO [*system:*] *file* [*.group* [*.acct*]]
UNIX	TO [*system:*] /[*path...*/] *file*
	The = wildcard can be entered in any filename element. It takes the same value as the corresponding element in the filelist (source file). A single @ can also be entered. It has the same meaning as the = wildcard. *tolocation* can also be an edit mask, as described earlier in this chapter.

MOVE (continued)

Parameters, continued

APPEND	Appends a file to an existing file with the same name.
BATCH	Processes the transaction in batch mode. Refer to *Batch Operations* at the beginning of this chapter.
COMPRESS	Compresses the new destination file(s).
CREATE	Makes any directories on the destination path that do not already exist.
CREATE=GROUP, ACCOUNT, and/or **CREATOR**	Creates MPE group, account, and/or creator if it does not exist. If none of **GROUP, ACCOUNT,** or **CREATOR** is specified, all will be created if necessary.

Note H–P's Security Monitor product enables system managers to prevent the creation of users, groups, or accounts without passwords. If **CREATE** is specified or configured on a step, and this Security Monitor feature is in use, users will receive the error message "Password required for user".

CREATOR = *creator* Overrides the MPE creator/UNIX owner of the new file. For MPE, it can be a specific MPE user. For UNIX, it can be a specific UNIX user login, or decimal user ID found in /etc/passwd. Alternatively, you can use one of the following:

!USERID	Uses current LIBRARIAN user.
!LOGON	Uses current login user.
!KEEP	Uses MPE creator/UNIX owner of the file being replaced.

Note All LIBRARIAN user IDs are case-sensitive.

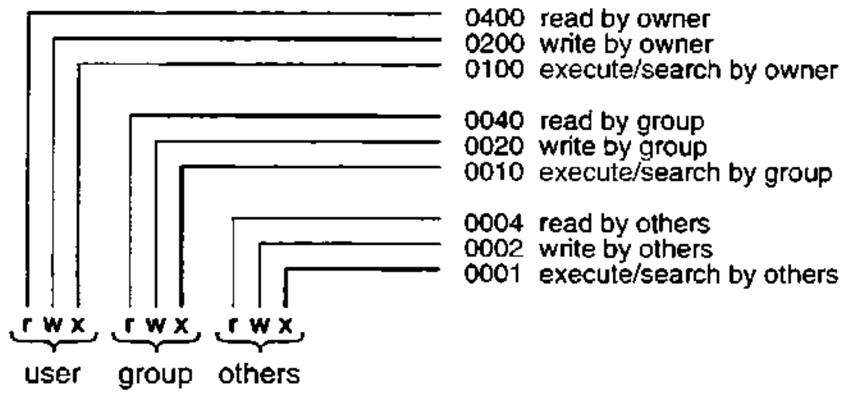
DECOMPRESS	Decompresses the new destination file(s).
EXTERNAL	Records the movement of a master or secondary file to a new location in the LIBRARIAN database. LIBRARIAN tracks the file in the new location, but does not physically move the files. Physical movement of files to the new location can be accomplished outside of LIBRARIAN.

You can use this parameter to record the move of an entire master library to a new system, then physically move the files using MPE's STORE/RESTORE command.

Another example would be to logically move secondary files from or to a PC.

Parameters, continued

UNIX **GROUP** = *unixgroup*	Changes the group ID of the new destination file. It can be a specific UNIX group name or decimal group ID found in /etc/group.
KEEP	Instructs LIBRARIAN not to copy a file if the destination file already exists. When you use batch mode or **QUIET ON**, the default is to purge existing destination files. Otherwise, the default is to prompt. **KEEP** overrides both of these defaults.
MPE **LOCKWORD** = *lockword*	Assigns a lockword to the destination file created by the operation. You can specify !KEEP as the lockword, which instructs LIBRARIAN to use the lockword from the old destination file.
MEMO = *memo-text*	Lets you include comments describing the current transaction. If you do not specify memo text, LIBRARIAN will invoke the configured editor so that you can enter the memo text.
OLDDATE	Leaves timestamp in the database as is and sets the create and modify dates of the destination file to be the same as the from file.
OLDNAME	Uses the original name of the retained file in the new destination location. Group, account, and system are established by the *tolocation* specification.
UNIX **PERMISSIONS** = *unixpermissions*	Changes the permissions of the new destination file according to the *unixpermissions* mode, which is a number constructed from the following mode bits:



VERIFY	Identifies files that have been modified by comparing the modification timestamp in the file label with the timestamp stored when it was last created by LIBRARIAN. If the file has been modified since it was moved into its current location, the file is not copied (violation).

MOVE (continued)

Operation

MOVE moves any file currently tracked by LIBRARIAN, including masters, retained masters, secondaries, and retained secondaries. **MOVE** copies a file into a new location and purges the source file.

When a master file is moved, all references to the former identity of the file are replaced with references to the new identity so that the links between a master file and its secondaries are not severed. The new file location is updated for all filesets in which the moved file appears. When a KSAM file is moved, the name of the associated key file does not change.

For security, **MOVE** cannot replace an existing file that is being tracked by LIBRARIAN.

Note When you use **MOVE** to remove the last master, related secondary, or retained master associated with a project, the old filename will automatically be removed and the new one will be added.

XMOVE operates on untracked files only and enforces normal file system security, unless the user has X capability.

Examples

Move the FINANCE fileset to another system and retain the same file, group, and account names by typing:

```
MPE     >MOVE %FINANCE TO SYS78:=.=.=

UNIX    >MOVE %FINANCE TO SYS78:/(1,*)
```

Move the FINANCE fileset to another system and retain the same file, group, and account names by typing:

```
MPE     >MOVE %FINANCE TO SYS78:=.=.=

UNIX    >MOVE %finance to sys78:/(1,*)
```

Move all of the files that have a J as the last character of the filename from your login group to files of the same name in the JOB group by typing:

```
MPE     >MOVE @J TO =.JOB

UNIX    >MOVE *j TO job/=
```

Move to your login the third generation of all master files by typing:

```
        >MOVE %SOURCE;GCOUNT = 3
```

MOVE (continued)

Examples , continued

From your login group, move all master files ending in E to files of the same name in the EXE group. Then,copy all master files ending in J or S to files of the same name in the SOURCE group by typing:

```
>MOVE @E TO =.EXE, @J, @S TO =.SOURCE
```

```
>MOVE *e TO exe/=, *j, *s TO source/=
```

Move all master files in the SOURCE area to files of the same name in your login area. Exclude all retained files and all files modified since they were moved to their current location. If a destination file of that name exists, abort the copy. To do this, type:

```
>MOVE @.SOURCE TO =, -G#######.SOURCE;VERIFY;KEEP
```

```
>MOVE src/* TO src/=, −src/.g####### ;VERIFY;KEEP
```

Copy all of the current and retained master files in REL2.0 of the MFG-FILES fileset to files of the same name in the REL2GRP group of the MFG account. Use the original names of the files. To do this, type:

```
>MOVE REL2.0 OF %MFG-FILES TO =.REL2GRP.MFG; OLDNAME
```

```
>MOVE rel2.0 OF %MFG−FILES TO /usr/mfgdevel/= ;OLDNAME
```

Related Information

See Chapter 3, "Master Library" in the *LIBRARIAN/iX Administrator's Guide*

ORPHAN

Disables tracking of read-mode secondary files and copies of previous revisions of master/secondary files.

Restrictions

Owner

Menu Mode

Select the **Orphan** option from the **File** menu. A dialog appears allowing you to specify files, revision criteria, and options.

Command Mode Syntax

```
>OR[PHAN] filelist

     [ ;BATCH ]
     [ ;MEMO [ = memo−text ] ]
```

Parameters

filelist	A list of files, as described in *How to Refer to Files* at the beginning of this chapter.
BATCH	Processes the transaction in batch mode. Refer to *Batch Operations* at the beginning of this chapter.
MEMO = *memo−text*	Lets you include comments describing the current transaction. If you do not specify memo text, LIBRARIAN will invoke the configured editor so that you can enter the memo text.

Operation

ORPHAN allows you to stop tracking a secondary in read-mode or copies of previous revisions of master or secondary files. Once orphaned, a file is no longer tracked by LIBRARIAN. You can also **ORPHAN** copies of retained master and secondary files.

Examples

Orphan all of the files in the HOLDAREA group of your login account by typing:

MPE
```
>ORPHAN @.HOLDAREA
```

UNIX
```
>ORPHAN ./holdarea/*
```

OVERLAY

Replaces a tracked file with the contents of another tracked or untracked file.

Restrictions

Application Manager; file owner (file system security enforced)

Menu Mode

Select the **Overlay** option from the **File** menu. A dialog appears allowing you to specify files, revision criteria, and options.

Command Mode Syntax

```
>OV[ERLAY] filelist WITH filelist

     [ ;BATCH ]
     [ ;MEMO [ = memo-text ] ]
```

Parameters

filelist	A list of files, as described in *How to Refer to Files* at the beginning of this chapter.
BATCH	Processes the transaction in batch mode. Refer to *Batch Operations* at the beginning of this chapter.
MEMO = *memo–text*	Lets you include comments describing the current transaction. If you do not specify memo text, LIBRARIAN will invoke the configured editor so that you can enter the memo text.

Operation

OVERLAY allows you to replace LIBRARIAN files with any other files, including files that were created outside the LIBRARIAN system. With **OVERLAY**, the source file is neither moved nor renamed.

OVERLAY provides a convenient way to bypass normal restrictions. However, its use is restricted to Application Managers and LIBRARIAN Managers. **OVERLAY** should be used only in unusual circumstances and not as a regular procedure. You cannot overlay a master file that has an associated delta file.

Examples

Replace all of the files in the SOURCE location of the development area with all of the files in the SOURCE location of the MASTER area by typing:

MPE
```
>OVERLAY @.SOURCE.DEVEL WITH @.SOURCE.MASTER
```
UNIX
```
>OVERLAY /usr/devel/source/* WITH /usr/master/source/*
```

Replace the files in the AP-FILES fileset with the files in the AP-FILES fileset in your login area by typing:

MPE
```
>OVERLAY %AP-FILES WITH %AP-FILES AT @.@
```

UNIX
```
>OVERLAY %AP–FILES WITH %AP–FILES AT ./*
```

OVERLAY (continued)

Examples, continued

Replace all the files in your login with files in the same location on system DST068 by typing:

```
>OVERLAY @ WITH DST068 :
```

Replace all secondary copies of files in the MFG-FILES fileset with the master files in MFG-FILES by typing:

```
MPE   >OVERLAY %MFG-FILES AT @:@.@.@ WITH %MFG-FILES
UNIX  >OVERLAY %MFG–FILES AT *:/* WITH %MFG–FILES
```

PCRECEIVE

Transfers files from a host computer to a PC using Reflection.

Restrictions

None. Restricted by MPE security.

Menu Mode

Not available in menu mode. Press F2 for command mode and then use **PCRECEIVE** as described below.

Command Mode Syntax

```
>PCR[ECEIVE] filename [ FROM systemfile [;L] ]
     [ { ASCII | BINARY } ]
     [ { APPEND | DELETE } ]
```

Parameters

filename	Name of file to be transferred including drive/path (if desired). No wildcards are permitted.
FROM *hostfile*	Name of file on the host machine. If omitted, the host filename is the same as the PC filename.
L	Maintains label information. Note: this option is required when receiving a file which must be returned to the host with all of its label information intact.
ASCII | **BINARY**	Transfers the file as a standard ASCII file or binary file. If this is omitted, the last file transfer value is used.
APPEND	Appends a file to an existing file with the same name.
DELETE	Deletes an existing PC file with the same name.

Operation

PCRECEIVE and its parameters may not exceed 80 characters, and must be typed on one line.

Examples

In the following example, the host file HOSTFILE is received on the B drive of the PC.

```
>PCRECEIVE B:MY.TXT FROM HOSTFILE.PUB ASCII
```

Related Information

See Reflection's **RECEIVE** command

PCSEND

Transfers files from a PC to a host computer using Reflection.

Restrictions

None. Restricted by MPE security.

Menu Mode

Not available in menu mode. Press F2 for command mode and then use **PCSEND** as described below.

Command Mode Syntax

```
>PCS[END] filename [ TO systemfile [;P] [;L] [;F] [;Q] ]
     [ { ASCII | BINARY } ]
     [ DELETE ]
     [ REC = number ]
```

Parameters

filename	The name of the file to be transferred including drive/path (if desired). No wildcards are permitted in this name.
TO *hostfile*	The name of the file on the host machine. If omitted, the host filename is the same as the PC filename.
P	Instructs Reflection to completely replace the existing file, including file characteristics — this is very important if the existing file is smaller than the one being transferred.
L	Maintains label information. Note: This option is required for programs transferred to the PC and then back to the host.
F	Produces a fixed-length file.
Q	Transfers files in the correct format for QEDIT. Note: Without the Q parameter, files are treated as ASCII files.
ASCII or **BINARY**	Transfers the file as a standard ASCII file or as a binary file.
DELETE	Deletes an existing host file with the same name.
REC = *number*	Transfers using a particular record size.

Examples

Send a fixed-length binary PC file to the HP 3000 by typing:

```
>PCSEND SAMP.DAT TO GRAPH;F BINARY
```

Related Information

See Reflection's **SEND** command

PERFORM

Executes a predefined step for a file or fileset.

Restrictions

Step Authorization

Menu Mode

Select **Steps...** from the **File** menu and choose the step you want to perform from the menu. A dialog appears allowing you to specify files, revision criteria, and options.

Command Mode Syntax

```
> step { .route [ .appl ] }
       { .project         }

[ filelist [ (mode) ] [ TO tolocation ] ] [,[ filelist [ (mode) ] [ TO tolocation ] ] ...]

     [ ;ANNOTATE [ = NODELETE ] ]
     [ ;APPEND ]
     [ ;AUTOUPDATE ] or [ ;NOAUTOUPDATE ]
     [ ;BATCH ] or [ ;NOBATCH ]
     [ ;BRANCH ] or [ ;NOBRANCH ]
     [ ;COMPRESS ] or [ ;NOCOMPRESS ]
     [ ;CONDITIONAL = maxcon ] or [ ;NOCONDITIONAL ]
     [ ;CREATE = [ GROUP ][ ,ACCOUNT ][ ,CREATOR ] ]
     [ ;CREATOR = creator | unixowner ]
     [ ;DBONLY] or [ ;NODBONLY]
     [ ;DECOMPRESS ] or [ ;NODECOMPRESS ]
     [ ;ERRORS = maxerr ] or [ ;NOERRORS ]
     [ ;EXTERNAL ]
     [ ;FROM = EXTERNAL ]
     [ ;GROUP = unixgroup ]
     [ ;INPROGRESS ] or [ ;NOINPROGRESS ]
     [ ;KEEP ]
     [ ;LOCKWORD = lockword ]
     [ ;MASTER = pending-master ]
     [ ;MEMO [ = memo-text ] ] or [ ;NOMEMO ]
     [ ;MERGE  = mergelist ]
     [ ;NOMOVE ]
     [ ;NOSEARCH ]
     [ ;NOTIFY ]
     [ ;OLDDATE ]
     [ ;ONLINE ]
     [ ;ORPHAN ] or [ ;NOORPHAN ]
     [ ;OWNER = owner ]
     [ ;PERMISSIONS = unixpermissions ]
     [ ;PROJECT = project ]
     [ ;PUSHREAD ]
     [ ;READ ] or [ ;WRITE ]
     [ ;RENUMBER ]
     [ ;REPLACE ]
     [ ;RESET ]
```

PERFORM (continued)

Command Mode Syntax, continued

```
[ ;RETAIN ] or [ ;NORETAIN ]
[ ;REV = rev ]
[ ;TAG = tage ]
[ ;TO = EXTERNAL ]
[ ;VERIFY ] or [;NOVERIFY]
[ ;VIOLATIONS = maxvio ] or [ ;NOVIOLATIONS ]
```

Parameters

step	Name of the step to execute. If you do not specify a route or application, you may be asked to select from a list of steps of the same name.
route	The route to which the step belongs. If this parameter is omitted, LIBRARIAN tries to uniquely identify the route. if LIBRARIAN cannot identify it, you will be prompted for the route. You can use *project* in place of *route*.
project	Name of the project to be associated with files for the step. If this parameter is omitted and projects are defined, you are prompted for the project. The *project* can be used in place of the *route*. Log records for this step include the project name (the special value $NP may be used to denote No Project).
appl	Application to which the step belongs. If this parameter is omitted, LIBRARIAN tries to uniquely identify the application. If LIBRARIAN cannot identify it, you are prompted for the application. In batch mode, always use the *step.route.application* designation to avoid ambiguous steps.
filelist (mode)	List of files, as described in *How to Refer to Files* at the beginning of this chapter. You can optionally append any part of the filelist with (R) for read, or (W) for write. *(mode)* overrides the global step parameter, READ or WRITE.
TO *tolocation*	Destination location for the filelist. The syntax is:

```
MPE          { [system:] file [.group [.acct ]] }
        TO   {                                  }
UNIX         { [system:] /[path.../] file       }
```

The equal sign (=) wildcard can be entered in any filename element. It takes the same value as the corresponding element in the filelist (source file). Also, a single at sign (@) can be used. It has the same meaning as the equal sign (=) wildcard. The *tolocation* can also be an Edit Mask, as described earlier in this chapter.

PERFORM (continued)

Parameters, continued

ANNOTATE	Creates an annotated copy of source code based on the delta file, showing lines that were inserted and deleted for each revision, including date/time, project, and user. Annotation appears as commented code appropriate for the programming language set for the file (see **SET LANGUAGE**). If you attempt to **ANNOTATE** files for which deltas are not being stored, LIBRARIAN will notify you that a violation has occurred.
APPEND	Appends data from the source file to the end of the destination file. For MPE files, EOF plus new data cannot exceed LIMIT.
AUTOUPDATE	Adds new master files processed by this step to filesets if they match descriptors defined on the Auto Filesets (AF) screen. This parameter is valid only for secondary-to-master steps.
BATCH	Processes the transaction in batch mode. Refer to "Batch Operations" at the beginning of this chapter.
BRANCH	Forces a branch from the most recent trunk revision, or from any leaf that terminates a branch. The default is to increment the leaf count. **BRANCH** forces the addition of a branch/leaf pair.
COMPRESS	Compresses the new destination file(s).
CONDITIONAL = *maxcon*	Sets the maximum number of conditional files allowed for the transaction. If the maximum number is exceeded, it terminates the operation.
DBONLY	Records a move or copy operation in the LIBRARIAN database without physically moving or copying file(s). You are, however, responsible for updating "timestamps" in the LIBRARIAN database with the **RESET TIMESTAMP** command.
CREATE	Makes any directories on the destination path that do not already exist.
CREATE=GROUP, ACCOUNT, and/or **CREATOR**	Creates MPE group, account, and/or creator if it does not exist. If none of **GROUP, ACCOUNT,** or **CREATOR** is specified, all will be created if necessary.

Note

H–P's Security Monitor product enables system managers to prevent the creation of users, groups, or accounts without passwords. If **CREATE** is specified or configured on a step, and this Security Monitor feature is in use, users will receive the error message "Password required for user".

Note

All LIBRARIAN user IDs are case sensitive.

PERFORM (continued)

Parameters, continued

CREATOR = *creator*	Overrides the MPE creator/UNIX owner of the new file. For MPE, it can be a specific MPE user. For UNIX, it can be a specific UNIX user login, or decimal user ID found in /etc/passwd. Alternatively, you can use one of the following:	
	!USERID	Uses current LIBRARIAN user.
	!LOGON	Uses current login user.
	!KEEP	Uses MPE creator/UNIX owner of the file being replaced.
CREATOR = *unixowner*	Changes the owner of the new destination file. It can be a specific UNIX user login or decimal user ID found in /etc/passwd.	
	!USERID	LIBRARIAN user ID as creator.
	!LOGON	Current UNIX login user as creator.
	!KEEP	Creator of the file being replaced.
DECOMPRESS	Decompresses the new destination file(s).	
ERRORS = *maxerr*	Sets the maximum number of authorization errors (conditional and violation files) allowed for this transaction. If the number exceeds the maximum, it aborts the operation. This parameter is especially useful in batch mode.	
EXTERNAL	Tracks both source and destination files as external to LIBRARIAN's operating environment (e.g., PCs, machines not linked to LIBRARIAN). You are responsible for ensuring that external movement completes successfully.	
FROM = EXTERNAL	Specifies that the source file is located on a computer where LIBRARIAN is not implemented.	
GROUP = *unixgroup*	Changes the group ID of the new destination file. It can be a specific UNIX group name or decimal group ID found in /etc/group.	
INPROGRESS	Updates tracking to reflect the write mode secondary without making a new copy in the secondary location. Only available on master-to-secondary steps.	
KEEP	Instructs LIBRARIAN not to copy a file if the destination file already exists. When you use batch mode or **QUIET ON /DISPLAY,** the default is to purge existing destination files. Otherwise, the default is to prompt. **KEEP** overrides both of these defaults.	
LOCKWORD = *lockword*	Assigns a lockword to the destination file created by the operation. You can specify !**KEEP** as the lockword, which instructs LIBRARIAN to use the lockword from the old destination file.	

PERFORM (continued)

Parameters, continued

MASTER=*pending–master*	For new files that you introduce as secondaries, the master parameter allows you to specify the corresponding master name (pending master). If you are introducing more than one file, you can use an edit mask as described at the beginning of this chapter. You do not need to use this parameter if the step includes rules for how to determine the pending master name automatically.
MEMO = *memo–text*	Lets you include comments describing the current transaction. If you do not specify memo text, LIBRARIAN will invoke the configured editor so that you can enter the memo text.
MERGE =*mergelist*	Allows you to specify revisions to merge when performing master-to-secondary (checkout) steps. Revisions from the mergelist are merged into the file being checked out. Revisions in the mergelist should be separated by commas. ■ The minus (-) prefix is used to exclude a revision's changes from being included in the merge. ■ The exclamation point (!) prefix is used to merge only the changes from a specific revision without including changes that led up to that revision. If you attempt to **MERGE** files for which deltas are not being stored, LIBRARIAN will notify you that a violation has occurred. If conflicts are found when the **MERGE** option is used on checkout, the file is flagged with an exception that must be reset before continuing. Exceptions are shown in **VERIFY** format 11 with a code of MC.
NOAUTOUPDATE	Overrides the **AUTOUPDATE** default parameter value for the step.
NOBATCH	Overrides the **BATCH** default parameter value for the step. It is equivalent to **ONLINE**.
NOBRANCH	Blocks the use of the **BRANCH** option. The LIBRARIAN or the Application Manager can prevent users from using **BRANCH** by coding **NOBRANCH** in a macro, or by configuring this option as an additional step parameter on the STO screen. The **NOBRANCH** option prevents the branch prompt prevents a user from using the **BRANCH** parameter. Additionally, **NOBRANCH** prevents the user from checking out a WRITE mode copy of a previous revision of a file.
NOCOMPRESS	Overrides the **COMPRESS** default parameter value for the step.
NOCONDITIONAL	Allows no conditional files for the transaction. Equivalent to **CONDITIONAL**=0.
NODBONLY	Prevents the use of the **DBONLY** parameter. The LIBRARIAN or Application Manager can enforce this action by coding **NODBONLY** in a macro, or configuring this option as an additional step parameter on the STO screen.

PERFORM (continued)

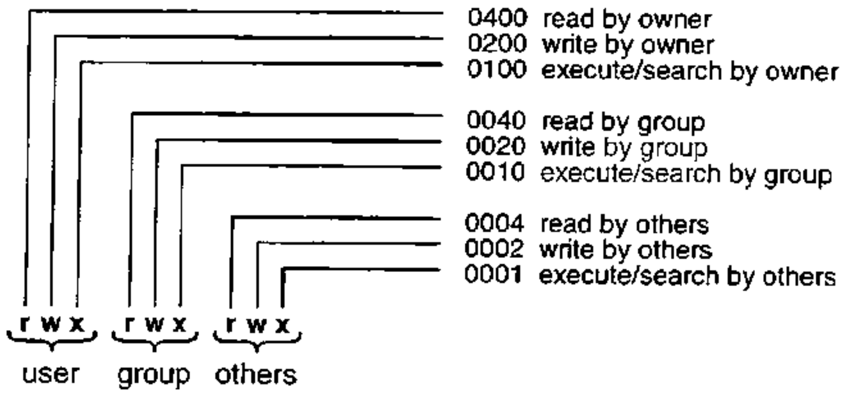
Parameters, continued

NODECOMPRESS	Overrides the **DECOMPRESS** default value for the step.
NOERRORS	Allows no errors for the transaction. Equivalent to **ERROR**=0.
NOINPROGRESS	Blocks use of the **INPROGRESS** parameter. The LIBRARIAN or Application Manager can enforce this action by coding **NOINPROGRESS** in a macro, or configuring this option as an additional step parameter on the STO screen.
NOMEMO	Overrides the **MEMO** default parameter value for the step.
NOMOVE	Authorizes the request but performs no actual file operation. Used for simulating possible transactions.
NOORPHAN	Overrides the **ORPHAN** default value for the step
NORETAIN	Overrides the **RETAIN** default value for the step. It is equivalent to **REPLACE.**
NOSEARCH	Disables search of previous version locations. Applies only to steps with forward versioning records as defined on the Forward Versioning (FV) screen.
NOTIFY	When checking in a file, this parameter instructs LIBRARIAN to notify other users who are working on different versions of the same file.
NOVERIFY	Overrides the **VERIFY** default value for the step.
NOVIOLATIONS	Allows no violations for the transaction. It is equivalent to **VIOLATIONS**=0.
OLDDATE	Instructs LIBRARIAN not to update the database with the new modification timestamp of the destination file. When specified, the destination file appears modified.
ONLINE	Overrides the **BATCH** default value for the step. It is equivalent to **NOBATCH.**
ORPHAN	Creates files that are not tracked by LIBRARIAN.
OWNER = *owner*	Sets the LIBRARIAN owner of the file to a specific LIBRARIAN user ID.

PERFORM (continued)

Parameters, continued

PERMISSIONS = *unixpermissions*	Changes permissions of the new destination file according to the *unixpermissions* mode, which is a number constructed from the following mode bits:



PROJECT = *project*	Filters the requested *fileref* by project association. Only files associated with this project will be authorized; others will appear as violations. For master files in *fileref,* retained revisions of those masters that are associated with the project will be authorized as well as master files.
PUSHREAD	Replaces a write mode file with a read mode file as an exception. Usage of this parameter requires LIBRARIAN Manager or Application Manager capability, unless it is defined as a step parameter. If **PUSHREAD** is used on checkin and there is another secondary in write mode, the owner is informed via a memo that an exception has taken place. This user must reset the exception (**RESET EXCEPTION** command) before proceeding with the next step.
READ	Assigns read mode to all files created during this step. Note that the filelist access mode specified overrides this parameter.
RENUMBER	Renumbers a numbered file when reconstructed from a delta file.
REPLACE	Overrides the RETAIN default value for the step. It is equivalent to NORETAIN.
RESET	Resets the modification timestamp in the database after the copy has been completed. Generally it is used with the **MODIFIED** parameter.
RETAIN	Retains any destination file that would otherwise be replaced. It applies only to files tracked by LIBRARIAN.
REV = *rev*	Authorizes the specified revision of the master file(s) requested, whether that revision happens to he the master itself or a retained revision. Not valid for secondary files.

PERFORM (continued)

Parameters, continued

TAG = *tag*	Filters the requested *fileref* by tag value. Only files with this tag value will be authorized; others will appear as violations. For master files in *fileref*, retained revisions of those masters that have this tag value will be authorized as well as master files.
TO = EXTERNAL	Specifies that the destination file is located on a computer where LIBRARIAN is not installed.
VERIFY	Identifies files that have been modified by comparing the modification timestamp in the file label with the timestamp stored when it was last created by LIBRARIAN. If the file has been modified since it was moved into its current location, the file is not copied (violation).
VIOLATIONS = *maxvio*	Sets the maximum number of violations allowed for the transaction. If the number exceeds the maximum, it aborts the operation. Use this parameter to define batch mode operations on an all-or-nothing basis.
WRITE	Assigns write mode access to all files created during this step. If write mode access is not available for a file, it becomes conditional on read mode. Note that the filelist access mode specified overrides this parameter. Refer to *filelist* above.

Operation

PERFORM carries out a step that has been defined in the database. Once a step is defined, you can use it to perform file transactions within the defined limits of the step.

When issuing **PERFORM**, you can override or specify additional parameters if allowed by the step definition. You can specify a subset of the defined fileset. You can override selected default parameters for file retention, file compression, memo text, etc. For more information on defining steps, refer to the *LIBRARIAN/iX User's Guide*.

Examples

Perform the CHECKOUT step using the fileset, source location, target location, and default step parameters defined for that step by typing:

```
>CHECKOUT
```

Perform the CHECKOUT step on all files in the default SOURCE area defined for the step using defined target locations and default step parameters by typing:

MPE
```
>CHECKOUT @.SOURCE
```

UNIX
```
>CHECKOUT src/*
```

Examples, continued

Perform the CHECKOUT step on the SOURCE-FILES fileset, request write mode access for these files, and prompt for memo text by typing:

```
>CHECKOUT %SOURCE-FILES;WRITE;MEMO
```

Perform the CHECKOUT step for two filesets, request write mode access on one fileset, and read mode access on the other one by typing:

```
>CHECKOUT %SOURCE-FILES (W), %PROG-FILES (R)
```

Perform the CHECKOUT.DEVEL.AP step on the INPUT fileset (subset of the defined fileset for the step). Copy the files into the login location, and retain the same filenames as in the source location by typing:

```
>CHECKOUT.DEVEL.AP %INPUT TO =
```

Perform the SUBMIT step, retaining copies of any files that would otherwise be replaced and prompt for memo text. Identify the project as MRP-PROJECT. Assign write mode access to all new files. Do not perform the operation if any files are conditional on read mode.

```
>SUBMIT.MRP-PROJECT;RETAIN;MEMO;WRITE;NOCONDITIONAL
```

Perform the APPROVE step for three components of the step fileset, MFG-EXE, MFG-JCL, and MFG-SOURCE files, using the defined step values and default step parameters by typing:

```
>APPROVE %MFG-EXE, %MFG-JCL, %MFG-SOURCE
```

Perform the CHECKIN step for all files in the MFGDEVEL account. If any conditional or violation errors occur, abort the operation. If there are new files that match the auto fileset descriptors, add them to the defined fileset for the step. To do this, type:

```
>CHECKIN @.@.MFGDEVEL;NOERRORS;AUTOUPDATE
```

Perform the RELEASE.DISTRIBUTION.MFG step. If any file has been modified since it was moved to its current location, or if any other violations occur, abort the step. To do this, type:

```
>RELEASE.DISTRIBUTION.MFG ;VERIFY ;NOVIOLATIONS
```

Note Because a time delay exists between authorization and actual file movement, a file could be replaceable during authorization, but might still fail with an exclusive access violation when LIBRARIAN attempts to move the file. This situation is rare.

PERFORM (continued)

Examples, continued

To merge two branches with the current master, type:

```
>MFG-OUT MFG080S;MERGE=2.1.2,2.2.1
```

Related Information

See Chapter 2, "Getting Started with Basic Rules" in the *LIBRARIAN/iX Administrator's Guide*
Chapter 3, "File Transactions" in the *LIBRARIAN/iX User's Guide*

PRINT (XPRINT)

Prints a file on the screen or printer.

Restrictions

File owner/file system security for tracked files; File system security for untracked files.

Menu Mode

Select the **Print** option from the **File** menu. A dialog appears allowing you to specify files, revision criteria, and options.

Command Mode Syntax

```
>PR[INT] filelist

†   [ ;ANNOTATE [ = NODELETE ] ]
    [ ;BATCH ]
    [ ;CHAR ]
    [ ;LINES = first, last ]
    [ ;NOPAGE ]
    [ ;NUMBERED ]
    [ ;OFFLINE ]
```

† Not available with **XPRINT**

Parameters

filelist	A list of files, as described in *How to Refer to Files* at the beginning of this chapter.
ANNOTATE	Creates an annotated copy of source code based on the delta file, showing lines that were inserted and deleted for each revision, including date/time, user, and project. Annotation appears as commented code appropriate for the programming language set for the file (see **SET LANGUAGE** command). If you attempt to **ANNOTATE** files for which deltas are not being stored, LIBRARIAN will notify you that a violation has occurred. **NODELETE** — Shows only insertions. Deleted lines normally appear as commented text but are not shown when the **NODELETE** option is specified.
BATCH	Processes the transaction in batch mode. Refer to *Batch Operations* at the beginning of this chapter.
CHAR	Replaces unprintable characters with periods (.). This is the default for binary files. You should use this option for ASCII files with unprintable data.
LINES = *first, last*	Specifies the first and last line to print.
NOPAGE	Specifies that there should be no page breaks between files.

Parameters, continued

NUMBERED	Prints line numbers.
OFFLINE	Prints file(s) offline to device LP (formal file designator of SCANLIST or LIBOUT).

Operation

PRINT sends the contents of a file on the screen, printer, or other device. Use the **SCAN** command to search for text strings.

When a file is displayed at the terminal, you are prompted after each page whether or not you wish to continue—unless the **NOPAGE** option is requested or you are in QUIET mode. At this prompt, you can continue the listing by pressing RETURN. To terminate the listing enter the letter N. You may also enter a line number. If you enter a line number, the next page of the listing begins with that line of the file.

XPRINT operates on untracked files enforcing normal file system security, unless the user has X capability.

For annotation, you can specify a set of escape sequences for printer initialization and bold print in a file called PRINTESC. The first line of the file is for printer initialization. The second line of the file is for bold print. The third line of the file is for normal print. You can use the **ESC** keyword in the file to indicate the **Escape** character. In addition, you can use the keyword **UNDERLINE** rather than an escape sequence for bold. You can use file equations for the PRINTESC file.

Note New files that you introduce with a step are stamped as intermediate revisions of the latest version for an application (VCOUNT = 1).

The following example sets escape sequences for the HP laser jet printer:

```
ESC(s10H
ESC(s3B
ESC(s0B
```

You can print QEDIT files (FILECODE = 111) if you are using QEDIT Version 4.L.55 or higher.

Examples

Print all files in the FINANCE fileset on the line printer by typing:

```
>PRINT %FINANCE;OFFLINE
```

Related Information

See **SCAN**
Chapter 3, "File Transactions" in the *LIBRARIAN/iX User's Guide*

PROJECT

Allows you to maintain project definitions, status, user authorizations, and display project information from the command line. You can use the **PROJECT** command in batch jobs. The same functions are also provided on the PJ, PS, PA, and PI screens.

Command Mode Syntax

```
>PROJECT { project-name|@ } . appl-id  [ ;ADD  |  SHOW [=status] | DELETE ]

      [ ;DESCRIPTION = description ]
      [ ;ROUTE = route-alias ]
      [ ;USERS = [ { user-list | * } ]
      [ ;MANAGER = user-id ]
      [ ;STATUS= status ]
      [ ;USER-STATUS or | USTATUS = user-status ]
      [ ;DATE = date-requested ]
      [ ;PRIORITY = priority ]
      [ ;ESTIMATE = estimate ]
      [ ;REQUESTOR = requestor ]
      [ ;DEPARTMENT = department ]
```

Parameters

project-name	The name of the project, up to 12 characters. LIBRARIAN uses this name as the name of the project fileset as well.
@	Lists all projects for an application.
appl-id	The name of the application to which the project belongs. If SET APPLICATION is in effect, you can omit the application name.
ADD	Adds a new project to the database.
SHOW [=*status*]	Displays all project data for a particular project, or lists all projects for an application. When listing projects, you can provide one of the following status selection values:

AL	All projects
AA	All active projects
AI	All inactive projects
AO	All open projects
AC	All closed projects
CC	Closed to CHECKOUT
CL	Closed to all steps
DC	Documented
FP	Flush Pending
FL	Flushed
OP	Opened
RO	Reopened

PROJECT (continued)

Parameters, continued

DELETE	Deletes project data from the database if the status is FL (Flushed).
DESCRIPTION=	A description of the project, up to 150 characters. This description will appear on menus and reports. The default description is blank.
ROUTE = *route–alias*	The route for which the project is valid. Use the at sign (@) to indicate that the project is valid for all routes in the application. The default is @ or the route set with SET ROUTE.
USERS = *user–list*	Lists users authorized to work on this project, optionally preceded with a plus sign (+). Use a minus sign (–) to delete users from a previously defined list of users. An empty list deletes all users. A value of "*" indicates that no authorization is required. Use commas to define a list of users.
MANAGER = *user*	The user responsible for the project. The user must have Project, Application, or LIBRARIAN manager capability. The default is the current user.
STATUS = status	Changes the status of a project to the given value if appropriate. For new projects, the default project status is OPEN. The following are valid status values: DOC[UMENTED] or DC OP[EN] CL[OSE] CLOSE–TO–CHECKOUT or CC RE[OPEN] or RO FL[USH]
USER–STATUS = *status* USTATUS = *status*	A free–form user–defined status value.
DATE= *date–requested*	For documentation only, up to 8 characters.
PRIORITY = *priority*	For documentation only, up to 4 characters.
ESTIMATE = *est*	For documentation only, up to 8 characters.
REQUESTOR = *requestor*	For documentation only, up to 20 characters.
DEPARTMENT = *department*	For documentation only, up to 20 characters.

PROJECT (continued)

Operation

The **PROJECT** command mirrors the functionality of the PJ, PS, PA, and PI screens.

Users can refer to the projects defined with this command to qualify steps. If projects are required for a route and none is specified, then LIBRARIAN displays a menu of projects for the user.

When you define a new project, LIBRARIAN also creates a fileset with the name which is the same as the project name. This fileset is a private user fileset owned by the Project manager.

To add a new project, use the ADD parameter. To display a project, use SHOW. To delete a project, use DELETE. If ADD, SHOW, and DELETE are not specified, the command changes the project data (if the project exists).

If you do not want to require specific project authorization, specify "USERS=*". However, if you do require project authorization, use the "USERS" parameter to identify the authorized users separated by commas.

If you want to document the project but do not wish to have it available for use immediately, set the STATUS to DOCUMENTED. DOCUMENTED status is only available for new projects. The default status is OPEN.

You can only delete a project if it has been flushed. To flush a project, set the status to FLUSH PENDING and then run the FLUSHLOG utility.

If you want information about a specific project, you need to indicate both the project and the application name and use the SHOW parameter. To list projects in an application, use "@" in place of the project name. To select a subset of an application's projects, you can include a status filter with the SHOW parameter.

Examples

Associate the file ABC1000S with the project SR1234 and adds its master the project fileset:

```
>SET ABC1000S.SOURCE PROJECT=SR1234
```

Add a new project called SR2035 for application HR:

```
>PROJECT SR2035.HR;ADD;DESCRIPTION="Fix 401K year-to-date
          calculation";ROUTE=HR-MAINT;PRIORITY=HIGH
```

Modify the estimated time for project SR2035:

```
>PROJECT SR2035.HR;ESTIMATE=5
```

Change the status of project SR2035 to CLOSED:

```
>PROJECT SR2035.HR;STATUS=CLOSED
```

Show all active projects in application HR:

```
>PROJECT @.HR;SHOW
```

Related Information

See Chapter 6, "Projects" in the *LIBRARIAN Administrator's Guide*

PURGE (XPURGE)

Removes disk files from the system and LIBRARIAN tracking data, if it exists in the LIBRARIAN database.

Restrictions

File owner for tracked files. File system security for untracked files.

Menu Mode

Select the **Purge** option from the **File** menu. A dialog appears allowing you to specify files, revision criteria, and options.

Command Mode Syntax

```
>PU[RGE] filelist

    [ ;BATCH ]
†   [ ;DELTA ]
    [ ;MEMO [ = memo-text ] ]
    [ ;RESTORE ] or [ ;NORESTORE ]
†   [ ;RETAIN ]
    [ ;REVISION = revision/ALL]
    [ ;VERIFY ]
```

† Not available with **XPURGE**

Parameters

filelist	A list of files, as described in *How to Refer to Files* at the beginning of this chapter.
BATCH	Processes the transaction in batch mode. Refer to *Batch Operations* at the beginning of this chapter.
DELTA	Purges the delta file associated with a master file, if a master file is being purged.
MEMO = *memo-text*	Lets you include comments describing the current transaction. If you do not specify memo text, LIBRARIAN will invoke the configured editor so that you can enter the memo text.
RESTORE	Causes the most recent revision on the trunk to become the master after the current master is purged.
RETAIN	Retains the files as generation files (or delta).
REVISION = *revision*/**ALL**	If you specify **ALL**, purges all revisions of a master file; otherwise, purges a specific revision.
NORESTORE	Causes the current master to be purged and removed from the master fileset. Previous revisions are still available using the **VERIFY** command and step operations if you refer to the specific file; the file will not be found if you use a wildcard or master fileset authorization.
VERIFY	Only authorizes files that have not been modified since LIBRARIAN created the file in its current location.

PURGE (continued)

Operation

All LIBRARIAN references to the purged file or fileset, including associated secondary copies and file detail records, are removed from the database, but log transactions are not deleted. If the purged file was a master file, any secondaries linked to it are no longer tracked.

PURGE automatically retains base revision files. To purge a base revision, use the **VERSION** command to obsolete it and then the FLUSH utility to purge it.

If neither **;RESTORE** nor **;NORESTORE** option is specified, the user will be prompted to select one of the options. **;RESTORE** is the default when this prompt is made, and when QUIET ON or QUIET DISPLAY are in effect.

Note When you use **PURGE** to remove the last master, related secondary, or retained file, the master filename will automatically be removed from the project fileset.

XPURGE operates on untracked files only and enforces normal file system security, unless the user has X capability.

Examples

Purge all copies of the FINANCE fileset currently in the GL account and the finance development area by typing:

MPE
```
>PURGE %FINANCE AT @.@.GL, %FINANCE AT @.@.FINDEVEL
```

UNIX
```
>PURGE %finance AT /gl/*, %finance AT /finance/devel/*
```

Retain backup copies of the files before you purge the entire GL-FILES fileset by typing:

```
>PURGE %GL-FILES ;RETAIN
```

Purge all the retained files in all groups in your login account in batch mode by typing:

MPE
```
>PURGE G#######.@ ;BATCH
```

UNIX
```
>PURGE .g#######;BATCH
```

Purge all of the files in all of the groups of the MFG account on the DEVELCPU system by typing:

MPE
```
>PURGE DEVELCPU:@.@.MFG
```

UNIX
```
>PURGE develcpu:/mfg/*
```

Related Information

See **CLEANDB**

QUIET

Sets the level of interaction between LIBRARIAN and the user, including prompts and standard messages.

Restrictions

None

Menu Mode

Select the **Settings...Quiet Mode** option from the **User** menu. A dialog appears allowing you to specify files, revision criteria, and options.

Command Mode Syntax

$$>\text{QUIET}\ \left\{\begin{array}{l}\textbf{ON}\\ \textbf{OFF}\\ \textbf{DISPLAY}\end{array}\right\}$$

Parameters

ON	Uses default answers to LIBRARIAN prompts.
OFF	Prompts for answers. LIBRARIAN waits for you to answer each authorization module question. The default is QUIET OFF.
DISPLAY	Uses default answers to LIBRARIAN prompts and suppresses standard messages and prompts.

Operation

With **QUIET OFF**, the authorization module prompts you for answers to several questions. With **QUIET ON**, all authorization module questions appear, but you are not prompted for the answers. Instead, the LIBRARIAN program supplies default answers. The defaults are:

Question	Default Answer
SHOW violations, authorizations, etc.	YES
EXPLAIN?	YES
CONTINUE?	YES
WHICH ONE DO YOU WANT?	NONE OF THE ABOVE
DO YOU WANT CONDITIONAL FILES?	NO

With **QUIET ON**, existing destination files are purged (unless the **KEEP** option is in effect), and displays are always listed.

QUIET DISPLAY operates like **QUIET ON**, except that standard messages and prompts are also suppressed. Only messages that show the result of a file operation and errors are shown.

QUIET (continued)

Examples

Set the program to use default answers for prompts, and to minimize the display of messages by typing:

```
>QUIET DISPLAY
```

Related Information

See **XEQ ECHO**

R1 (or R7)

Executes Reflection commands within LIBRARIAN.

Restrictions

None

Menu Mode

Not available in menu mode. Press **F2** for command mode and then use **R1** (or **R7**) as described below.

Command Mode Syntax

>R1 [*Reflection command*]

or

>R7 [*Reflection command*]

Parameters

Reflection command	Any Reflection command.

Operation

The **R1** (or **R7**) command allows you to execute Reflection commands within LIBRARIAN. For information on Reflection command syntax and usage, refer to your *Reflection Command Language Manual*.

Examples

Execute the DOS **DIR** command with Reflection in LIBRARIAN by typing:

```
>R1 DIR
```

REDO

Presents a previous command entry for editing.

Restrictions

None

Menu Mode

Not available in menu mode. Press F2 for command mode and then use **REDO** as described below.

Command Mode Syntax

>RED[O] [*cmdid*]

Parameters

cmdid	Command to re-execute. You can specify one of the following:
	-n The *n*th command before most current one, where *n* is a number in the command line stack relative to most recent command, which is -1.
	m The number *m* in the command line stack. The number *m* is absolute (not relative).
	string The most recent command beginning with the text *string*.
	If you do not specify a value, LIBRARIAN presents the previous command for editing.

Operation

Use **REDO** to correct errors in the last command issued. LIBRARIAN'S **REDO** functions similarly to MPE's **REDO** command. **REDO** includes the editing character ">," to append to the end of the command, and "U" to undo the last edit. For more information, refer to the *MPE Commands Reference Manual.* **REDO** recognizes the following editing characters:

I <*text*>	Insert text starting at position of I.
D <*...D*>	Delete characters at position of D.
R <*text*>	Replace text starting at position of R.
U	Undo the last edit.
>	Append to the end.

REDO (continued)

Examples

Edit the previous command by typing:

```
>REDO
```

Edit command 20 by typing:

```
>REDO 20
```

Related Information

See **DO**
LISTREDO

RELEASE

Removes the MPE security from a file or fileset.

Restrictions

File owner

Menu Mode

Select the **Release** option from the **File** menu. A dialog appears allowing you to specify files, revision criteria, and options.

Command Mode Syntax

```
>REL[EASE] filelist

     [ ;BATCH ]
     [ ;MEMO [ = memo-text ] ]
```

Parameters

filelist	A list of files, as described in *How to Refer to Files* at the beginning of this chapter.
BATCH	Processes the transaction in batch mode. Refer to *Batch Operations* at the beginning of this chapter.
MEMO = *memo-text*	Lets you include comments describing the current transaction. If you do not specify memo text, LIBRARIAN will invoke the configured editor so that you can enter the memo text.

Operation

RELEASE suspends the MPE security for all files or filesets specified. Use **RELEASE** to release security when an operation requires access to files that are not normally available. Only the LIBRARIAN Manager and the Application Manager can release files belonging to a different login account or MPE user.

You can reinstate the security with the **SECURE** command.

LIBRARIAN does not record the changes made by **RELEASE** in the database.

Examples

Release all of the master files in the REL1.0 version of the FINANCE fileset by typing:

```
>RELEASE REL1.0 OF %FINANCE
```

Release all of the files in the SOURCE group of the system DST068 by typing:

```
>RELEASE DST068 :@.SOURCE
```

Release all of the files in the PROGRAM-FILES fileset by typing:

```
>RELEASE %PROGRAM-FILES
```

RELEASE (continued)

Examples, continued

Release all secondary copies of files in the PROGRAM-FILES fileset that currently reside on the system DST068 by typing:

```
>RELEASE %PROGRAM-FILES AT DST068 :@.@.@
```

Related Information

See **SECURE**

If you are running UNIX, see the chmod manual entry for equivalent functionality.

RENAME (XRENAME)

Renames a file or the files in a fileset.

Restrictions

File owner for tracked files, file system security for untracked files.

Menu Mode

Select the **Rename** option from the **File** menu. A dialog appears allowing you to specify files, revision criteria, and options.

Command Mode Syntax

```
>REN[AME] filelist [ TO tolocation ]

      [ ;BATCH ]
      [ ;COMPRESS ]
      [ ;CREATE = [ GROUP ] [ ,ACCOUNT ] [ ,CREATOR ] ]
      [ ;CREATOR = creator | unixowner ]
      [ ;DECOMPRESS ]
      [ ;GROUP = unixgroup ]
      [ ;KEEP ]
      [ ;LOCKWORD = lockword ]
      [ ;MEMO [ = memo-text ] ]
†     [ ;OLDDATE ]
†     [ ;OLDNAME ]
      [ ;PERMISSIONS = unixpermissions ]
†     [ ;VERIFY ]
```

† Not available with **XRENAME**.

Parameters

filelist	A list of files, as described in *How to Refer to Files* at the beginning of this chapter.
TO *tolocation* MPE UNIX	The destination location for the filelist. The syntax is: TO [*system:*] *file* [*.group* [*.acct*]] TO [*system:*] /[*path...*/] *file* The equal sign (=) wildcard can be entered in any filename element. It takes the same value as the corresponding element in the filelist (source file). Also, a single at sign (@) can be used. It has the same meaning as the equal sign (=) wildcard. The *tolocation* can also be an Edit Mask, as described earlier in this chapter. If *tolocation* is omitted, your current login location is assumed.

RENAME (continued)

Parameters, continued

BATCH	Processes the transaction in batch mode. Refer to "Batch Operations" earlier in this chapter.
COMPRESS	Compresses the new destination file(s).
CREATE	Makes any directories on the destination path that do not already exist.
CREATE=GROUP, ACCOUNT, and/or **CREATOR**	Creates MPE group, account, and/or creator if it does not exist. If none of **GROUP, ACCOUNT,** or **CREATOR** is specified, all will be created if necessary.

Note H–P's Security Monitor product enables system managers to prevent the creation of users, groups, or accounts without passwords. If **CREATE** is specified or configured on a step, and this Security Monitor feature is in use, users will receive the error message "Password required for user".

CREATOR = *creator* Overrides the MPE creator/UNIX owner of the new file. For MPE, it can be a specific MPE user. For UNIX, it can be a specific UNIX user login, or decimal user ID found in /etc/passwd. Alternatively, you can use one of the following:

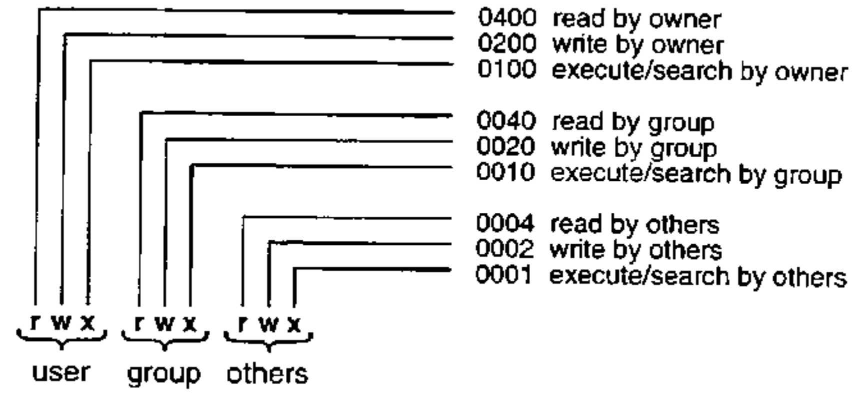
!USERID	Uses current LIBRARIAN user.
!LOGON	Uses MPE current login user.
!KEEP	Uses MPE creator/UNIX owner of the file being replaced.

Note All LIBRARIAN user IDs are case-sensitive.

DECOMPRESS	Decompresses the new destination file(s).
GROUP = *unixgroup*	Changes the group ID of the new destination file. It can be a specific UNIX group name or decimal group ID found in /etc/group.
KEEP	Instructs LIBRARIAN not to copy a file if the destination file already exists. When you use batch mode or **QUIET ON**, the default purges existing destination files. When you use session mode, the default prompts you. **KEEP** overrides both of these defaults.
LOCKWORD = *lockword*	Assigns a lockword to the destination file created by the operation. You can specify !KEEP as the lockword, which instructs LIBRARIAN to use the lockword from the old destination file.

Parameters, continued

MEMO = *memo–text*	Lets you include comments describing the current transaction. If you do not specify memo text, LIBRARIAN will invoke the configured editor so that you can enter the memo text.
OLDDATE	Leaves timestamp in the database as is and sets the create and modify dates of the destination file to be the same as the from file.
OLDNAME	Uses the original name of the retained file in the new destination location. System, group, and account are established by the *tolocation* specification.
PERMISSIONS = *unixpermissions*	Changes the permissions of the new destination file according to the *unixpermissions* mode, which is a number constructed from the following mode bits:



VERIFY	Identifies files that have been modified by comparing the modification timestamp in the file label with the timestamp stored when it was last created by LIBRARIAN. If the file has been modified since it was moved into its current location, the file is not copied (violation).

Operation

RENAME changes the identity of the specified files. When master files are renamed, LIBRARIAN updates associated secondaries and retained files to point to the new name. For MPE files, if you rename a master file, the original lockword is retained; if you rename a secondary file, your personal lockword is placed on the file.

When you rename a KSAM file, the associated key filename does not change.

Note When you use **RENAME** to remove the last master, related secondary, or retained master associated with a project, the old filename will automatically be removed and the new one will be added.

XRENAME operates on untracked files only and enforces normal file system security, unless the user has X capability.

Examples

Change the group name for a list of files currently in your login group and account by typing:

```
>RENAME FILEA, FILEB, FILEC, FILED TO =.DATA
```

```
>RENAME filea, fileb, filec, filed TO data/=
```

Change the group name for all files currently in the SUZY account by typing:

```
>RENAME @.SUZY TO =.SUZETTE
```

```
>RENAME /usr/suzy/* TO suzette/=
```

Rename all of the files in the REL2.0 version of the %MFG-FILES fileset to files with their original names to another area by typing:

```
>RENAME REL2.0 OF %MFG-FILES TO =.REL2GRP ;OLDNAME
>RENAME REL2.0 OF %MFG-FILES TO /apps/mfg/rel2group/=;OLDNAME
```

Related Information

See **MOVE**

RESET * (**)

Releases the reference to a set of files frozen with the **SET *** command.

Restrictions

None

Menu Mode

Select **Settings...Save Star** from the **User** menu and then choose the **OFF** option to release the files saved when you turned this option **ON**.

Command Mode Syntax

MPE
```
>RESET *
```

UNIX
```
>RESET **
```

Parameters

None

Operation

The asterisk as a file reference indicates files successfully created or processed in the previous transaction. This asterisk reference can be frozen, using the **SET *** command, so that * always refers to the files of a particular transaction.

Use **RESET *** if you previously used **SET *** to freeze the files referred to by the asterisk (*), and you now want to use the asterisk to refer to files in the last transaction.

Example

Reset the last transaction back reference by typing:

MPE
```
>RESET *
```

UNIX
```
>RESET **
```

Related Information

See **SET ***

RESET (APPLICATION)

Resets the default application established with the **SET APPLICATION** command.

Restrictions

None

Menu Mode

Select **Settings...Application** from the **User** menu and then choose **<none>** from the menu.

Command Mode Syntax

```
>RESET APPLICATION
```

Parameters

None

Operation

Use **RESET (APPLICATION)** to reset the default application established with **SET (APPLICATION)**.

Examples

Reset the current default application by typing:

```
    >RESET APPLICATION
```

Related Information

See **SET (APPLICATION)**

RESET (EXCEPTION)

Removes the exception flag on a file.

Restrictions

File Owner

Menu Mode

Select the Set...**Exception reset** option from the **File** menu. A dialog appears allowing you to specify files, revision criteria, and options.

Command Mode Syntax

```
>RESET filelist EXCEPTION

    [ ;BATCH ]
    [ ;MEMO [ = memo-text ] ]
```

Parameters

filelist	A list of files, as described in *How to Refer to Files* at the beginning of this chapter.
BATCH	Processes the transaction in batch mode. Refer to *Batch Operations* at the beginning of this chapter.
MEMO = *memo−text*	Lets you include comments describing the current transaction. If you do not specify memo text, LIBRARIAN will invoke the configured editor so that you can enter the memo text.

Operation

When a read mode copy is checked in using **PUSHREAD** and a write mode copy exists, the write mode copy is flagged with an exception. LIBRARIAN also records exceptions when it encounters conflicts during a merge operation, or restores an older version of a file that is currently checked out. Use the **VERIFY** command (Format 11) to see which exception is set on a file.

Any step attempted on these files fails with an **EXCEPTION** violation. The purpose of this violation is to require the owner of the write mode copy to acknowledge the exception by using **RESET (EXCEPTION)**. This command clears the exception flag, so that a step can be performed against it.

Examples

Reset the exception flag for file ABC1000S by typing:

```
>RESET ABC1000S EXCEPTION
```

Related Information

See **PERFORM**
    VERIFY

RESET (PROJECT)

Resets a default project set for a LIBRARIAN session with the **SET PROJECT** command.

Restrictions

None

Menu Mode

Select **Settings...Project** from the **User** menu and then choose **<none>** from the menu.

Command Mode Syntax

```
>RESET PROJECT
```

Parameters

None

Operation

Use **RESET (PROJECT)** to reset the default project established with **SET (PROJECT)**.

Examples

Set the ABC project by typing

```
    >RESET PROJECT ABC
```

Related Information

See **SET (PROJECT)**

RESET (ROUTE)

Resets the default route established with the **SET ROUTE** command.

Restrictions

None

Menu Mode

Select **Settings...Route** from the **User** menu and then choose **<none>** from the menu.

Command Mode Syntax

```
>RESET ROUTE
```

Parameters

None

Operation

Use **RESET (ROUTE)** to reset the default route established with **SET (ROUTE)**.

Examples

Reset the current default route by typing:

```
    >RESET ROUTE
```

Related Information

See **SET (ROUTE)**

RESET (TIMESTAMP)

Resets a file's modification timestamp in the LIBRARIAN database to reflect the current timestamp recorded by the file system.

Restrictions

Application Manager

Menu Mode

Select the Set...**Timestamp reset** option from the **File** menu. A dialog appears allowing you to specify files, revision criteria, and options.

Command Mode Syntax

```
>RESET filelist TIMESTAMP

     [ ;BATCH ]
     [ ;MEMO [ = memo-text ] ]
```

Parameters

filelist	A list of files, as described in *How to Refer to Files* at the beginning of this chapter.
BATCH	Processes the transaction in batch mode. Refer to *Batch Operations* at the beginning of this chapter.
MEMO = *memo-text*	Lets you include comments describing the current transaction. If you do not specify memo text, LIBRARIAN will invoke the configured editor so that you can enter the memo text.

Operation

After each successful LIBRARIAN file operation, the Modification Date/Time values are extracted from the file label of the destination file and stored in the LIBRARIAN database.

If you use the **VERIFY**, **MODIFIED** or **UNMODIFIED** parameter in a file operation, you can use the **RESET TIMESTAMP** command to update the values stored in the database so that the file would no longer appear modified. Use the **TOUCH** command to make a file appear modified.

Examples

Reset the timestamp on all files in MFG on system DST068 by typing:

MPE
```
>RESET DST068:@.@.MFG TIMESTAMP
```

UNIX
```
>RESET dst068:/appl/mfg/* TIMESTAMP
```

RESET (TIMESTAMP) (continued)

Examples , continued

Reset the timestamps for FILEA on systems DST037, DST044, and DST068 by typing:

MPE
```
>RESET DST037:FILEA, DST044:FILEA, DST068:FILEA TIMESTAMP
```

UNIX
```
>RESET dst037:filea, dst044:filea, dst068:filea TIMESTAMP
```

Related Information

See **TOUCH**

RESTORE

Reconstructs a previous revision of a file making it the most current. For masters, it automatically retains the current file before replacing it with the older version.

Restrictions

LIBRARIAN Manager, Operator for master files, or File Owner for secondary files

Menu Mode

Select the **Restore** option from the **File** menu. A dialog appears allowing you to specify files, revision criteria, and options.

Command Mode Syntax

```
>REST[ORE] filelist

     [ ;BATCH ]
     [ ;COMPRESS ]
     [ ;CREATE = [ GROUP ] [ ,ACCOUNT ] [ ,CREATOR ] ]
     [ ;CREATOR = creator | unixowner ]
     [ ;DELTA ]
     [ ;DECOMPRESS ]
     [ ;KEEP ]
     [ ;MEMO [ = memo-text ] ]
     [ ;RENUMBER ]
     [ ;VERIFY ]
```

Parameters

filelist	A list of files, as described in *How to Refer to Files* at the beginning of this chapter. The **REVISION** and **GCOUNT** parameters are particularly useful with this command.
BATCH	Processes transactions in batch mode. Refer to *Batch Operations* at the beginning of this chapter.
COMPRESS	Compresses the new destination file(s).
UNIX **CREATE**	Makes any directories on the destination path that do not already exist.
MPE **CREATE=GROUP, ACCOUNT,** and/or **CREATOR**	Creates MPE group, account, and/or creator if it does not exist. If none of **GROUP, ACCOUNT,** or **CREATOR** is specified, all will be created if necessary.

Note

H–P's Security Monitor product enables system managers to prevent the creation of users, groups, or accounts without passwords. If **CREATE** is specified or configured on a step, and this Security Monitor feature is in use, users will receive the error message "Password required for user".

Parameters, continued

CREATOR = *creator*	Overrides the MPE creator/UNIX owner of the new file. For MPE, it can be a specific MPE user. For UNIX, it can be a specific UNIX user login, or decimal user ID found in /etc/passwd. Alternatively, you can use one of the following:

!USERID	Uses current LIBRARIAN user.
!LOGON	Uses MPE current login user.
!KEEP	Uses MPE creator/UNIX owner of the file being replaced.

Note All LIBRARIAN user IDs are case-sensitive.

DELTA	Reconstructs the current master file from the corresponding delta file. Use this when you suspect the master file has been corrupted.
DECOMPRESS	Decompresses the new destination file(s).
KEEP	Instructs LIBRARIAN not to copy a file if the destination file already exists. When you use batch mode or **QUIET ON**, the default purges existing destination files. When you use session mode, the default prompts you. **KEEP** overrides both of these defaults.
MEMO = *memo–text*	Lets you include comments describing the current transaction. If you do not specify memo text, LIBRARIAN will invoke the configured editor so that you can enter the memo text.
RENUMBER	Renumbers a numbered file when reconstructed from a delta file.
VERIFY	Identifies files that have been modified by comparing the modification timestamp in the file label with the timestamp stored when it was last created by LIBRARIAN. If the file has been modified since it was moved into its current location, the file is not copied (violation).

Operation

RESTORE allows you to restore a retained file to its original name and location. A restored master file becomes a master file once again, and a restored secondary becomes a secondary. You specify a previous revision of a file by using the **REVISION** or **GCOUNT** parameters (see *filelist*).

If the retained file was compressed and the current master file is not compressed, the file is decompressed automatically after it is restored.

The restored file replaces any file that may currently exist in the destination location after being retained, unless you use the **KEEP** parameter.

Operation, continued

For secondary files, you can specify the name(s) and LIBRARIAN will automatically restore the latest retained file based on the modification timestamp. To use this feature, enter the following:

```
>RESTORE secondary–name
```

Examples

Replace the current production files in the FINANCE fileset with the REL3.0 version.

```
>RESTORE REL3.0 OF %FINANCE
```

Alternatively, restore a retained secondary file by specifying the secondary filename. LIBRARIAN automatically restores the latest retained file, based on the modification timestamp. To use this feature, type:

```
>RESTORE myfile
```

Restore files retained in the last transaction by typing:

MPE
```
>RESTORE *
```
UNIX
```
>RESTORE **
```

This syntax can be used in an XEQ procedure to automatically back out a partial transaction, as shown in the following example:

```
DISTRIBUTE !XEQLIST;RETAIN
IF LIBFAIL>0 THEN
    CONTINUE
    RESTORE *
    MAIL OPERATOR;DISTRIBUTION Failed!
ENDIF
```

In batch mode, restore and decompress the latest retained files in the source area. If a destination file already exists, do not replace it. Add memo text describing the transaction.

MPE
```
>RESTORE @.SOURCE; GCOUNT=–1; DECOMPRESS;KEEP;BATCH;MEMO
```
UNIX
```
>RESTORE ~/src/*; GCOUNT=–1; DECOMPRESS; KEEP; BATCH; MEMO
```

Related Information

See **VERIFY**
Revision History Report (RRH10) in Chapter 6, "Reports"

RETRY

Sets the number of times to attempt linking to a remote system that is not responding, and the interval between attempts.

Restrictions

None

Menu Mode

Not available in menu mode. Press F2 for command mode and then use **RETRY** as described below.

Command Mode Syntax

```
>RET[RY] [ max-retries [ ,retry-interval ] ]
```

Parameters

max-retries	The number of times for LIBRARIAN to attempt linking to remote systems. Default: 0.
retry-interval	The number of minutes between each attempt. Default: 0.

Operation

RETRY sets the number of times LIBRARIAN attempts to link to a remote system after the first unsuccessful attempt and the number of minutes between each attempt.

If you do not specify any parameters, the current **RETRY** settings are displayed.

Setting a maximum **RETRY** value has an effect only when LIBRARIAN detects that the remote system is not responding or has rejected the login request. Typically, this happens when the remote system is down, the network has not been started, or the remote session limit is too low to provide a login. No retry attempts are made if the error is an invalid device specification.

Examples

Attempt to link to the remote system ten times, with thirty-minute intervals between attempts by typing:

```
>RETRY 10,30
```

Display the current **RETRY** settings by typing:

```
>RETRY
```

Related Information

See Chapter 3, "File Transactions" in the *LIBRARIAN/iX User's Guide*

SCAN (XSCAN)

Scans files for strings of text, and optionally replaces matching text.

Restrictions

File owner/file system security for tracked files; file system security for untracked files.

Menu Mode

Select the **Scan** option from the **Tools** menu. A dialog appears allowing you to specify files, search text, revision criteria, and options.

Command Mode Syntax

```
>SCA[N] filelist
    [ ;ASK ]
    [ ;BATCH ]
    [ ;CHAR ]
    [ ;COLUMNS = [ firstcol ] [ ,lastcol ] ]
    [ ;IGNORE | ;UPSHIFT ]
    [ ;LINES = [ firstline ] [ ,lastline ] ]
    [ ;LISTFILE = filename ]
    [ ;MATCHES = nummatches ]
    [ ;MEMO [ = memo–text ] ]
    [ ;NOPAGE ]
    [ ;NOSHOW ]
    [ ;NUMBERED ]
    [ ;OFFLINE ]
    [ ;REVERSE ]
    [ ;SHOW ]
    [ ;SUMMARY ]
    [ ;TEXT = search [ /replace [,...] ]
    [ ;WINDOW = [ linesbefore ] [ ,linesafter ] ]
```

Parameters

filelist	A list of files, as described in *How to Refer to Files* at the beginning of this chapter.
ASK	Asks the user to confirm each replacement.
BATCH	Processes the transaction in batch mode. Refer to *Batch Operations* at the beginning of this chapter.
CHAR	Displays unprintable characters as periods (.). This is the default for binary files. You should use this option for ASCII files with unprintable data.
COLUMNS	Restricts the range of columns to scan, where *firstcol* is the beginning column (Default: 1), and *lastcol* is the ending column (default is record length).
IGNORE *or* **UPSHIFT**	Ignores upper and lower-case.

SCAN (continued)

Parameters, continued

LINES	Restricts the range of lines scanned, where *firstline* is the beginning line (Default: 1), and *lastline* is the ending line (Default: file size).
LISTFILE = *filename*	Writes the names of files with at least one match to a listfile, where *filename* is the name of a listfile in your login directory.
MATCHES = *nummatches*	Specifies the number of matches required before terminating the search. By default, it scans the entire file.
MEMO = *memo−text*	Lets you include comments describing the current transaction. If you do not specify memo text, LIBRARIAN will invoke the configured editor so that you can enter the memo text. Memos are only logged if **SCAN** is used with the *replace* option.
NOPAGE	Specifies that there should be no page breaks.
NOSHOW	Instructs LIBRARIAN not to display matching lines.
NUMBERED	Displays line numbers for matching lines, in conjunction with the SHOW parameter.
OFFLINE	Directs scan listing offline.
REVERSE	Selects lines in which the search string is not found.
SHOW	Displays matching lines (default).
SUMMARY	Displays a summary of matching files
TEXT = *search [/replace [, search / replace]], ...]*	Specifies search and replace strings containing a maximum of 50 characters each. You can specify any number of search/replace pairs, separated by commas. Enclose the search string (*search*) in quotes only if it includes commas, semicolons, slashes, or blanks. The following pattern-matching wildcards can appear anywhere in the search string.

@	match any number of any character
?	match any single alphanumeric character
#	match any single numeric character
*	match any single alphabetic character
^	match any single blank character
!	match any single character
{...}	match a character in the set of characters enclosed in braces (e.g., {ABC}). You can refer to a maximum of ten character sets in a single command.

SCAN (continued)

Parameters, continued

All pattern-matching wildcards (except for @) can be followed by +, indicating a match for one or more occurrences. A minus sign (-) following the wildcard, indicates zero or more occurrences. For example, the search string #+ informs LIBRARIAN to search for a string containing one or more consecutive numeric characters.

The following characters can be used at the beginning and end of search strings, respectively:

[	match string at beginning of line only.
]	match string at end of line only.

The backslash (\) can precede any pattern-matching character and itself to indicate a literal match.

Each search string can have an associated replace string (*replace*) with a maximum of 50 characters. Use a slash (/) between search and replace strings. Replace strings should be enclosed in quotes if they include commas, semicolons, slashes, or blanks.

You can use the following variables for replacement:

!NEXTG	substitutes next generation count (GCOUNT + 1) for this file
!NEXTV	substitutes next version count (VCOUNT + 1) for this files
!VCOUNT	substitutes version count for this file
!GCOUNT	substitutes generation count for this file
!VERSION	substitutes version name for this file
!REVISION	substitutes revision ID for this file

Edit masks can also be used to transform the search string into a new replace string. Enclose the edit mask in parentheses. Edit masks follow the same conventions as edit masks for filenames. See the beginning of this chapter for more information.

You can append text to the end of lines where a match was found by putting a plus (+) in front of the replace string.

You can delete lines on which a match was found by specifying **!DELETE** in place of a replace string.

WINDOW Sets the number of lines to show before and after matching lines, where *linesbefore* sets the number of lines to show before a matching line (Default: 0). *linesafter* sets the number of lines to show after a matching line (Default: 0).

SCAN (continued)

Operation

Use **SCAN** to search files for a particular string (or strings) of characters. You can replace strings of text in files containing a maximum of 4096 bytes in length.

Pattern-matching characters can be used in the search string. The **OFFLINE** option generates a listing on the LP device. This listing can be redirected with a file equation for LIBOUT or SCANLIST.

You can use **SCAN** to review the contents of a file on your terminal. If you omit the **TEXT** parameter, you will see 23 lines of the file at a time. At the end of each page you are prompted to continue. The next page is displayed by pressing RETURN. If you type a record number, a page of text is displayed starting at that record number.

A JCW, LIBMATCHES, is available after **SCAN** operations complete. This JCW shows the number of files in which a string match was found.

XSCAN operates on untracked files enforcing normal file system security, unless the user has X capability.

You can print QEDIT files (FILECODE = 111) if you are using QEDIT Version 4.L.55 or higher.

Examples

Search all files in the SOURCE area for the FINANCE application for the string $INCLUDE. Display each line of text for all occurrences found to match without sensitivity to case by typing:

MPE
```
>SCAN @.SOURCE.FINANCE;TEXT=$INCLUDE;IGNORE
```

UNIX
```
>SCAN /appl/finance/source/*;TEXT=/#include;IGNORE
```

Search the FINANCE-FILES fileset for files that include the string VERSION:= '2.*xx*', where *xx* are any numeric values that match without sensitivity to case and allow any number of blanks in between. After the first match quit and generate a listfile with the names of files for which a match was found. To do this, type:

```
>SCAN %FINANCE-FILES;TEXT=VERSION!:=!'2.##';IGNORE;MATCHES=1; &
   LISTFILE=VERS2LIST
```

Search and replace by typing:

```
>SCAN %FINANCE-FILES ;TEXT="REL#.##"/"REL!VERSION!NEXTV"
```

Related Information

See Chapter 5, "Printing, Comparing, and Scanning Files" in the *LIBRARIAN/iX User's Guide*

SCOMPARE (XSCOMPARE)

Accesses the **S/COMPARE** utility (a proprietary product of the ALDON Computer Group). Must be installed on the LIBRARIAN/iX server.

Restrictions

File owner/file system security for tracked files; file system security for untracked files.

Menu Mode

Select the **Compare** option from the **Tools** menu. A dialog appears allowing you to specify files, revision criteria, and options. The method of comparison is determined by the compare method that you can change on the **Settings** window from the **User** menu.

Command Mode Syntax

```
>SCO[MPARE] filelist TO filelist

     [ ;BATCH ]
     [ ;BEGINPOS = column-number ]
     [ ;ENDPOS = column-number ]
     [ ;BLANKS ] or [ ;NOBLANKS ]
     [ ;COMMENTS = YES | NO | NOBLANK ]
     [ ;COPYLIB ] or [ ;NOCOPYLIB ]
     [ ;EDIT ] or [ ;NOEDIT ]
     [ ;FORMAT = [ NOOFFSET ], [ NOBOX ] ]
†    [ ;FROMVCOUNT = fromvcount ] or [ ;FROMGCOUNT = fromgcount ]
†    [ ;TOVCOUNT = tovcount ] or [ ;TOGCOUNT = togcount ]
†    [ ;FROMREV = revid ]
†    [ ;TOREV = revid ]
     [ ;LANGUAGE = langid ]
     [ ;MATCH = beforenum [ ,afternum ] ] or [ ;NOMATCH = beforenum [ ,afternum ] ]
     [ ;MEMBER = refmem [ ,cmpmem ] ]
     [ ;OFFLINE [ = [ device ] [ , [ priority ] [ , [ copies ] [ , [ env ] ] ] ] ] ] or [ ;ONLINE ]
     [ ;PACK ] or [ ;NOPACK ]
     [ ;PAGESIZE = pgsize ]
     [ ;RANGE = [ first record ] [ ,last record ] [ ,RELATIVE ] ]
```

† Not available with **XSCOMPARE**

Parameters

filelist	A list of files, as described in *How to Refer to Files* at the beginning of this chapter.
TO filelist	Specifies the reference files to be used in comparisons.
BATCH	Processes the transaction in batch mode. Refer to *Batch Operations* at the beginning of this chapter.

Parameters, continued

BEGINPOS = *column-number*	Sets the left-most column to compare, where *column-number* is the left column.
ENDPOS = *column number*	Sets the right-most column to compare, where *column-number* is the right column.
BLANKS	Comparison does not ignore blank lines.
NOBLANKS	Comparison ignores blank lines.
COMMENTS=*setting*	Includes or excludes comments from a comparison. *setting* may be one of:

NO	excludes all comments
YES	compares all comments
NOBLANKS	compares all except blank comments.

A blank comment is one that contains only spaces or comment delimiters between the columns specified by **BEGINPOS** and **ENDPOS.**

COPYLIB	Treats sequential files as copy libraries.
NOCOPYLIB	Treats KSAM files as flat files.
EDIT	Creates an editor USE file containing the differences between the reference file and the compare file.
NOEDIT	Instructs S/COMPARE not to create a USE file containing the differences between the files.
FORMAT=*type*	Allows you to select **NOOFFSET, NOBOX**, or both as the format type.
FROMVCOUNT = *fromvcount*	Uses compare files with a version count equal to *fromvcount* This parameter must have a positive value and be used in conjunction with the *versionid* parameter.
FROMGCOUNT = *fromcount*	Uses compare files with a generation count equal to *fromgcount*. This parameter can have either a positive or negative value.

A negative value describes the generation equal to the current generation minus the **GCOUNT**. For example, GCOUNT=-2 specifies the file two generations earlier than the current generation.

TOVCOUNT = *tovcount*	Same as **FROMVCOUNT** described above, but applies to reference files.
TOGCOUNT = *togcount*	Same as the **FROMGCOUNT** described above, except defined for the **TO** *filelist*.

SCOMPARE (continued)

Parameters, continued

FROMREV = *revid*	Specifies which revision to retrieve as the compare file.
TOREV = *revid*	Same as **FROMREV**, but applies to the reference file(s).
	The remaining parameters have the same value as the corresponding parameter in the **SET** command of the S/COMPARE product, except where noted.
LANGUAGE = *langid*	Indicates the language of the source files to be compared. Possible languages values include:

BASIC	NONE	RPG
COBOL	PASCAL	SPL
FORTRAN	POWERHOUSE	TEXT
JCL	TRANSACT	

MATCH	Controls the display of matching lines, where *beforenum* is before the first number and *afternum* is after the last number.
NOMATCH	Excludes some on all of the matching list from the output report.
MEMBER	Specifies the copy library members to compare.
OFFLINE	Directs output to a line printer. You can specify the print device, output priority for number of copies, and environment filename for listing.
ONLINE	Directs output to your terminal.
PACK	Treats consecutive spaces as a single space.
NOPACK	Does not compress consecutive spaces in records to a single space.
PAGESIZE = *pgsize*	Sets the number of lines per printed page.
RANGE	Limits the comparison to those records between the first record and the last record. If **RELATIVE**, these are relative record numbers.

Operation

SCOMPARE accesses the S/COMPARE product providing it is installed on the system.

SCOMPARE supports most of the parameters of the S/COMPARE **COMPARE** and **SET** commands.

SCOMPARE (continued)

Operation, continued

Additionally, **SCOMPARE** provides you with an online display of file differences using video enhancements, and greater flexibility when specifying files. For example, with the **SCOMPARE** command you can:

1. Compare filesets using logical fileset identifiers.
2. Compare groups of physical files using wildcards.
3. Compare files without specifying the actual filenames, using **AT** and version/ revision specifications instead.

SCOMPARE's output also provides LIBRARIAN file information (as comments), including file IDs, file type, versions, and generations of the files being compared. If the files are compressed, **SCOMPARE** automatically decompresses them temporarily before the compare operation.

XSCOMPARE operates on untracked files enforcing normal file system security, unless the user has X capability.

Examples

Compare the REL3.0 and REL4.0 versions of the AP-FILES fileset by typing:

```
>SCOMPARE REL3.0 OF %AP-FILES TO REL4.0 OF %AP-FILES
```

The following example compares the current and the REL1.0 versions of the SFILES fileset. The language of the source code is Pascal and the output device is LP.

```
>SCOMPARE %SFILES TO REL1.0 OF %SFILES ;LANGUAGE=PASCAL ;OFFLINE=LP
```

The following example compares the current version of the LOGFILES fileset to the secondary copy of the fileset in the SQUDEVEL account. The language for the source code is Pascal. Show the ten lines matching before and after a mismatch. The output device is LP.

```
>SCOMPARE %LOGFILES TO %LOGFILES AT @.@.SQUDEVEL &
;LANGUAGE=PASCAL;MATCH=10,10;OFFLINE=LP
```

Related Information

See Chapter 5, "Printing, Comparing, and Scanning Files" in *LIBRARIAN/iX User's Guide.*

SECURE

Turns on the MPE security for a file or fileset.

Restrictions

File owner

Menu Mode

Select the **Secure** option from the **File** menu. A dialog appears allowing you to specify files, revision criteria, and options.

Command Mode Syntax

```
>SEC[URE] filelist

    [ ;BATCH ]
    [ ;MEMO [ = memo-text ] ]
```

Parameters

filelist	A list of files, as described in *How to Refer to Files* at the beginning of this chapter.
BATCH	Processes the transaction in batch mode. Refer to *Batch Operations* at the beginning of this chapter.
MEMO = *memo-text*	Lets you include comments describing the current transaction. If you do not specify memo text, LIBRARIAN will invoke the configured editor so that you can enter the memo text.

Operation

SECURE reinstates normal MPE file access security that was previously suspended by the **RELEASE** command. You can also secure files on remote systems.

LIBRARIAN does not record changes made by the **SECURE** command in the database.

Examples

Secure all of the master files in the FINANCE fileset by typing:

```
    >SECURE %FINANCE
```

Secure all of the files in the FINANCE fileset that were retained two generations earlier by typing:

```
    >SECURE %FINANCE ;GCOUNT=-2
```

Secure all files in the SOURCE group of your login account on system DST068 by typing:

```
    >SECURE DST068:@.SOURCE
```

Secure all secondary copies in the PROGRAM-FILES fileset that reside on the current system by typing:

```
    >SECURE %PROGRAM-FILES AT @.@.@
```

SECURE (continued)

Related Information

See **RELEASE**

If you are running UNIX, see the **chmod** manual page for equivalent functionality.

SET * (**)

Freezes the asterisk reference to files in the last transaction for use in subsequent transactions.

Restrictions

None

Menu Mode

Select **Settings...Save Star** from the **User** menu and then choose **ON** to save the filenames from the last transaction.

Command Mode Syntax

MPE
```
>SET *
```

UNIX
```
>SET **
```

Operation

SET * allows you to save the transaction number associated with the previous transaction so that you can recall associated filenames at a later time. To do this, use * for MPE and ** for UNIX, as a way to refer to files. If the previous transaction had destination files, the destination filenames are used.

Release the asterisk (*) for redefinition by using the **RESET** * command.

When you exit the LIBRARIAN program, the files designated by **SET** * are not remembered. To save the fileset, create a listfile with the following commands:

```
>LMAINT
LM>OUTPUT * TO MYFILES
LM>EXIT
>EXIT
```

Examples

If you perform a CHECKIN transaction and then issue **SET** *, the system remembers the transaction so that it can search for the files created by the CHECKIN step. You can then reference them in another command (e.g., **DISTRIBUTE** *), even if you have performed transactions with other files.

MPE
```
>SET *
>DISTRIBUTE * TO SYS1:=
>DISTRIBUTE * TO SYS2:=
>DISTRIBUTE * TO SYS3:=
```

SET * (continued)

Examples, continued

UNIX

```
>SET **
>DISTRIBUTE ** TO SYS1:=
>DISTRIBUTE ** TO SYS2:=
>DISTRIBUTE ** TO SYS3:=
```

Related Information

See **RESET ***
Chapter 3, "Master Library" in the *LIBRARIAN/iX Administrator's Guide*
Chapter 3, "File Transactions" in the *LIBRARIAN/iX User's Guide*

SET (APPLICATION)

Sets a default application to resolve ambiguous steps and versions.

Restrictions

None

Menu Mode

Select **Settings...Application** from the **User** menu. A picklist appears to let you select an application.

Command Mode Syntax

```
>SET APPLICATION [applid]
```

Parameters

applid	The name of the application to set as the default.

Operation

Use **SET (APPLICATION)** to specify the default application for a LIBRARIAN session. By setting an application, LIBRARIAN can resolve certain ambiguities automatically (e.g., versions and steps). Using this command without specifying an application displays the current default application.

Examples

Set the ABC application by typing:

```
    >SET APPLICATION ABC
```

Related Information

See **RESET (APPLICATION)**

SET (EXPDATE)

Changes the expiration date for read mode secondaries and retained revisions.

Restrictions

File Owner

Menu Mode

Select the **Set...Expiration Date** option from the **File** menu. A dialog appears allowing you to specify an expiration date, files, revision criteria, and options.

Command Mode Syntax

$$>\text{SET } \textit{filelist} \textbf{ EXPDATE} = \left\{ \begin{array}{l} \textbf{TODAY} \\ \textbf{NONE} \\ \textit{expire date} \end{array} \right\}$$

```
    [ ;BATCH ]
    [ ;MEMO [ = memo-text ] ]
    [ ;REVISIONS = revision/ALL ]
```

Parameters

filelist	A list of files, as described in *How to Refer to Files* at the beginning of this chapter.	
EXPDATE=*expiration*	Indicates the date the file will expire. You must use one of the following keywords, or a specific date:	
	TODAY	Sets expiration date to the current date.
	NONE	Sets the file not to expire.
	expire-date	Sets expiration date to this date. You must use the date format defined for your system on the System Profile screen (SP).
BATCH	Processes the transaction in batch mode. Refer to *Batch Operations* at the beginning of this chapter.	
MEMO = *memo–text*	Lets you include comments describing the current transaction. If you do not specify memo text, LIBRARIAN will invoke the configured editor so that you can enter the memo text.	
REVISION = *revision*/**ALL**	If you specify **ALL**, sets expiration date of all revisions of a master file; otherwise, sets expiration date of a specific revision.	

Operation

SET EXPDATE assigns an expiration date to retained or read mode secondary files. Expired files are flushed the next time you run the FLUSH utility.

Master files are never flushed and do not accept an expiration date. Retained base revisions are not flushed until they are made obsolete, regardless of the expiration date. Write mode secondaries are also ignored by the FLUSH utility, regardless of expiration date, until the write mode has been relinquished.

The RFN10 and RFN20 reports list the files that will be flushed when FLUSH is next run.

SET (EXPDATE) (continued)

Examples

Set the expiration date to today for all of the retained files in your login group and account by typing:

MPE
```
>SET G#######.@.@ EXPDATE=TODAY
```

UNIX
```
>SET /*/.g####### EXPDATE=TODAY
```

Set no expiration date for the files in the SOURCE group of your login account by typing:

MPE
```
>SET @.SOURCE EXPDATE=NONE
```

UNIX
```
>set src/* expdate=none
```

Set the expiration date to 25 December 1991 for the files in the MYFILES fileset by typing:

```
>SET %MYFILES EXPDATE=12/25/91
```

Related Information

See **FLUSH**
VERIFY
Pre–Flush Notification Reports (RFN10/RFN20) in Chapter 6, "Reports"

SET (LANGUAGE)

Allows you to assign a programming language to files for use when LIBRARIAN annotates source code.

Restrictions

Application Manager

Menu Mode

Select the **Set...Language** option from the **File** menu. A dialog appears allowing you to specify a language, files, revision criteria, and options.

Command Mode Syntax

```
>SET filelist LANGUAGE = langid
    [ ;REVISIONS = revision/ALL ]
```

Parameters

filelist	A list of files, as described in *How to Refer to Files* at the beginning of this chapter.
LANGUAGE = *langid*	Indicates the programming language to assign to the source file(s). The language values include:

BASIC	NONE	RPG
COBOL	PASCAL	SPL
FORTRAN	POWERHOUSE	TEXT
JCL	TRANSACT	

REVISION = *revision*/**ALL**	If you specify **ALL**, sets language value of all revisions of a master file; otherwise, sets language value of a specific revision.

Operation

Use **SET (LANGUAGE)** to select your programming language. A source code file must be assigned a language type for annotation to work properly.

Examples

Assign the COBOL language to all of the files in Libprod Source area by typing:

MPE
```
>SET @.SOURCE.LIBPROD LANGUAGE=COBAL
```

UNIX
```
>SET /apps/src/libprod/* LANGUAGE=COBOL
```

SET (LOCKWORD)

Assigns a new lockword to a file or fileset.

Restrictions

File Owner

Menu Mode

Select the **Set...Lockword** option from the **File** menu. A dialog appears allowing you to specify a new lockword, files, revision criteria, and options.

Command Mode Syntax

```
>SET filelist LOCKWORD = lockword

     [ ;BATCH ]
     [ ;MEMO [ = memo-text ] ]
     [ ;REVISIONS = revision/ALL ]
```

Parameters

filelist	A list of files, as described in *How to Refer to Files* at the beginning of this chapter.
LOCKWORD = *lockword*	Indicates the new lockword to assign. It must conform to MPE naming conventions. If no lockword is specified, any current lockword is removed.
BATCH	Processes the transaction in batch mode. Refer to *Batch Operations* at the beginning of this chapter.
MEMO = *memo−text*	Lets you include comments describing the current transaction. If you do not specify memo text, LIBRARIAN will invoke the configured editor so that you can enter the memo text.
REVISION = *revision*/**ALL**	If you specify **ALL**, sets a lockword for all revisions of a master file; otherwise, sets a lockword for a specific revision.

Operation

SET LOCKWORD changes the current lockword of a file or fileset. **SET LOCKWORD** updates the system directory with the new lockword. If the lockword value is left blank, existing lockwords are removed.

Examples

Assign the lockword KEEPOUT to all of the files in the FINANCE fileset by typing:

```
>SET %FINANCE LOCKWORD=KEEPOUT
```

For all files in your login group and account, remove the current lockword by typing:

```
>SET @ LOCKWORD=PRESENT
```

SET (MODE)

Changes the access mode for secondary files.

Restrictions

Application Manager

Menu Mode

Select the Set...Mode option from the **File** menu. A dialog appears allowing you to specify a mode, files, revision criteria, and options.

Command Mode Syntax

$$>\text{SET}\ \textit{filelist}\ \textbf{MODE} = \left\{ \begin{array}{l} \textbf{WRITE} \\ \textbf{READ} \end{array} \right\}$$

```
    [ ;BATCH ]
    [ ;MEMO [ = memo-text ] ]
```

Parameters

filelist	A list of files, as described in *How to Refer to Files* at the beginning of this chapter.
MODE	Specifies the access mode to be assigned to the file, as one of the following:
WRITE	Assigns write mode access to file(s). A write mode secondary can replace its master.
READ	Assigns read mode access to file(s). A read mode secondary cannot replace its master.
BATCH	Processes the transaction in batch mode. Refer to *Batch Operations* at the beginning of this chapter.
MEMO = *memo-text*	Lets you include comments describing the current transaction. If you do not specify memo text, LIBRARIAN will invoke the configured editor so that you can enter the memo text.

Operation

SET MODE changes the current access mode of a file or fileset. Although you can use **SET MODE** to change the access mode, you are restricted by the access control level of the file. If a file has been checked out and modified but cannot replace the master file because it is in read mode, use **SET MODE** to change the access mode to write mode. However, if the file has serial access control and a write mode copy has already been checked out, change the other file to read mode before changing the modified file to write mode.

Examples

Assign write mode access to all secondary copies of files in the FINANCE fileset by typing:

MPE
```
>SET %FINANCE AT @.@.@ MODE=WRITE
```

UNIX
```
>SET %FINANCE AT /* MODE=WRITE
```

Assign write mode access to all files with names ending in S in the DEVEL group of the MFG account on system DST068 by typing:

MPE
```
>SET DST068:@S.DEVEL.MFG MODE=WRITE
```

UNIX
```
>SET dst068:/mfg/devel/*s MODE=WRITE
```

Change the access mode to read mode for secondary copies of files in the MFG-FILES fileset that are in your login group by typing:

MPE
```
>SET %MFG-FILES AT @.@ MODE=READ
```

UNIX
```
>SET %MFG–FILES AT ./* MODE=READ
```

Related Information

See **PERFORM**

SET (OWNER)

Changes the LIBRARIAN owner of a file.

Restrictions

Application Manager

Menu Mode

Select the **Set...Owner** option from the **File** menu. A dialog appears allowing you to specify a new owner, files, revision criteria, and options.

Command Mode Syntax

```
>SET filelist OWNER = userid

     [ ;BATCH ]
     [ ;MEMO [ = memo–text ] ]
```

Parameters

filelist	A list of files, as described in *How to Refer to Files* at the beginning of this chapter.
OWNER = *userid*	Specifies a valid LIBRARIAN user.
Note	All LIBRARIAN user IDs are case-sensitive.
BATCH	Processes the transaction in batch mode. Refer to *Batch Operations* at the beginning of this chapter.
MEMO = *memo–text*	Lets you include comments describing the current transaction. If you do not specify memo text, LIBRARIAN will invoke the configured editor so that you can enter the memo text.

Operation

SET OWNER changes the LIBRARIAN owner of a file.

Examples

Set the owner of a file, ABC100S, to LIBMGR by typing:

```
>SET ABC100S OWNER = LIBMGR
```

Related Information

See **PERFORM**

SET (PROCEDURE)

Instructs LIBRARIAN to catalog all procedures in a macro procedure file.

Restrictions

None

Menu Mode

Select **Load procedures** from the **Macros** menu to catalog procedures in a procedure file. Select **Unload procedures** from the **Macros** menu to unload all procedures.

Command Mode Syntax

```
>SET PROCEDURE TO [ filename ]
```

Parameters

filename	The name of a macro procedure file.

Operation

SET PROCEDURE reads through the macro procedure file specified, cataloging all procedures so they can be used as LIBRARIAN commands. To see the list of cataloged procedures, use the **HELP PROCEDURES** command. Use **SET PROCEDURE** without parameters to release access to procedure files previously set by this command.

Examples

Catalog macro procedures in the file MYPROC so they can be used as commands by typing:

```
>SET PROCEDURE TO MYPROC
```

Related Information

See Chapter 9, "Macros" in the *LIBRARIAN/iX User's Guide*

SET (PROJECT)

Sets a default project for the LIBRARIAN session so user does not need to select a project from the project menu. Additionally, it allows you to change the project to which a file currently belongs.

Restrictions

None

Menu Mode

Select the **Settings...Project** option from the **User** menu. A menu appears allowing you to select a project.

Command Mode Syntax

```
>SET [ filelist ] PROJECT [ [=] project-name ]
    [ ;REVISIONS = revision/ALL ]
```

Parameters

filelist	The file(s) for which you want to change the project.
project-name	The name of the project to be set as the default.
REVISION = *revision*/**ALL**	If you specify **ALL**, sets the project of all revisions of a master file; otherwise, sets the project of a specific revision.

Operation

To specify the default project for a LIBRARIAN session, do not include the *filename* parameter. If you use **SET** without the *project-name* parameter, the current default project will be displayed.

To change the project assigned to a secondary, master, or retained file, include the *filename* parameter. When used with a secondary or master file, it also removes the filename from the old project's fileset, and adds it to the new project's fileset. To perform this command, the user must

- have 'L' (librarian manager) capability,
- have 'A' (application manager) capability for the application, or
- be the owner of the file.

Examples

Set the ABC project by typing

```
>SET PROJECT ABC
```

Related Information

See **RESET (PROJECT)**

SET (ROUTE)

Sets a default route to resolve step/project name ambiguities.

Restrictions

None

Menu Mode

Select the **Settings...Route** option from the **User** menu. A menu appears to let you select a route.

Command Mode Syntax

```
>SET ROUTE [ routeid ]
```

Parameters

routeid	The name of the route to set as the default.

Operation

Use **SET (ROUTE)** to specify the default route for a LIBRARIAN session. By setting a route, LIBRARIAN can resolve step and project ambiguities automatically. Using this command without specifying any route causes the current default route to be displayed.

Examples

Set DEVEL as the default route by typing:

```
>SET DEVEL ROUTE
```

Related Information

See **RESET (ROUTE)**

SET (TAG)

Assigns a user-defined tag to a file or set of files.

Restrictions

Application Manager

Menu Mode

Select the **Set...Tag** option from the **File** menu. A dialog appears allowing you to specify a tag, files, revision criteria, and options.

Command Mode Syntax

```
>SET fileset TAG = [ tagid ]
    [ ;REVISIONS = revision/ALL ]
```

Parameters

filelist	A list of files, as described in *How to Refer to Files* at the beginning of this chapter.
TAG =*tagid*	A user-defined text string to be assigned to a file or set of files. *tagid* can be a maximum of twelve (12) characters.
REVISION = *revision*/**ALL**	If you specify **ALL**, sets tag value of all revisions of a master file; otherwise, sets tag value of a specific revision.

Operation

SET TAG accepts a user-defined tag that will be associated with any type of tracked file or set of files. This tag can be used to select files in any command that accepts filenames.

You can perform steps against files that have a given tag or verify files with a particular tag. The tag feature is for convenience, providing another way for you to organize and reference different configurations of files.

Tags allow you to mark subsets of applications with an identifier that can be used to distribute subsets of an application (e.g., a patch). Tags can be used to establish a baseline for a subset of an application; whereas, versions establish a baseline for the entire application.

Note

Since a **SET TAG** transaction is not logged in the audit database, you can not use * or ** after this command to refer to files in the last transaction.

SET (TAG) (continued)

Examples

The following example assigns the tag PATCH1 to a set of files in a project.

```
>SET %PROJECT1 TAG=PATCH1
```

Related Information

See Chapter 7, "Versions" in the *LIBRARIAN/iX Administrator's Guide*

SHOWLOG

Accesses the **SHOWLOG** transaction log reporting utility.

Restrictions

None

Menu Mode

Select the **Log...Showlog** option from the **Info** menu.

Command Mode Syntax

```
>SH[OWLOG]
```

Parameters

None

Operation

SHOWLOG provides direct access to the LIBRARIAN transaction log reporting subsystem, which consists of commands that allow users to inspect transaction log data. This subsystem allows you to select, format, and sort log database transaction records, and edit memo text entered for transactions.

Once you access the SHOWLOG utility, the program uses the >SHOWLOG prompt.

Examples

Access the SHOWLOG selection menu by typing:

```
>SHOWLOG
```

Related Information

See Chapter 4, "SHOWLOG Commands"

TOUCH (XTOUCH)

Updates the modification timestamp in the MPE file label with the current date and time.

Restrictions

File owner or MPE write access.

Menu Mode

Select the **Touch** option from the **File** menu. A dialog appears allowing you to specify files, revision criteria, and options.

Command Mode Syntax

```
>T[OUCH] filelist

     [ ;BATCH ]
     [ ;MEMO [ = memo-text ] ]
†    [ ;RESET ]
†    [ ;VERIFY ]
```

† Not available with **XTOUCH**

Parameters

filelist	A list of files, as described in *How to Refer to Files* at the beginning of this chapter.
BATCH	Processes the transaction in batch mode. Refer to *Batch Operations* at the beginning of this chapter.
MEMO = *memo−text*	Lets you include comments describing the current transaction. If you do not specify memo text, LIBRARIAN will invoke the configured editor so that you can enter the memo text.
RESET	Resets the modification timestamp in the database after the touch has been completed.
VERIFY	Identifies files that have been modified by comparing the modification timestamp in the file label with the timestamp stored when it was last created by LIBRARIAN. If the file has been modified since it was moved into its current location, the file is not touched (violation).

TOUCH (continued)

Operation

Generally, you need **TOUCH** when you are using the MAKE facility. By touching a target file, you make it up–to–date, so MAKE will not build it. By touching a dependency file, you make the file(s) on which it depends out–of–date, so that MAKE will rebuild it. **TOUCH** makes a touched file appear modified to MAKE and LIBRARIAN commands that use the **MODIFIED** and/or **VERIFY** parameters.

Note If you attempt to **TOUCH** a file currently being accessed, LIBRARIAN will issue a violation.

Examples

Make a file appear modified by typing;

```
>TOUCH MYFILE
```

Related Information

See **MAKE**

Chapter 8, "Using MAKE to Rebuild Applications" in the *LIBRARIAN/iX User's Guide*

If you are running UNIX, see the **touch** manual page for equivalent functionality.

UNLOCK

Releases files that are locked.

Restrictions

File Owner

Menu Mode

Select the **Unlock** option from the **File** menu. A dialog appears allowing you to specify files, revision criteria, and options.

Command Mode Syntax

```
>UN[LOCK] filelist
      [ ;BATCH ]
      [ ;MEMO [ = memo-text ] ]
```

Parameters

filelist	A list of files, as described in *How to Refer to Files* at the beginning of this chapter.
BATCH	Processes the transaction in batch mode. Refer to *Batch Operations* at the beginning of this chapter.
MEMO = *memo−text*	Lets you include comments describing the current transaction. If you do not specify memo text, LIBRARIAN will invoke the configured editor so that you can enter the memo text.

Operation

UNLOCK releases files that were previously placed on hold by the **LOCK** command.

Examples

Release all secondary copies of files in the FIN fileset that have not been modified since they were created by LIBRARIAN by typing:

MPE
```
>UNLOCK %FIN AT @.@.@.@;UNMODIFIED
```
UNIX
```
>UNLOCK %FINANACE AT *:/*;UNMODIFIED
```

Release all of the locked files that are in your login by typing:

MPE
```
>UNLOCK @
```
UNIX
```
>UNLOCK ./*
```

Related Information

See **LOCK**

UPDATE

Updates a read mode secondary with the current associated master.

Restrictions

LIBRARIAN Manager, Application Manager, or File Owner

Menu Mode

Select the **Update** option from the **File** menu. A dialog appears allowing you to specify files, revision criteria, and options.

Command Mode Syntax

```
>UP[DATE] filelist

     [ ;APPEND ]
     [ ;BATCH ]
     [ ;COMPRESS ]
     [ ;DECOMPRESS ]
     [ ;MEMO [ = memo-text ] ]
     [ ;OLDDATE ]
     [ ;VERIFY ]
```

Parameters

filelist	A list of files, as described in *How to Refer to Files* at the beginning of this chapter.
APPEND	Appends data from the source file to the end of the destination file. EOF plus new data may not exceed LIMIT.
BATCH	Processes the transaction in batch mode. Refer to *Batch Operations* at the beginning of this chapter.
COMPRESS	Compresses the new destination file(s).
DECOMPRESS	Decompresses the new destination file(s).
MEMO = *memo−text*	Lets you include comments describing the current transaction. If you do not specify memo text, LIBRARIAN will invoke the configured editor so that you can enter the memo text.
OLDDATE	Leaves timestamp in the database as is and sets the create and modify dates of the destination file to be the same as the from file.
VERIFY	Identifies files that have been modified by comparing the modification timestamp in the file label with the timestamp stored when it was last created by LIBRARIAN. If the master file has been modified since it was last checked in, the file is not copied (violation).

UPDATE (continued)

Operation

UPDATE updates a read mode secondary copy of a file with the current associated master of the file. You do not have to specify the name of the master file to update your copy.

Examples

Update a program with the most current master version by typing:

MPE
```
>UPDATE AP.PROG.FINTEST
```

UNIX
```
>UPDATE /fintest/prog/ap
```

USER

Establishes a new user for the current LIBRARIAN session, and/or changes user password/lockword.

Restrictions

None

Menu Mode

Select the **Identification** option from the User menu to establish a new user for this session. Select the **Passwords** option from the User menu to change passwords. A dialog appears allowing you to specify a new password and/or lockword.

Command Mode Syntax

```
>US[ER] [ user [ ;password ] ]

     [ ;PASS[WORD] [ = newpass ] ]
     [ ;LOCK[WORD] [ = lockword ] ]
```

Parameters

user	The user ID of any user defined in the LIBRARIAN database.
password	The password of the specified user. If the password is not specified but the user ID is specified, the system prompts for the password.
PASSWORD [= *newpass*]	Changes the password. You will be prompted for your current password if not specified.

Note All LIBRARIAN users and passwords are case-sensitive.

MPE **LOCKWORD** [= *lockword*]	Changes the lockword. You will be prompted for the current lockword if not specified.

If you issue the **USER** command without parameters, the system displays the current active user.

Operation

USER establishes the active user. All subsequent file operations are authorized on the basis of this user. You can issue the **USER** command at any time during a session to change your user.

If you want to change your password and/or lockword, LIBRARIAN first validates your current user ID and password. It then processes the **PASSWORD** and **LOCKWORD** parameters. If the keywords are used alone, LIBRARIAN prompts you for the new password and/or lockword.

Operation, continued

Using the System Profile (SP) screen, the LIBRARIAN Manager can place restrictions on user passwords such as expiration policy, minimum password length, and maximum number of attempts allowed for users to provide a valid password.

Note The **USER** command enables the LIBRARIAN Manager to change any user's password, without having to know the user's old password. This is especially useful when users forget their passwords.

Examples

Display the active user by typing:

```
>USER
```

The LIBRARIAN program displays.

```
*USER = GLMGR
```

Set the active user ID to OPER by typing:

```
>USER OPER
```

The LIBRARIAN program then prompts for the password.

```
Password?
```

Set the active user ID to OPER and specify a password at the same time by typing:

```
USER OPER:PASS5439
```

Change the current user's password to GLIP by typing:

```
USER OPER; PASSWORD=GLIP
```

Related Information

See System Profile (SP) Screen in Chapter 5, "Screens"
Chapter 5, "Users and Authorizations" in the *LIBRARIAN/iX Administrator's Guide*
Chapter 2, "Getting Started" in the *LIBRARIAN/iX User's Guide*

VERIFY

Shows information about files and revisions.

Restrictions

None

Menu Mode

Select the **Files...Verify** option from the **Info** menu. A dialog appears allowing you to specify files, revision criteria, and options.

Command Mode Syntax

```
>V[ERIFY] filelist

    [ ;EXTERNAL ]
    [ ;FORMAT = format ]
    [ ;LP ]
    [ ;MODE = mode ]
    [ ;ONDISK ]
    [ ;OWNER = owner ]
    [ ;PROJECT = project ]
    [ ;REV[ISION] = revision/ALL ]
    [ ;STEP = step.route.appl ]
    [ ;UNTRACKED ]
```

Parameters

filelist	A list of files, as described in *How to Refer to Files* at the beginning of this chapter.
EXTERNAL	Causes **VERIFY** to check only the LIBRARIAN database without going to the disk to check for files. Do not use wildcards.
FORMAT = *format*	Specifies the format number. Default is to present a menu of formats.
LP	Directs output to the LP device, unless a file equation for LIBOUT or LIBVFY has been issued.
MODE = *mode*	Selects only files whose mode is READ or WRITE, as specified.
ONDISK	When verifying master files/filesets, only includes files that currently exist on disk. The **ONDISK** parameter excludes master files that have been purged, but which have retained revisions. Physical file specifications for master files using wildcards will also only report files that exist on disk. Purged master files that still have retained revisions will not be found.
OWNER = *owner*	Selects only files owned by the user specified.
PROJECT = *project*	Selects only files that are associated with the specified project.

VERIFY (continued)

Parameters, continued

REVISION = *revision*/**ALL**	Retrieves all prior revisions of the file(s) requested, even if the master file has been purged. You can **VERIFY** revisions of a file even if the master file has been purged.
STEP = *step.route.appl*	Selects only files whose last associated step is the one specified.
UNTRACKED	Shows only files not currently being tracked by LIBRARIAN.

Operation

VERIFY works as an online or offline report providing information about files. Files are classified as Unknown, Masters, Secondaries, Retained, or Delta. Each format applies to files in one or more categories, as shown in the VERIFY menu.

Note Purged master files (and previous revisions) are reported by VERIFY only if requested by specific filename or fileset; requests by wildcard expression will not find these files since they do not exist on disk.

When you issue **VERIFY**, a menu of formats is displayed, unless the **FORMAT** option is used.

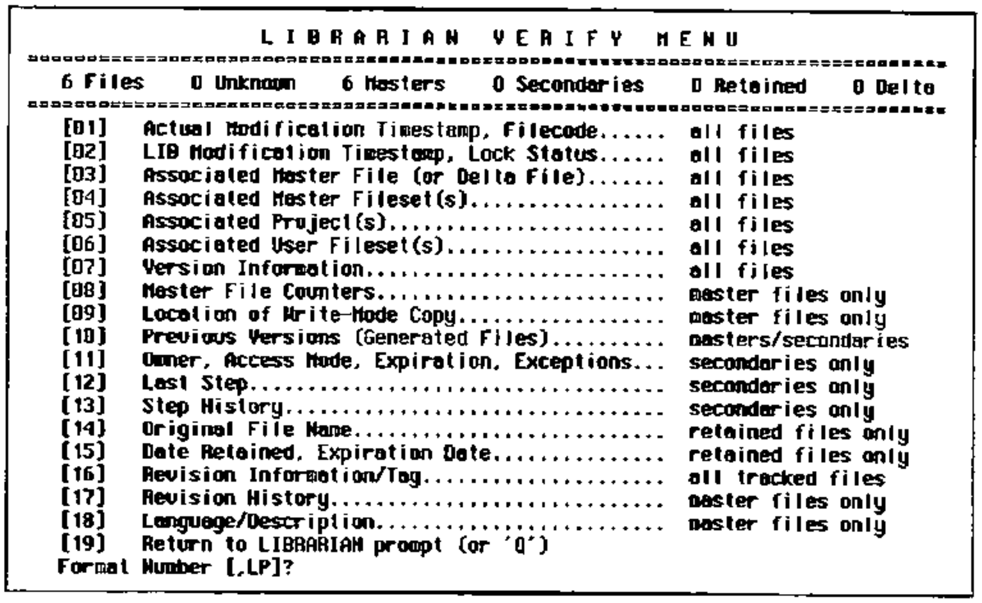
When you enter an option number, the requested information displays. You can continue to choose options until you exit VERIFY (option 18) or type Q.

You can direct VERIFY output offline by either using the **LP** parameter when you issue the command, or by including LP preceded by a comma after the format option.

Issue the **VERIFY** command with the **FORMAT** parameter to bypass the initial menu display.

You can also specify a new format following a VERIFY display without having to return to the VERIFY menu. The prompt is "Format[,LP] (RETURN for Menu/Q to Quit)."

Following is an illustration of the LIBRARIAN Verify Menu, which lists the 18 formats shown on the following pages.

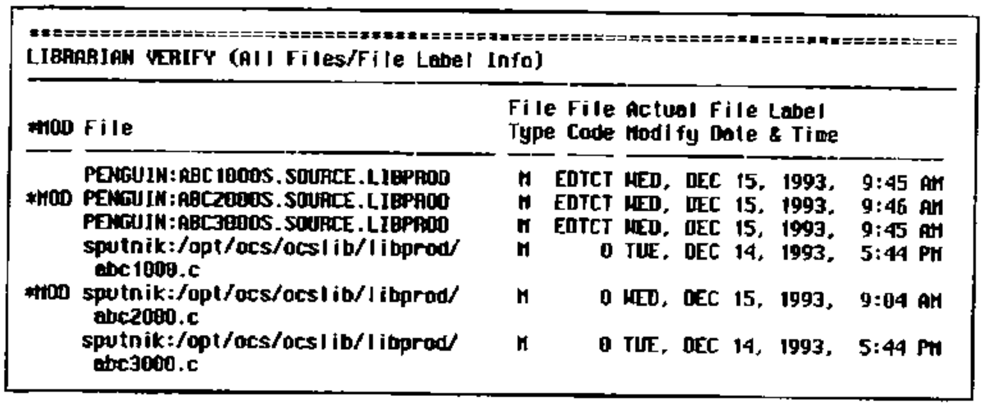


```
                 L I B R A R I A N   V E R I F Y   M E N U
==========================================================================
  6 Files    0 Unknown    6 Masters    0 Secondaries    0 Retained    0 Delta
==========================================================================
   [01]   Actual Modification Timestamp, Filecode......  all files
   [02]   LIB Modification Timestamp, Lock Status......  all files
   [03]   Associated Master File (or Delta File).......  all files
   [04]   Associated Master Fileset(s).................  all files
   [05]   Associated Project(s)........................  all files
   [06]   Associated User Fileset(s)...................  all files
   [07]   Version Information..........................  all files
   [08]   Master File Counters.........................  master files only
   [09]   Location of Write-Mode Copy..................  master files only
   [10]   Previous Versions (Generated Files)..........  masters/secondaries
   [11]   Owner, Access Mode, Expiration, Exceptions...  secondaries only
   [12]   Last Step....................................  secondaries only
   [13]   Step History.................................  secondaries only
   [14]   Original File Name...........................  retained files only
   [15]   Date Retained, Expiration Date...............  retained files only
   [16]   Revision Information/Tag.....................  all tracked files
   [17]   Revision History.............................  master files only
   [18]   Language/Description.........................  master files only
   [19]   Return to LIBRARIAN prompt (or 'Q')
  Format Number [,LP]?
```

Figure 1-1. VERIFY Menu

VERIFY (continued)

Format 1: Actual Modification Timestamp, Filecode



```
==============================================================================
LIBRARIAN VERIFY (All Files/File Label Info)
------------------------------------------------------------------------------
                                          File File Actual File Label
*MOD File                                 Type Code Modify Date & Time
---- ------------------------------------ ---- ---- -------------------------
     PENGUIN:ABC1000S.SOURCE.LIBPROD       M  EDTCT WED, DEC 15, 1993,  9:45 AM
*MOD PENGUIN:ABC2000S.SOURCE.LIBPROD       M  EDTCT WED, DEC 15, 1993,  9:46 AM
     PENGUIN:ABC3000S.SOURCE.LIBPROD       M  EDTCT WED, DEC 15, 1993,  9:45 AM
     sputnik:/opt/ocs/ocslib/libprod/      M      0 TUE, DEC 14, 1993,  5:44 PM
      abc1000.c
*MOD sputnik:/opt/ocs/ocslib/libprod/      M      0 WED, DEC 15, 1993,  9:04 AM
      abc2000.c
     sputnik:/opt/ocs/ocslib/libprod/      M      0 TUE, DEC 14, 1993,  5:44 PM
      abc3000.c
```

Figure 1-2. VERIFY Format 1, Actual Modification Timestamp

Field Descriptions

***MOD**	A flag that indicates whether a file has been modified since the time it was created by LIBRARIAN
File	Name of the file including the system name.
File Type	Type of file. One of the following:

??	Unknown file
M	Master file
S	Secondary file
D	Delta file
GM	Generation of a master file (retained)
GS	Generation of a secondary file (retained)
CM	Copy of a retained master file
CS	Copy of a retained secondary file

File Code The filecode associated with the file. A filecode of 1012 indicates that a file is in compressed format.

Actual File Label Modify Date & Time
The modification timestamp recorded by the file system for the file.

VERIFY (continued)

Format 2: LIB Modification Timestamp, Lock Status

```
==============================================================================
LIBRARIAN VERIFY (All Files/LIB Timestamp)
------------------------------------------------------------------------------
                                          File       LIBRARIAN Timestamp
*MOD File                                 Type Lock  Modify Date & Time
---- ------------------------------------ ---- ----  -------------------------
     PENGUIN:ABC1000S.SOURCE.LIBPROD       M         WED, DEC 15, 1993,  9:45 AM
*MOD PENGUIN:ABC2000S.SOURCE.LIBPROD       M         WED, DEC 15, 1993,  9:45 AM
     PENGUIN:ABC3000S.SOURCE.LIBPROD       M         WED, DEC 15, 1993,  9:45 AM
     sputnik:/opt/ocs/ocslib/libprod/      M         TUE, DEC 14, 1993,  5:44 PM
      abc1000.c
*MOD sputnik:/opt/ocs/ocslib/libprod/      M         TUE, DEC 14, 1993,  5:54 PM
      abc2000.c
     sputnik:/opt/ocs/ocslib/libprod/      M         TUE, DEC 14, 1993,  5:44 PM
      abc3000.c
```

Figure 1-3.   VERIFY Format 2, LIB Modification Timestamp

Field Descriptions

***MOD** A flag that indicates whether a file has been modified since the time it was created by LIBRARIAN.

File Name of the file including the system name.

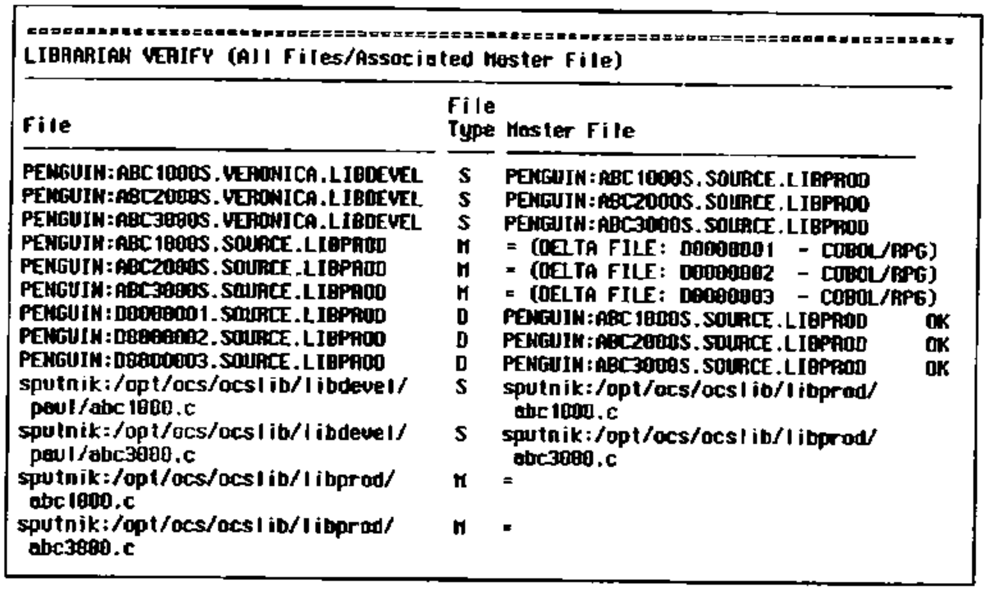
File Type Type of file. One of the following:

??	Unknown file
M	Master file
S	Secondary file
D	Delta file
GM	Generation of a master file (retained)
GS	Generation of a secondary file (retained)
CM	Copy of a retained master file
CS	Copy of a retained secondary file

Lock A flag that indicates whether a file has been locked using the LOCK command.

LIBRARIAN Timestamp Modify Date & Time
The modification timestamp recorded by LIBRARIAN when it created the file (or when it loaded the master during AUTOUPDATE).

Format 3: Associated Master File (or Delta File)



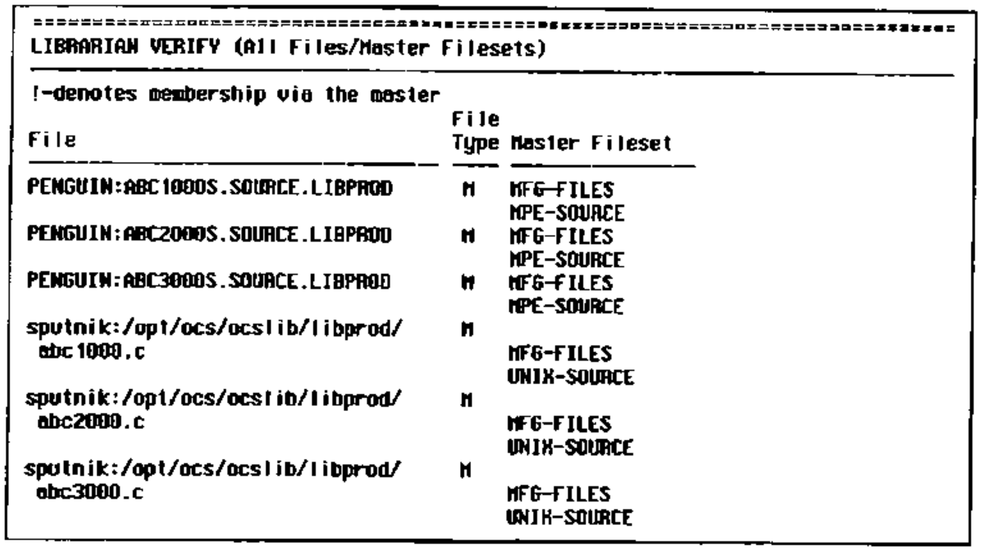
```
==============================================================================
LIBRARIAN VERIFY (All Files/Associated Master File)
------------------------------------------------------------------------------
                                    File
File                                Type Master File
----------------------------------- ---- ------------------------------------
PENGUIN:ABC1000S.VERONICA.LIBDEVEL   S   PENGUIN:ABC1000S.SOURCE.LIBPROD
PENGUIN:ABC2000S.VERONICA.LIBDEVEL   S   PENGUIN:ABC2000S.SOURCE.LIBPROD
PENGUIN:ABC3000S.VERONICA.LIBDEVEL   S   PENGUIN:ABC3000S.SOURCE.LIBPROD
PENGUIN:ABC1000S.SOURCE.LIBPROD      M   = (DELTA FILE: D0000001  - COBOL/RPG)
PENGUIN:ABC2000S.SOURCE.LIBPROD      M   = (DELTA FILE: D0000002  - COBOL/RPG)
PENGUIN:ABC3000S.SOURCE.LIBPROD      M   = (DELTA FILE: D0000003  - COBOL/RPG)
PENGUIN:D0000001.SOURCE.LIBPROD      D   PENGUIN:ABC1000S.SOURCE.LIBPROD      OK
PENGUIN:D0000002.SOURCE.LIBPROD      D   PENGUIN:ABC2000S.SOURCE.LIBPROD      OK
PENGUIN:D0000003.SOURCE.LIBPROD      D   PENGUIN:ABC3000S.SOURCE.LIBPROD      OK
sputnik:/opt/ocs/ocslib/libdevel/    S   sputnik:/opt/ocs/ocslib/libprod/
 paul/abc1000.c                           abc1000.c
sputnik:/opt/ocs/ocslib/libdevel/    S   sputnik:/opt/ocs/ocslib/libprod/
 paul/abc3000.c                           abc3000.c
sputnik:/opt/ocs/ocslib/libprod/     M   =
 abc1000.c
sputnik:/opt/ocs/ocslib/libprod/     M   =
 abc3000.c
```

Figure 1-4. VERIFY Format 3, Associated Master File

Field Descriptions

File	Name of the file including the system name.
File Type	Type of file. One of the following:
	?? Unknown file
	M Master file
	S Secondary file
	D Delta file
	GM Generation of a master file (retained)
	GS Generation of a secondary file (retained)
	CM Copy of a retained master file
	CS Copy of a retained secondary file
Master File	Name of the associated master file. If File Type is M and deltas are in effect for the application, then the associated delta file name and language is shown. If File Type is D, a checksum verification status appears to the right of this field. If the status is not OK then the delta file has an integrity problem.

Format 4: Associated Master Fileset(s)



```
==================================================================================
LIBRARIAN VERIFY (All Files/Master Filesets)
----------------------------------------------------------------------------------
!-denotes membership via the master
                                      File
File                                  Type Master Fileset
------------------------------------- ---- ---------------
PENGUIN:ABC1000S.SOURCE.LIBPROD        M   MFG-FILES
                                           MPE-SOURCE
PENGUIN:ABC2000S.SOURCE.LIBPROD        M   MFG-FILES
                                           MPE-SOURCE
PENGUIN:ABC3000S.SOURCE.LIBPROD        M   MFG-FILES
                                           MPE-SOURCE
sputnik:/opt/ocs/ocslib/libprod/       M
 abc1000.c                                 MFG-FILES
                                           UNIX-SOURCE
sputnik:/opt/ocs/ocslib/libprod/       M
 abc2000.c                                 MFG-FILES
                                           UNIX-SOURCE
sputnik:/opt/ocs/ocslib/libprod/       M
 abc3000.c                                 MFG-FILES
                                           UNIX-SOURCE
```

Figure 1-5. VERIFY Format 4, Associated Master Filesets

Field Descriptions

File Name of the file including the system name.

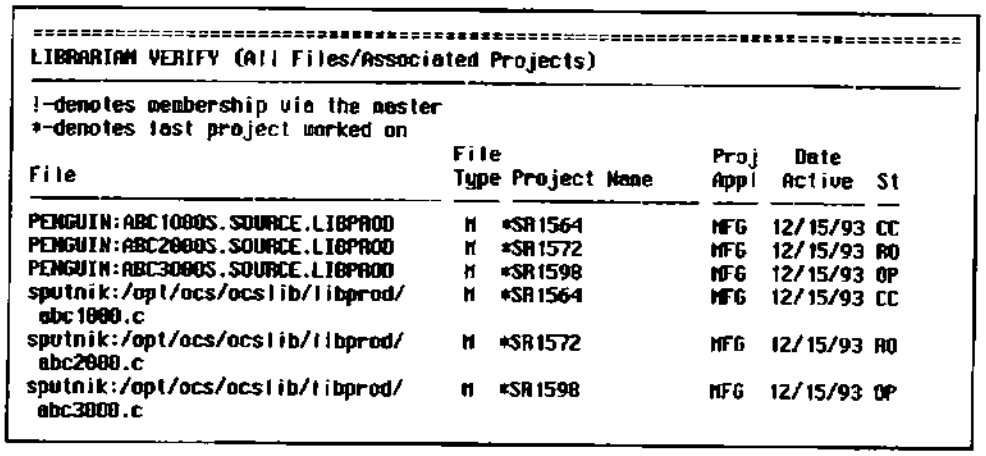
File Type Type of file. One of the following:

??	Unknown file
M	Master file
S	Secondary file
D	Delta file
GM	Generation of a master file (retained)
GS	Generation of a secondary file (retained)
CM	Copy of a retained master file
CS	Copy of a retained secondary file

Master Fileset The names of all master filesets to which the file belongs. For non–master files, all filesets to which the associated master file belongs; these filesets appear with a prefix of "!".

VERIFY (continued)

Format 5: Associated Project(s)



```
==============================================================================
LIBRARIAN VERIFY (All Files/Associated Projects)
------------------------------------------------------------------------------
!-denotes membership via the master
*-denotes last project worked on
                                   File                       Proj   Date
File                               Type Project Name          Appl  Active   St
---------------------------------- ---- ----------------      ---- --------  --
PENGUIN:ABC1000S.SOURCE.LIBPROD     M   *SR1564               MFG  12/15/93 CC
PENGUIN:ABC2000S.SOURCE.LIBPROD     M   *SR1572               MFG  12/15/93 RO
PENGUIN:ABC3000S.SOURCE.LIBPROD     M   *SR1598               MFG  12/15/93 OP
sputnik:/opt/ocs/ocslib/libprod/    M   *SR1564               MFG  12/15/93 CC
 abc1000.c
sputnik:/opt/ocs/ocslib/libprod/    M   *SR1572               MFG  12/15/93 RO
 abc2000.c
sputnik:/opt/ocs/ocslib/libprod/    M   *SR1598               MFG  12/15/93 OP
 abc3000.c
```

Figure 1-6. VERIFY Format 5, Associated Projects

Field Descriptions

File Name of the file including the system name.

File Type Type of file. One of the following:

??	Unknown file
M	Master file
S	Secondary file
D	Delta file
GM	Generation of a master file (retained)
GS	Generation of a secondary file (retained)
CM	Copy of a retained master file
CS	Copy of a retained secondary file

Project Name The names of all projects (project filesets) to which the file belongs. For non-master files, all projects (project filesets) to which the associated master file belongs; these filesets appear with a prefix of "!". A "*" indicates that the file is currently being worked on under this project, or was the last project this file was associated with.

Proj Appl The name of the application to which the project belongs.

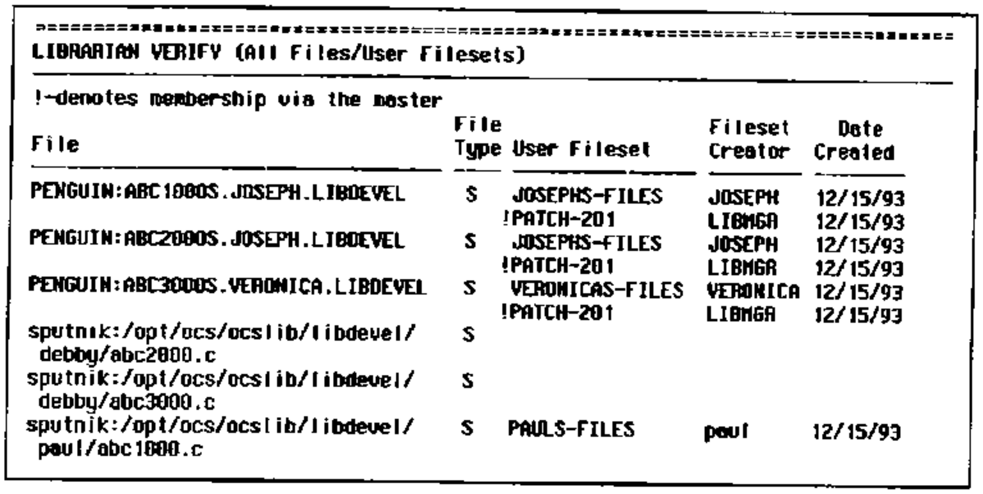
Date Active The date the project was first worked on.

St. The current project status:

OP	Opened
RO	Re-Opened
CL	Closed
CC	Closed-to-Checkout
FP	Flush Pending

VERIFY (continued)

Format 6: Associated User Fileset(s)



```
=============================================================================
LIBRARIAN VERIFY (All Files/User Filesets)
-----------------------------------------------------------------------------
!-denotes membership via the master
                                   File                   Fileset   Date
File                               Type User Fileset      Creator   Created
---------------------------------- ---- ----------------- --------- --------
PENGUIN:ABC1000S.JOSEPH.LIBDEVEL    S    JOSEPHS-FILES    JOSEPH    12/15/93
                                        !PATCH-201        LIBMGR    12/15/93
PENGUIN:ABC2000S.JOSEPH.LIBDEVEL    S    JOSEPHS-FILES    JOSEPH    12/15/93
                                        !PATCH-201        LIBMGR    12/15/93
PENGUIN:ABC3000S.VERONICA.LIBDEVEL  S    VERONICAS-FILES  VERONICA  12/15/93
                                        !PATCH-201        LIBMGR    12/15/93
sputnik:/opt/ocs/ocslib/libdevel/   S
 debby/abc2000.c
sputnik:/opt/ocs/ocslib/libdevel/   S
 debby/abc3000.c
sputnik:/opt/ocs/ocslib/libdevel/   S   PAULS-FILES       paul      12/15/93
 paul/abc1000.c
```

Figure 1-7. VERIFY Format 6. Associated User Filesets

Field Descriptions

File Name of the file including the system name.

File Type Type of file. One of the following:

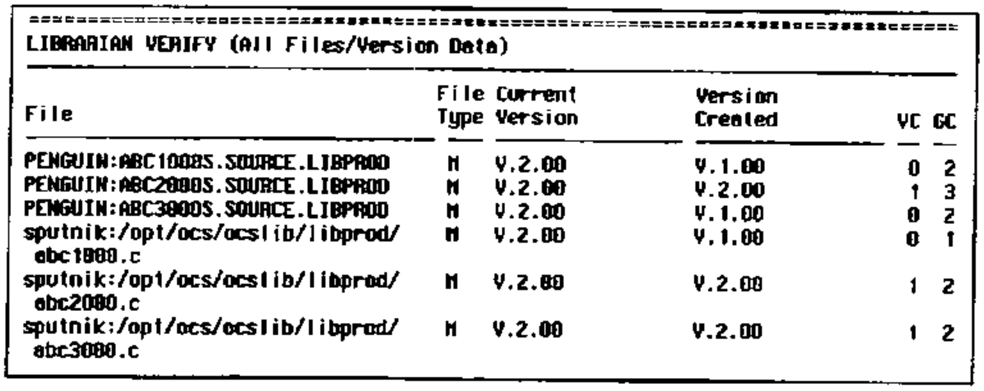
??	Unknown file
M	Master file
S	Secondary file
D	Delta file
GM	Generation of a master file (retained)
GS	Generation of a secondary file (retained)
CM	Copy of a retained master file
CS	Copy of a retained secondary file

User Fileset The names of all user filesets to which the file belongs. For non–master files, includes all filesets to which the associated master file belongs as well; these filesets appear with a prefix of "!".

Fileset Creator The user who created the user fileset.

Date Created The date the fileset was created.

Format 7: Version Information



```
================================================================================
LIBRARIAN VERIFY (All Files/Version Data)
--------------------------------------------------------------------------------
                                     File Current           Version
File                                 Type Version           Created        VC GC
------------------------------------ ---- ----------------- -------------- -- --
PENGUIN:ABC1000S.SOURCE.LIBPROD       M   V.2.00            V.1.00          0  2
PENGUIN:ABC2000S.SOURCE.LIBPROD       M   V.2.00            V.2.00          1  3
PENGUIN:ABC3000S.SOURCE.LIBPROD       M   V.2.00            V.1.00          0  2
sputnik:/opt/ocs/ocslib/libprod/      M   V.2.00            V.1.00          0  1
 abc1000.c
sputnik:/opt/ocs/ocslib/libprod/      M   V.2.00            V.2.00          1  2
 abc2000.c
sputnik:/opt/ocs/ocslib/libprod/      M   V.2.00            V.2.00          1  2
 abc3000.c
```

Figure 1-8. VERIFY Format 7, Version Information

Field Descriptions

File	Name of the file including the system name.
File Type	Type of file. One of the following:
	?? Unknown file
	M Master file
	S Secondary file
	D Delta file
	GM Generation of a master file (retained)
	GS Generation of a secondary file (retained)
	CM Copy of a retained master file
	CS Copy of a retained secondary file
Current Version	The current version to which the file belongs. For retained files, the last version to which the file belonged.
Version Created	The first version to which the file belonged.
VC	The version count of the file.
GC	The generation count of the file.

VERIFY (continued)

Format 8: Master File Counters

```
================================================================================
LIBRARIAN VERIFY (Master Files/Counters)
--------------------------------------------------------------------------------
                                     ---- CUR ----    -------- TOTAL --------

                                     Rd  Wr  Ret Ret   Rd   Wr   Ret  Ret  Check
File                                 Cop Cop Sec Mst   Cop  Cop  Sec  Mst   Ins
------------------------------------ --- --- --- ---   ---  ---  ---  ---  -----
PENGUIN:ABC1000S.SOURCE.LIBPROD       0   1   0   1     0    2    0    1     1
PENGUIN:ABC2000S.SOURCE.LIBPROD       0   1   0   1     0    2    0    1     1
PENGUIN:ABC3000S.SOURCE.LIBPROD       0   1   0   1     0    2    0    1     1
sputnik:/opt/ocs/ocslib/libprod/      0   1   0   0     0    2    0    0     0
 abc1000.c
sputnik:/opt/ocs/ocslib/libprod/      0   1   0   0     0    1    0    0     0
 abc2000.c
sputnik:/opt/ocs/ocslib/libprod/      0   1   0   0     0    1    0    0     0
 abc3000.c
```

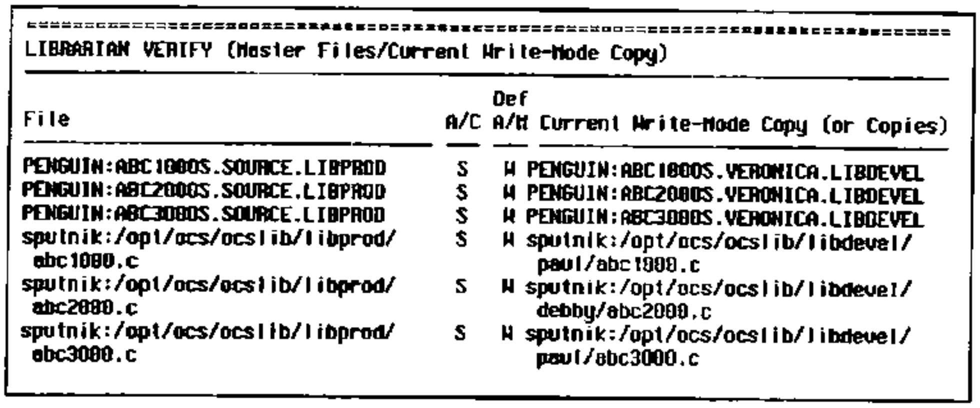
Figure 1-9. VERIFY Format 8, Master File Counters

Field Descriptions

File	Name of the file including the system name.
—— CUR ——	
Rd Cop	Current number of read mode secondary copies.
Wr Cop	Current number of write mode secondary copies.
Ret Sec	Current number of retained secondary copies.
Ret Mst	Current number of retained master copies.
—— TOTAL ——	
Rd Cop	Total number of read mode secondary copies ever created.
Wr Cop	Total number of write mode secondary copies ever created.
Ret Sec	Total number of retained secondary copies ever created.
Ret Mst	Total number of retained master copies ever created.
Check Ins	Total number of times the file has ever been checked in.

VERIFY (continued)

Format 9: Location of Write Mode Copy



```
==============================================================================
LIBRARIAN VERIFY (Master Files/Current Write-Mode Copy)
------------------------------------------------------------------------------
                                          Def
File                                 A/C  A/M  Current Write-Mode Copy (or Copies)
------------------------------------ ---  ---  -----------------------------------
PENGUIN:ABC1000S.SOURCE.LIBPROD       S    W  PENGUIN:ABC1000S.VERONICA.LIBDEVEL
PENGUIN:ABC2000S.SOURCE.LIBPROD       S    W  PENGUIN:ABC2000S.VERONICA.LIBDEVEL
PENGUIN:ABC3000S.SOURCE.LIBPROD       S    W  PENGUIN:ABC3000S.VERONICA.LIBDEVEL
sputnik:/opt/ocs/ocslib/libprod/      S    W  sputnik:/opt/ocs/ocslib/libdevel/
 abc1000.c                                      paul/abc1000.c
sputnik:/opt/ocs/ocslib/libprod/      S    W  sputnik:/opt/ocs/ocslib/libdevel/
 abc2000.c                                      debby/abc2000.c
sputnik:/opt/ocs/ocslib/libprod/      S    W  sputnik:/opt/ocs/ocslib/libdevel/
 abc3000.c                                      paul/abc3000.c
```

Figure 1–10. VERIFY Format 9, Location of Write Mode Copy

Field Descriptions

File Name of the file including the system name.

A/C Access control level for the master file:

X	Exclusive Access
R	Read Only Access
S	Serial Write Access
M	Multi Write Access

Def A/M Default access mode for copies of the master file:

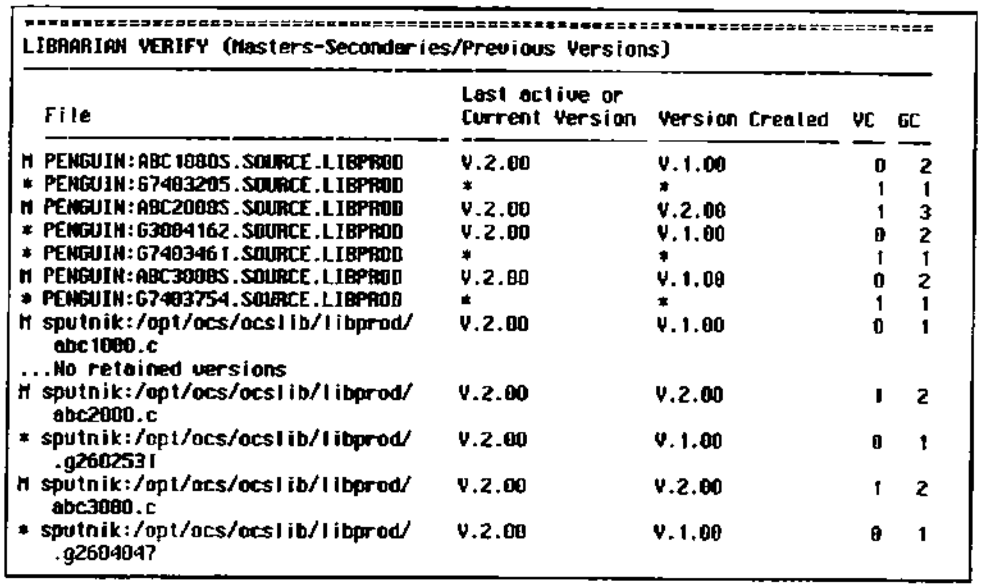
R	Read
W	Write

Current Write–Mode Copy (or Copies)

Name(s) of the current write–mode secondary files associated with the master file

VERIFY (continued)

Format 10: Previous Versions (Generated Files)



```
==========================================================================================
LIBRARIAN VERIFY (Masters-Secondaries/Previous Versions)
------------------------------------------------------------------------------------------
                                        Last active or
  File                                  Current Version  Version Created  VC  GC
  ------------------------------------  ---------------  ---------------  --  --
M PENGUIN:ABC1000S.SOURCE.LIBPROD       V.2.00           V.1.00            0   2
* PENGUIN:G7403205.SOURCE.LIBPROD       *                *                 1   1
M PENGUIN:ABC2000S.SOURCE.LIBPROD       V.2.00           V.2.00            1   3
* PENGUIN:G3004162.SOURCE.LIBPROD       V.2.00           V.1.00            0   2
* PENGUIN:G7403461.SOURCE.LIBPROD       *                *                 1   1
M PENGUIN:ABC3000S.SOURCE.LIBPROD       V.2.00           V.1.00            0   2
* PENGUIN:G7403754.SOURCE.LIBPROD       *                *                 1   1
M sputnik:/opt/ocs/ocslib/libprod/      V.2.00           V.1.00            0   1
   abc1000.c
...No retained versions
M sputnik:/opt/ocs/ocslib/libprod/      V.2.00           V.2.00            1   2
   abc2000.c
* sputnik:/opt/ocs/ocslib/libprod/      V.2.00           V.1.00            0   1
   .g2602531
M sputnik:/opt/ocs/ocslib/libprod/      V.2.00           V.2.00            1   2
   abc3000.c
* sputnik:/opt/ocs/ocslib/libprod/      V.2.00           V.1.00            0   1
   .g2604047
```

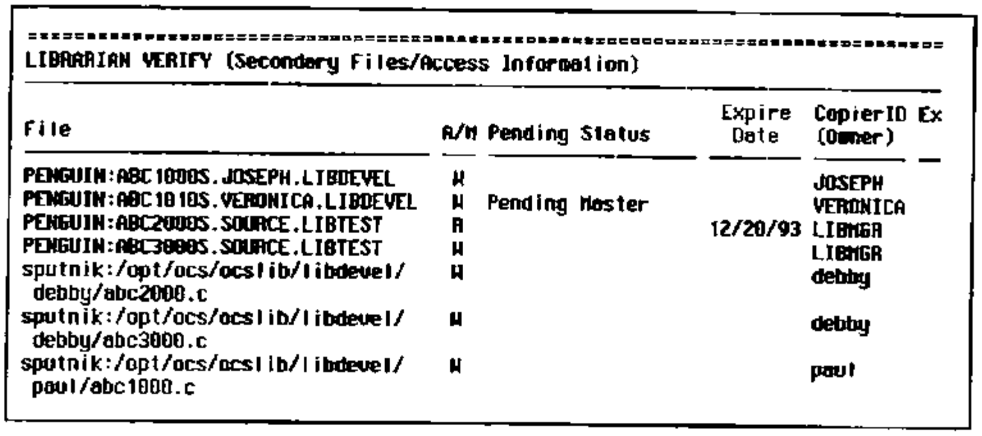
Figure 1-11. VERIFY Format 10, Previous Versions

Field Descriptions

File	Name of the master or secondary file including system name. An M in the first column indicates a master file, and an S in the first column indicates a secondary file. A * indicates a generation of the master or secondary file listed above it.
Last active or Current Version	The current version to which the file belongs.
VersionCreated	The first version to which the file belonged.
VC	The version count of the file.
GC	The generation count of the file.

VERIFY (continued)

Format 11: Owner, Access Mode, Expiration



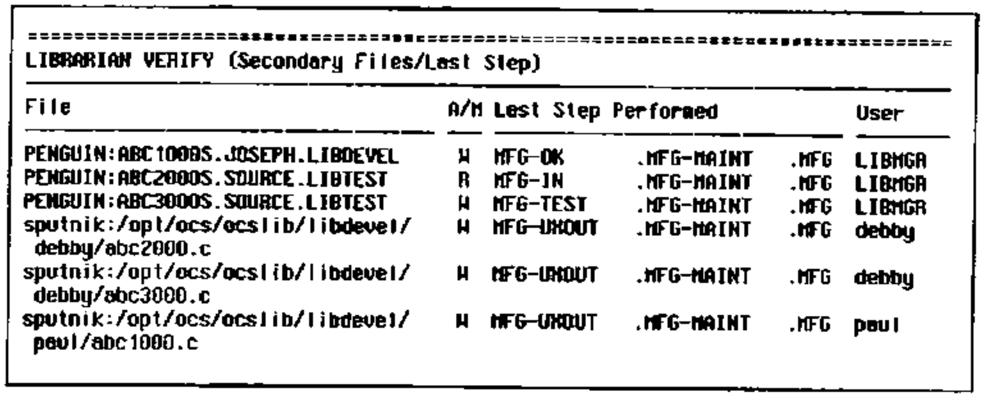
```
================================================================================
LIBRARIAN VERIFY (Secondary Files/Access Information)
-------------------------------------------------------------------------------
                                                           Expire  CopierID Ex
File                                 A/M Pending Status     Date   (Owner)
------------------------------------ --- -------------- -------- -------- --
PENGUIN:ABC1000S.JOSEPH.LIBDEVEL      W                           JOSEPH
PENGUIN:ABC1010S.VERONICA.LIBDEVEL    W  Pending Master           VERONICA
PENGUIN:ABC2000S.SOURCE.LIBTEST       R                  12/20/93 LIBMGR
PENGUIN:ABC3000S.SOURCE.LIBTEST       W                           LIBMGR
sputnik:/opt/ocs/ocslib/libdevel/     W                           debby
 debby/abc2000.c
sputnik:/opt/ocs/ocslib/libdevel/     W                           debby
 debby/abc3000.c
sputnik:/opt/ocs/ocslib/libdevel/     W                           paul
 paul/abc1000.c
```

Figure 1-12. VERIFY Format 11, Owner, Access Mode, Expiration

Field Descriptions

File	Name of the file including the system name.
A/M	The access mode of the secondary file: R Read W Write
Pending Status	A flag to indicate whether the file is new and has no associated physical master file yet.
Expire Date	For read mode secondaries, the date the file expires or expired.
Copier ID (Owner)	The LIBRARIAN user who created the secondary file.
Ex	The following is a list of exception codes: MC Merge conflict PR Read mode copy checked in (emergency fix) RS Previous version of master file restored while checked out

Format 12: Last Step



```
================================================================================
LIBRARIAN VERIFY (Secondary Files/Last Step)
--------------------------------------------------------------------------------
File                                 A/M Last Step Performed             User
------------------------------------ --- ------------------------------ --------
PENGUIN:ABC1000S.JOSEPH.LIBDEVEL      W  MFG-OK      .MFG-MAINT   .MFG  LIBMGR
PENGUIN:ABC2000S.SOURCE.LIBTEST       R  MFG-IN      .MFG-MAINT   .MFG  LIBMGR
PENGUIN:ABC3000S.SOURCE.LIBTEST       W  MFG-TEST    .MFG-MAINT   .MFG  LIBMGR
sputnik:/opt/ocs/ocslib/libdevel/     W  MFG-UWOUT   .MFG-MAINT   .MFG  debby
 debby/abc2000.c
sputnik:/opt/ocs/ocslib/libdevel/     W  MFG-UWOUT   .MFG-MAINT   .MFG  debby
 debby/abc3000.c
sputnik:/opt/ocs/ocslib/libdevel/     W  MFG-UWOUT   .MFG-MAINT   .MFG  paul
 paul/abc1000.c
```

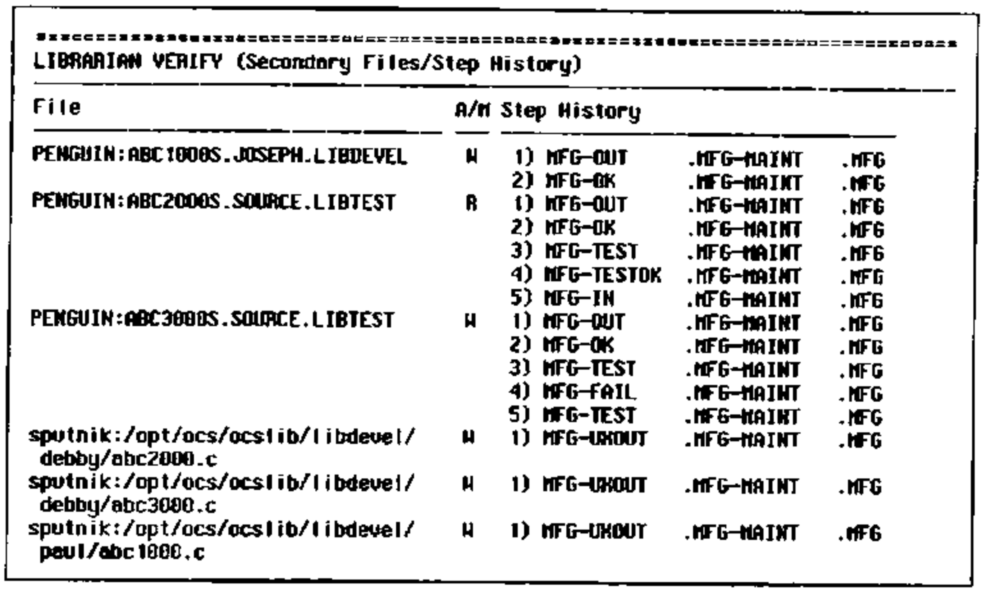
Figure 1-13. VERIFY Format 12, Last Step

Field Descriptions

File	Name of the file including the system name.
A/M	The access mode of the secondary file: R Read W Write
Last Step Performed	The name of the last step that created the file.
User	The LIBRARIAN user who performed the last step on the file.

VERIFY (continued)

Format 13: Step History



```
=================================================================================
LIBRARIAN VERIFY (Secondary Files/Step History)
---------------------------------------------------------------------------------
File                                A/M Step History
----------------------------------- --- -------------------------------------
PENGUIN:ABC1000S.JOSEPH.LIBDEVEL     W   1) MFG-OUT      .MFG-MAINT    .MFG
                                         2) MFG-OK       .MFG-MAINT    .MFG
PENGUIN:ABC2000S.SOURCE.LIBTEST      R   1) MFG-OUT      .MFG-MAINT    .MFG
                                         2) MFG-OK       .MFG-MAINT    .MFG
                                         3) MFG-TEST     .MFG-MAINT    .MFG
                                         4) MFG-TESTOK   .MFG-MAINT    .MFG
                                         5) MFG-IN       .MFG-MAINT    .MFG
PENGUIN:ABC3000S.SOURCE.LIBTEST      W   1) MFG-OUT      .MFG-MAINT    .MFG
                                         2) MFG-OK       .MFG-MAINT    .MFG
                                         3) MFG-TEST     .MFG-MAINT    .MFG
                                         4) MFG-FAIL     .MFG-MAINT    .MFG
                                         5) MFG-TEST     .MFG-MAINT    .MFG
sputnik:/opt/ocs/ocslib/libdevel/    W   1) MFG-UKOUT    .MFG-MAINT    .MFG
 debby/abc2000.c
sputnik:/opt/ocs/ocslib/libdevel/    W   1) MFG-UKOUT    .MFG-MAINT    .MFG
 debby/abc3000.c
sputnik:/opt/ocs/ocslib/libdevel/    W   1) MFG-UKOUT    .MFG-MAINT    .MFG
 paul/abc1000.c
```

Figure 1-14. VERIFY Format 13, Step History

Field Descriptions

File Name of the file including the system name.

A/M The access mode of the secondary file:

R Read
W Write

Step History A chronological listing of up to the last 32 steps performed on the file.

Format 14: Original File Name

```
================================================================================
LIBRARIAN VERIFY (Retained Files/Original File Name)
--------------------------------------------------------------------------------
                                       File
File                                   Type Original File Name
-------------------------------------- ---- ------------------------------------
PENGUIN:67442526.SOURCE.LIBPROD         GM  PENGUIN:ABC1000S.SOURCE.LIBPROD
PENGUIN:67442775.SOURCE.LIBPROD         GM  PENGUIN:ABC2000S.SOURCE.LIBPROD
PENGUIN:67443231.SOURCE.LIBPROD         GM  PENGUIN:ABC3000S.SOURCE.LIBPROD
sputnik:/opt/ocs/ocslib/libprod/        GM  sputnik:/opt/ocs/ocslib/libprod/
 .g0725541                                   abc1000.c
sputnik:/opt/ocs/ocslib/libprod/        GM  sputnik:/opt/ocs/ocslib/libprod/
 .g7064204                                   abc2000.c
sputnik:/opt/ocs/ocslib/libprod/        GM  sputnik:/opt/ocs/ocslib/libprod/
 .g7070212                                   abc3000.c
```

Figure 1-15. VERIFY Format 14, Original File Name

Field Descriptions

File Name of the file including the system name.

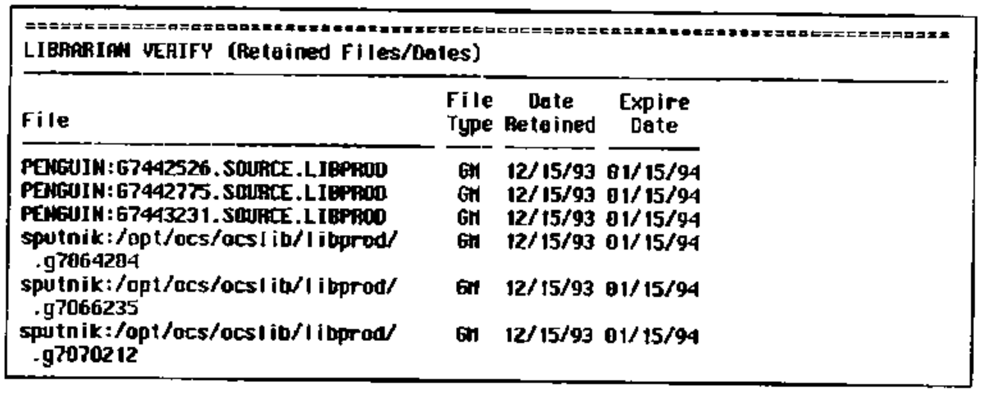
File Type Type of file:

GM	Generation of a master file (retained)
GS	Generation of a secondary file (retained)
CM	Copy of a retained master file
CS	Copy of a retained secondary file

Original File Name

Name of the file prior to having been retained.

Format 15: Date Retained, Expiration Date



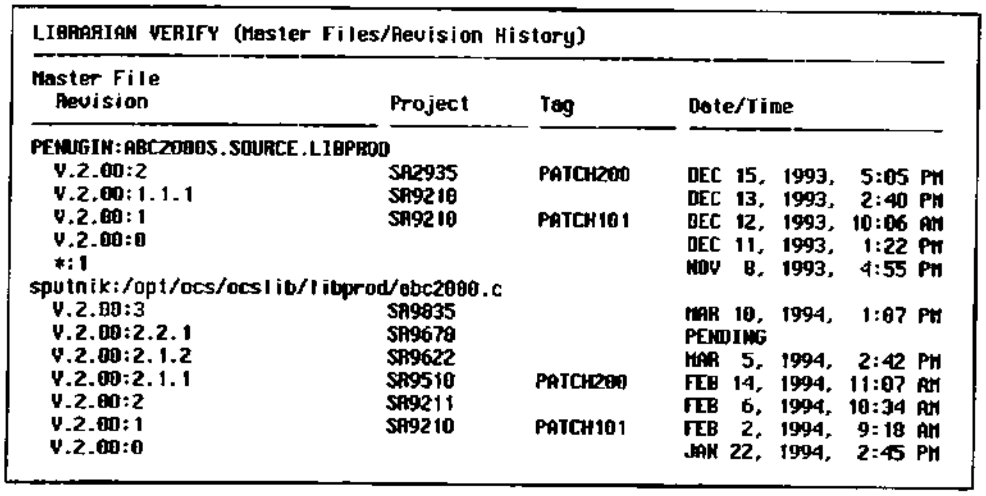
```
================================================================================
LIBRARIAN VERIFY (Retained Files/Dates)
--------------------------------------------------------------------------------
                                       File   Date     Expire
File                                   Type Retained    Date
-------------------------------------- ---- --------  --------
PENGUIN:G7442526.SOURCE.LIBPROD         GM  12/15/93 01/15/94
PENGUIN:G7442775.SOURCE.LIBPROD         GM  12/15/93 01/15/94
PENGUIN:G7443231.SOURCE.LIBPROD         GM  12/15/93 01/15/94
sputnik:/opt/ocs/ocslib/libprod/        GM  12/15/93 01/15/94
 .g7064204
sputnik:/opt/ocs/ocslib/libprod/        GM  12/15/93 01/15/94
 .g7066235
sputnik:/opt/ocs/ocslib/libprod/        GM  12/15/93 01/15/94
 .g7070212
```

Figure 1-16. VERIFY Format 15, Date Retained/Expiration Date

Field Descriptions

File	Name of the file including the system name.
File Type	Type of file:
	GM Generation of a master file (retained)
	GS Generation of a secondary file (retained)
	CM Copy of a retained master file
	CS Copy of a retained secondary file
Date Retained	Date the file was retained.
Expired Date	Date the retained file expires or expired.

Format 16: Revision Information/Tag



```
LIBRARIAN VERIFY (Master Files/Revision History)
------------------------------------------------------------------------------
Master File
  Revision                          Project     Tag          Date/Time
----------------------------------  ----------  -----------  -----------------
PENUGIN:ABC2000S.SOURCE.LIBPROD
  V.2.00:2                          SA2935      PATCH200     DEC 15, 1993,  5:05 PM
  V.2.00:1.1.1                      SA9210                   DEC 13, 1993,  2:40 PM
  V.2.00:1                          SA9210      PATCH101     DEC 12, 1993, 10:06 AM
  V.2.00:0                                                   DEC 11, 1993,  1:22 PM
  *:1                                                        NOV  8, 1993,  4:55 PM
sputnik:/opt/ocs/ocslib/libprod/abc2000.c
  V.2.00:3                          SA9835                   MAR 10, 1994,  1:07 PM
  V.2.00:2.2.1                      SA9670                   PENDING
  V.2.00:2.1.2                      SA9622                   MAR  5, 1994,  2:42 PM
  V.2.00:2.1.1                      SA9510      PATCH200     FEB 14, 1994, 11:07 AM
  V.2.00:2                          SA9211                   FEB  6, 1994, 10:34 AM
  V.2.00:1                          SA9210      PATCH101     FEB  2, 1994,  9:18 AM
  V.2.00:0                                                   JAN 22, 1994,  2:45 PM
```

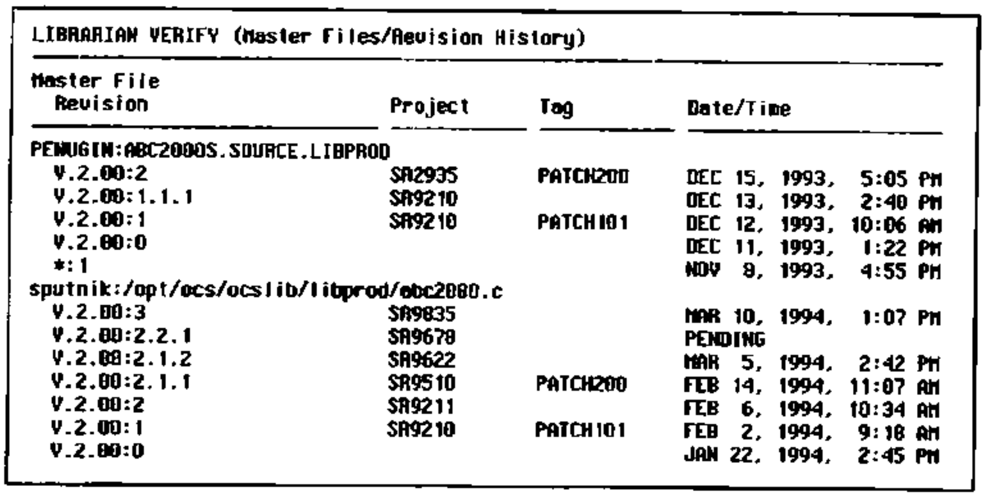
Figure 1-17. VERIFY Format 16, Revision Information tag

Field Descriptions

File	Name of the master/secondary file including system name.
Latest Revision	The revision ID for the file, including version, version count and branch/leaf pairs (if applicable).
Tag	If the file was assigned a tag using the SET TAG command, the tag appears under the revision ID for the file.

VERIFY (continued)

Format 17: Revision History



```
LIBRARIAN VERIFY (Master Files/Revision History)
_____________________________________________________________________________
Master File
  Revision                      Project      Tag          Date/Time
_______________________________ ____________ ____________ _______________________
PENUGIN:ABC2000S.SOURCE.LIBPROD
  V.2.00:2                      SA2935       PATCH200     DEC 15, 1993,  5:05 PM
  V.2.00:1.1.1                  SA9210                    DEC 13, 1993,  2:40 PM
  V.2.00:1                      SA9210       PATCH101     DEC 12, 1993, 10:06 AM
  V.2.00:0                                                DEC 11, 1993,  1:22 PM
  *:1                                                     NOV  8, 1993,  4:55 PM
sputnik:/opt/ocs/ocslib/libprod/abc2080.c
  V.2.00:3                      SA9835                    MAR 10, 1994,  1:07 PM
  V.2.00:2.2.1                  SA9678                    PENDING
  V.2.00:2.1.2                  SA9622                    MAR  5, 1994,  2:42 PM
  V.2.00:2.1.1                  SA9510       PATCH200     FEB 14, 1994, 11:07 AM
  V.2.00:2                      SA9211                    FEB  6, 1994, 10:34 AM
  V.2.00:1                      SA9210       PATCH101     FEB  2, 1994,  9:18 AM
  V.2.00:0                                                JAN 22, 1994,  2:45 PM
```

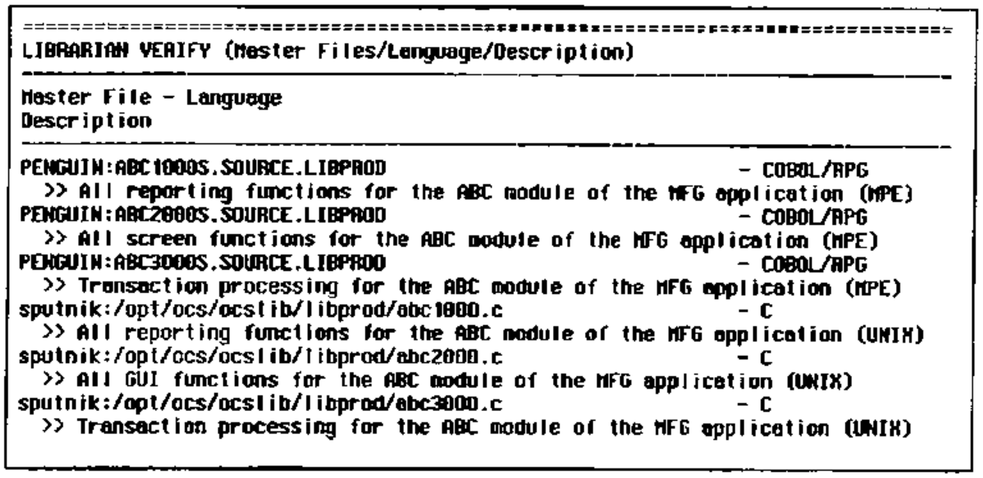
Figure 1-18. VERIFY Format 17, Revision History

Field Descriptions

Master File	Name of the master file including system name.
Revision(s)	A listing of all revisions (including branches) for the file.
Project	Project with which this revision is associated.
Tag	Tag associated with this revision.
Date/Time	The date and time the revision was created.

VERIFY (continued)

Format 18: Language/Description



```
================================================================================
LIBRARIAN VERIFY (Master Files/Language/Description)
--------------------------------------------------------------------------------
Master File - Language
Description
--------------------------------------------------------------------------------
PENGUIN:ABC1000S.SOURCE.LIBPROD                              - COBOL/RPG
  >> All reporting functions for the ABC module of the MFG application (MPE)
PENGUIN:ABC2000S.SOURCE.LIBPROD                              - COBOL/RPG
  >> All screen functions for the ABC module of the MFG application (MPE)
PENGUIN:ABC3000S.SOURCE.LIBPROD                              - COBOL/RPG
  >> Transaction processing for the ABC module of the MFG application (MPE)
sputnik:/opt/ocs/ocslib/libprod/abc1000.c                    - C
  >> All reporting functions for the ABC module of the MFG application (UNIX)
sputnik:/opt/ocs/ocslib/libprod/abc2000.c                    - C
  >> All GUI functions for the ABC module of the MFG application (UNIX)
sputnik:/opt/ocs/ocslib/libprod/abc3000.c                    - C
  >> Transaction processing for the ABC module of the MFG application (UNIX)
```

Figure 1-19. VERIFY Format 18, Language/Description

Field Descriptions

File	Name of the master file including system name.
Language	The programming language assigned to the file during AUTOUPDATE or with the SET LANGUAGE command. Language is used for source code annotation, and merge conflict comments.
Description	The file's description as defined on the FA screen.

VERIFY (continued)

Examples

Review information on all the files in your login by typing:

```
>VERIFY @
```

```
>VERIFY *
```

Review the files in REL1.0 of the FINANCE fileset sending the output offline by typing:

```
>VERIFY REL1.0 OF %FINANCE ;LP
```

Review all master and secondary files in the fileset MFG-FILES by typing:

```
>VERIFY %MFG-FILES AT @.@.@.@, %MFG-FILES
```

```
>VERIFY %MFG-FILES AT /*, %MFG-FILES
```

VERSION

Defines, shows, makes obsolete, and deletes versions for an application.

Restrictions

None

Menu Mode

Select the **Version** option from the **Admin** menu. A dialog appears allowing you to specify the application and version ID.

Command Mode Syntax

```
>VERSION appl-id [ ;ID = version-id ]

    [ ;DELETE ]
    [ ;DESCRIPTION = description ]
    [ ;OBSOLETE ]
    [ ;UNOBSOLETE ]
```

Parameters

appl-id	The unique identifier of an application, as defined on the AP screen.
version-id	The unique identifier of the version. The version can include alphabetic, numeric characters including the period (.), hyphen (-), and underscore (_) characters. A version cannot be used more than once for an application. The version is assigned to all the current master files in the application. The version ID can contain a maximum of 16 characters.
DELETE	Deletes a flushed version.
DESCRIPTION = *description*	Specifies a description for a version. The description can contain a maximum of 32 characters.
OBSOLETE	Specifies a defined version as obsolete, so that retained base revisions can be flushed. All previous versions must be obsolete. Use the **OBSOLETE** parameter in conjunction with the application ID and version ID.
UNOBSOLETE	Instructs LIBRARIAN to undo the **OBSOLETE** operation, providing the version has not already been flushed.

Operation

If you specify only the application ID, LIBRARIAN lists all versions for that application.

Define a new version by specifying an application ID and version ID. Once a version is defined, it can be made obsolete if all previous versions are obsolete using the **OBSOLETE** parameter in conjunction with application ID and version ID. Use the **UNOBSOLETE** parameter to undo the obsolete operation. This option is available only if the version has not been flushed, using the FLUSH utility.

VERSION (continued)

Examples

Create a new version for the MFG application by typing:

```
>VERSION MFG;ID=REL2.0;DESCRIPTION=2ND RELEASE
```

Obsolete REL1.0 of the MFG application by typing:

```
>VERSION MFG;ID=REL1.0;OBSOLETE
```

Related Information

See **FLUSH**
Chapter 7, "Versions" in the *LIBRARIAN/iX Administrator's Guide.*

XEQ

Executes a macro or procedure.

Restrictions

A user with read and execute access to the macro or procedure file. Anyone can execute macros located in XEQ.OSCLIB (MPE) or /opt/ocs/ocslib/xeq (UNIX).

Menu Mode

Select the **Execute** or **Procedures** option from the **Macros** menu. A dialog appears allowing you to specify files and parameters.

Command Mode Syntax

Format 1 (OPTION FILES):

>X[EQ] *macro* [**FOR**] *filelist*

  [;**EDIT** = *edit mask*]
  [;**PARMS** = *parmlist*]

Format 2 (NO OPTION FILES):

>X[EQ] macro [*parmlist*]

Parameters

macro	The name of the macro file or loaded procedure.
filelist	The list of filenames to be substituted in the macro during processing. In order to use filelist substitution, OPTION FILES must be specified earlier in the macro. Each command that uses the %%[ ] parameter will execute repeatedly, once for each file to be substituted, or using the LOOP...NEXT control statements.
EDIT = *edit mask*	Allows you to transform authorized filenames into new filenames written to the XEQLIST file. *edit mask* is the edit mask to be used when transforming the filenames. You can use the at sign (@), the question mark (?), the minus sign (-), and the equal sign (=) editing characters. For more information on these edit masks, refer to *Edit Masks* earlier in this chapter.
PARMS = *parmlist*	Allows you to substitute parameters within the macro file. The *parmlist* is a list of positional parameters to be substituted. You can specify a maximum of 100 parameters in a macro in the format %%*n*, where *n* is the parameter number from 0 to 99.

Operation

To execute a macro, enter the macro file or procedure name. If the file cannot be found in your local directory, LIBRARIAN then checks the XEQ group of the OCSLIB (MPE) account or the /opt/ocs/ocslib/xeq path (UNIX). You can omit the word XEQ, and execute any macro by specifying the file or procedure name alone.

XEQ (continued)

Examples

The following is an example of a macro file used to submit source for testing and compiling each program using the MAKE facility.

```
OPTION FILES=ABC-SUBMIT.ABC-MAINT.ABC;ABC-SUBMIT !XEQLIST
ABC-SUBMIT !XEQLIST
LOOP
    MAKE ABCMAKE.PUB.ABCQA,%%[@P.OBJECT.ABCQA]
NEXT
```

Related Information

See Chapter 7, "Macro Control Language Commands"
Chapter 7, "Macros" in the *LIBRARIAN/iX User's Guide*

User Fileset Commands 2

This chapter describes the FMAINT module commands for maintaining user filesets. User filesets allow LIBRARIAN users to group files according to their individual needs.

The following topics are discussed in this chapter:

- Accessing FMAINT
- FMAINT Commands Summary
- FMAINT Commands

For more information on the FMAINT module, refer to Chapter 6, "User Filesets" in the *LIBRARIAN/iX User's Guide.*

Accessing FMAINT

You can access FMAINT by performing one of the following:

1. Enter the FMAINT module by typing FM at the LIBRARIAN prompt, ">".

   From the **FM>** prompt you can issue any FMAINT command, as listed in this chapter. To return to the LIBRARIAN prompt, enter the **FM>EXIT** command .

2. Select **Tools...User Filesets** from the main menu. Then select an option from the menu.

FMAINT Commands Summary

FMAINT provides users with a set of commands to define and maintain user filesets.

Table 2–1 lists the FMAINT commands along with their functions and page references for locating detailed descriptions.

Table 2–1. FMAINT Commands Summary

Command	Function	Page
Fileset Operations		
FM>ADD	Adds a file to a user fileset.	2–3
FM>CREATE	Creates a user fileset.	2–4
FM>DELETE	Deletes a file from a user fileset.	2–5
FM>MAKE	Makes a user fileset public or private.	2–10
FM>PURGE	Purges a user fileset.	2–11
FM>RELATE	Creates a fileset-to-fileset relationship.	2–12
FM>SEVER	Severs a fileset–to–fileset relationship.	2–13
Information		
FM>LIST	Lists all user filesets defined for a user.	2–8
FM>SHOW	Shows the files that are in a user fileset.	2–14
Other Commands		
FM>EX(IT)	Returns to the LIBRARIAN main prompt.	2–6
FM>HE(LP)	Provides help on FMAINT commands.	2–7
FM>LM(AINT)	Invokes the LMAINT maintenance module.	2–9

FMAINT Commands

For each FMAINT command, the following information is provided:

Menu Mode	How to perform the operation from menu mode (if applicable).
Command Mode Syntax	How the command is entered from the command line prompt.
Parameters	Detailed description for each command parameter.
Operation	Basic function and descriptions of the command.
Examples	Example for using the command.
Related Information	Location of related reading material.

FM>ADD

Adds a file or group of files to an existing user fileset.

Menu Mode

Select the **User Filesets...Add** option from the **Tools** menu. A dialog appears allowing you to specify a fileset, files, revision criteria, and options.

Command Mode Syntax

```
FM>ADD filelist TO fileset
      [ ;ALL ]
      [ ;USE MASTERID ]
```

Parameters

filelist	A list of files, as described in *How to Refer to Files* at the beginning of Chapter 1, "Commands".
TO *fileset*	The user fileset to which the files will be added.
ALL	Includes untracked (unknown) as well as tracked files to a user fileset. Without this parameter, only tracked files are added.
USE MASTERID	Identifies the files in the user fileset by their master file location, rather than the secondary location actually specified.

Operation

FM>ADD adds a file or group of files to an existing user fileset.

The **USE MASTERID** parameter allows you to add master files to the user fileset by specifying the secondary location. Secondary copies can then be referenced with the **AT** parameter for file operations.

FM>ADD does not create a new user fileset.

Examples

The following example adds files to the JSTDEVEL user fileset:

MPE
```
FM>ADD @.SOURCE.DEVEL TO JSTDEVEL;ALL
```
UNIX
```
FM>ADD devel/src/* TO JSTDEVEL;ALL
```

Add the master file associated with the specified file to the DEVELOPMENT user fileset by typing:

MPE
```
FM>ADD FILEA.OBJECT.PRODDEVEL TO DEVELOPMENT;USE MASTERID
```
UNIX
```
FM>ADD /usr/devel/object/filea TO DEVELOPMENT;USE MASTERID
```

Related Information

See **FM>CREATE**

FM>CREATE

Creates a user fileset.

Menu Mode

Select the **User Filesets...Create** option from the **Tools** menu. A dialog appears allowing you to specify a fileset, files, revision criteria, and options.

Command Mode Syntax

```
FM>CREATE fileset  [ FROM filelist ]

     [ ;PRIVATE ]
     [ ;PUBLIC ]
     [ ;USE MASTERID ]
```

Parameters

fileset	The name of the user fileset you are creating.
FROM *filelist*	A list of files, as described in *How to Refer to Files* at the beginning of Chapter 1, "Commands".
PRIVATE	Identifies the user fileset as private. Only the creator of the user fileset and the LIBRARIAN manager can change a private fileset.
PUBLIC	Identifies the user fileset as public. Any user can change a public fileset. The default is a public fileset.
USE MASTERID	Identifies the files in the user fileset by their master file location, rather than the secondary location actually specified.

Operation

FM>CREATE establishes a new user fileset. A user fileset is defined by its logical and physical components. When creating a user fileset, you can add files at the same time by using the FROM parameter. However, you may omit the FROM parameter to create an empty fileset. Later, you can link other user filesets (**FM>RELATE**), and/or add files (**FM>ADD**). Use the **FM>MAKE** command to change the public/private designation.

Examples

Create a public user fileset named DEVELOPMENT containing no members by typing:

```
FM>CREATE DEVELOPMENT
```

Create a private user fileset with an initial set of files by typing:

MPE
```
FM>CREATE RMTDEVEL FROM @.JCL.APDEVEL;PRIVATE
```
UNIX
```
FM>CREATE RMTDEVEL FROM /usr/apdevel/jcl/*;PRIVATE
```

Related Information

See **FM>ADD**
FM>MAKE
FM>RELATE

FM>DELETE

Deletes one or more files from a user fileset. The files are not physically deleted from the system.

Menu Mode

Select the **User Filesets...Delete** option from the **Tools** menu. A dialog appears allowing you to specify a fileset, files, revision criteria, and options.

Command Mode Syntax

FM>DELETE *filelist* **FROM** *fileset*

Parameters

filelist	A list of files, as described in *How to Refer to Files* at the beginning of Chapter 1, "Commands".
FROM *fileset*	The user fileset from which you are deleting the file.

Operation

FM>DELETE terminates the relationship between physical file(s) and a user fileset; however, it does not purge the file from the system or the database.

Use the **FM>SEVER** command to terminate a relationship between two user filesets.

Examples

Remove a file from the MYFILES user fileset by typing:

```
MPE   FM>DELETE FILEA.JCL.APDEVEL FROM MYFILES
UNIX  FM>DELETE /usr/apdevel/jcl/filea FROM MYFILES
```

Related Information

See **FM>SEVER**

FM>EXIT

Leaves the FMAINT module and returns to the LIBRARIAN prompt.

Command Mode Syntax

FM>EX[IT]

Parameters

None

Operation

FM>EXIT exits from the FMAINT module.

Example

Exit FMAINT and return to the LIBRARIAN prompt by typing:

```
FM>EXIT
```

FM>HELP

Accesses online help for information on FMAINT commands.

Command Mode Syntax

```
FM> HE[LP] [ command ]
```

Parameters

command	Any FMAINT command name.

Operation

HELP invokes LIBRARIAN online help. By specifying any FMAINT command, you will be provided information for that command. If you do not specify any parameter with HELP, it provides you with an overview of FMAINT and its commands.

Examples

Obtain help on using the **FM>SHOW** command by typing:

```
FM>HELP SHOW
```

FM>LIST

Displays the user filesets defined for a particular user.

Menu Mode

Select the **User Filesets...List** option from the **Tools** menu. A dialog appears to let you specify a user.

Command Mode Syntax

FM>LIST [[**FOR USER** [=]] *user*]

Parameters

FOR USER	Specifies the user whose filesets you want to display.
user	The user whose filesets you want to list. The default is the current active user.

Operation

FM>LIST displays the user filesets created by a specific user. The information displayed includes the public/private designation of each user fileset.

Examples

List the user filesets that have been defined for the OPER user by typing:

```
FM>LIST FOR USER=OPER
```

Optionally, you can list the user filesets defined for the OPER user by typing:

```
FM>LIST OPER
```

FM>LMAINT

Accesses the LMAINT listfile creation and maintenance module.

Command Mode Syntax

FM>LM[AINT]

Parameters

None

Operation

FM>LMAINT terminates the FMAINT module and invokes the LMAINT module.

Examples

Access the LMAINT module by typing:

```
    FM>LMAINT
```

FM>MAKE

Changes the public/private designation on a user fileset. Only the creator of the user fileset and the LIBRARIAN manager can make a private user fileset public.

Menu Mode

Select the **User Filesets...Make Public/Private** option from the **Tools** menu. A dialog appears allowing you to specify a fileset and public/private flag.

Command Mode Syntax

$$\text{FM>MAKE } \textit{fileset} \begin{Bmatrix} \text{PUBLIC} \\ \text{PRIVATE} \end{Bmatrix}$$

Parameters

fileset	Fileset you want to designate as public or private.
PUBLIC	Designates the user fileset as public. Any user can change a public user fileset.
PRIVATE	Designates the user fileset as private. Only the creator of the user fileset and the LIBRARIAN manager can change a private user fileset.

Operation

FM>MAKE allows you to change the public/private designation of a user fileset. For example, you can create a private user fileset so that no one can change the fileset. Later, you can make the fileset public to allow other users to change it.

You can initially set the public/private designation when creating the user fileset.

Examples

Change the APDEVEL user fileset to public by typing:

```
FM>MAKE APDEVEL PUBLIC
```

FM>PURGE

Removes a user fileset from the database.

Menu Mode

Select the **User Filesets...Purge** option from the Tools menu. A dialog appears allowing you to specify a fileset and options.

Command Mode Syntax

```
FM>PURGE fileset
    [ ;EXPLODE ]
```

Parameters

fileset	The user fileset you want to purge.
EXPLODE	Purges all filesets related to and including the specified fileset.

Operation

FM>PURGE purges all references to the user fileset definition from the database. However, this command does not purge the files physically from the LIBRARIAN database or from the system.

If you issue **FM>PURGE** without the **EXPLODE** parameter, you will delete references to the physical file members that directly belong to the user fileset. By using **EXPLODE**, you will delete references to all of the physical file members and all of the component user filesets with their physical file members.

Examples

Purge the MYFILES user fileset by typing:

```
FM>PURGE MYFILES
```

Purge the DEVELOPMENT user fileset and all its component user filesets by typing:

```
FM>PURGE DEVELOPMENT ;EXPLODE
```

FM>RELATE

Creates a subset relationship between two user or project filesets.

Menu Mode

Select the **User Filesets...Relate** option from the **Tools** menu. A dialog appears allowing you to specify a parent fileset and its subset.

Command Mode Syntax

FM>RELATE *subset* TO *parent-fileset*

Parameters

subset	The user fileset that you want to make a subset of another user fileset.
TO *parent-fileset*	The parent user fileset.

Operation

FM>RELATE establishes a relationship between two user filesets. It lets you build a user fileset hierarchy similar to the hierarchy of an application fileset. When you relate user filesets, you can issue individual LIBRARIAN commands for any subset, parent set, or physical file.

Additionally, you can use **FM>RELATE** to create *project* hierarchies. That is, this command enables you to relate project filesets to other project filesets. As a result, when you check out files that belong to a project, these files automatically belong to any parent project fileset. You can then perform checkins, approvals and/or distribution by referring to parent project filesets.

Transactions are logged under the parent fileset. When you refer to them in commands, however, the last project for the file reflects the actual project name at the time of checkout.

Examples

Designate the MYFILES user fileset as a component of the JSTDEVEL user fileset by typing:

```
FM>RELATE MYFILES TO JSTDEVEL
```

FM>SEVER

Terminates a subset relationship between two user filesets.

Menu Mode

Select the **User Filesets...Sever** option from the **Tools** menu. A dialog appears allowing you to specify a parent fileset and its subset.

Command Mode Syntax

FM>SEVER *subset* **FROM** *parent-fileset*

Parameters

subset	The name of the user fileset that you no longer wish to be a subset of a parent user fileset.
FROM *parent-fileset*	The parent user fileset.

Operation

FM>SEVER terminates the relationship between two user filesets. If you would like to remove a user fileset from its hierarchy, use this command to terminate the relationship. Both user filesets remain in the database and will be independent of each other. Use the **FM>PURGE** command to remove a user fileset from the database.

Examples

Sever the relationship between the DEVELOPMENT and RTMDEVEL user filesets by typing:

```
FM>SEVER DEVELOPMENT FROM RTMDEVEL
```

Related Information

See **FM>PURGE**

FM>SHOW

Displays the files and subsets that comprise a user fileset.

Menu Mode

Select the **User Filesets...Show** option from the **Tools** menu. A dialog appears allowing you to specify a fileset and options.

Command Mode Syntax

```
FM>SHOW fileset

    [ ;EX[PLODE] ]
    [ ;STR[UCTURE] ]
```

Parameters

fileset	The user fileset that you want to review.
EXPLODE	Explodes the user fileset to include all of the files and filesets that relate to this user fileset. Each fileset shows the level relative to the fileset you are reviewing.
STRUCTURE	Shows the hierarchy of filesets related to this user fileset, but does not show filenames.

Operation

FM>SHOW displays the file structure for any user fileset. Without the **EXPLODE** parameter, this command lists only the physical file members that directly belong to the user fileset. **EXPLODE** lists all physical file members and all component user filesets with their physical file members. For each fileset, the level of the fileset within the hierarchy is also shown.

Examples

Show the file structure of the MYFILE user fileset by typing:

```
FM>SHOW MYFILE
```

Show the file structure of the APDEVEL user fileset and all of its related filesets by typing:

```
FM>SHOW APDEVEL;EXPLODE
```

Listfile Maintenance Commands 3

This chapter describes the LMAINT listfile maintenance module.

LMAINT can create listfiles based on multiple selection criteria. Listfiles are files that contain a list of filenames, with or without a system name as a further qualification.

Listfiles can be used in LIBRARIAN commands as a way to refer to files, and must be preceded with an exclamation point (!) or a caret (^). Listfiles can be used as indirect store lists for MPE's **STORE** and **RESTORE** commands. Many other MPE and UNIX system utilities also accept indirect files as input.

The following topics are covered in this chapter:

- Accessing LMAINT
- FMAINT Commands Summary
- FMAINT Commands

For more information about LMAINT, refer to Chapter 7, "Listfiles" in the *LIBRARIAN/iX User's Guide.*

Accessing LMAINT

You can access LMAINT in the following ways:

1. Issue the LMAINT command from the LIBRARIAN > prompt.

   From the LM> prompt, you can issue any LMAINT command. To return to the LIBRARIAN prompt, enter **LM>EXIT**.

2. Select **Tools** from the main menu. Then select **Listfiles...** for a menu of LMAINT operations.

LMAINT Commands Summary

Table 3–1 below lists the LMAINT commands as well as their functions and page references for locating detailed descriptions.

Table 3-1 LMAINT Commands Summary

Command	Function	Page
Listfile Operations		
LM>ALTER †	Changes APPEND/REPLACE mode of a listfile.	3–3
LM>DOCUMENT †	Associates documentation notes with a listfile.	3–4
LM>EDIT †	Edits a listfile using EDITOR.	3–5
LM>LIST †	Lists the filenames contained in the listfile.	3–9
LM>OUTPUT	Selects files and writes them to a listfile.	3–10
LM>REPORT †	Formats the documentation notes and filenames.	3–14
LM>S(ORT)	Sorts a listfile by filename.	3–15
Other Operations		
LM>EX(IT)	Returns to the LIBRARIAN main prompt.	3–6
LM>FM(AINT)	Invokes the FMAINT module.	3–7
LM>HE(LP)	Provides online help on LMAINT commands	3–8
† = MPE only		

LMAINT Commands

The LMAINT command descriptions and syntax include the following:

Menu Mode	How to perform the operation from menu mode (if applicable).
Command Mode Syntax	How the command is entered from the command line prompt.
Parameters	Detailed description for each command parameter.
Operation	Basic function and description of the command.
Example(s)	Example(s) of the command.
Related Information	Locations of related information.

LM>ALTER

Alters the mode of a listfile to either **APPEND** or **REPLACE**.

Menu Mode

Select the Listfiles...Alter option from the Tools menu. A dialog appears allowing you to specify a listfile name and its **APPEND/REPLACE** attribute.

Command Mode Syntax

LM>ALTER *filename*

> [;**APPEND**]
> [;**REPLACE**]

Parameters

filename	Any valid listfile.
APPEND	Sets the listfile to **APPEND** mode. When set to **APPEND**, all subsequent **LM>OUTPUT** transactions to the listfile are automatically appended.
REPLACE	Sets the listfile to **REPLACE** mode. By default, the **LM>OUTPUT** command replaces the contents of the listfile (unless no files are found).

Operation

LM>ALTER allows you to toggle between **APPEND** and **REPLACE** mode for a listfile. When first created, a listfile is in the REPLACE mode.

Examples

Change the default mode to **APPEND** by typing:

```
LM>ALTER MYFILE;APPEND
```

Related Information

See **LM>OUTPUT**

LM>DOCUMENT

Associates documentation with a listfile.

Menu Mode

Select the **Listfiles...Document** option from the **Tools** menu. A dialog appears allowing you to specify a listfile name.

Command Mode Syntax

```
LM>DOCUMENT filename
```

Parameters

filename	Any valid listfile.

Operation

LM>DOCUMENT lets you create documentation for a listfile. **LM>DOCUMENT** invokes the standard HP editor program, **EDITOR.PUB.SYS**. Existing documentation is brought automatically into the editor. New or modified documentation is stored automatically when you exit.

Documentation is stored in the user label area of the listfile; therefore, you are limited to a maximum of 750 80-character lines.

Two special directives can be embedded in your documentation which affects the **LM>REPORT** display:

.title.	When .title. is encountered starting in the first column of a line, the **LM>REPORT** command substitutes the character sequence.
.page.	When .page. is encountered starting in the first column of a line, the **LM>REPORT** command issues a page break.

Examples

Document a listfile called MYFILE by typing:

```
LM>DOCUMENT MYFILE
```

Related Information

See **LM>REPORT**

LM>EDIT

Edits the contents of a listfile using **EDITOR.PUB.SYS**.

Menu Mode

Select the Listfiles...Edit option from the Tools menu. A dialog appears allowing you to specify a listfile name.

Command Mode Syntax

LM>EDIT *filename*

Parameters

filename	Any valid listfile.

Operation

LM>EDIT lets you edit the contents of a listfile with **EDITOR.PUB.SYS**. The listfile is automatically brought into the editor. You can manually add, delete, or change filenames that appear in this file.

Use the **LM>OUTPUT** command to have LIBRARIAN automatically create or add to a listfile.

Examples

Edit a listfile called MYFILE by typing:

```
LM>EDIT MYFILE
```

Related Information

See **LM>OUTPUT**

LM>EXIT

Terminates the LMAINT module and returns to the LIBRARIAN prompt.

Command Mode Syntax

```
LM>EX[IT]
```

Parameters

None

Operation

LM>EXIT terminates the LMAINT module.

Example

Exit LMAINT and return to the LIBRARIAN prompt by typing:

```
    LM>EXIT
```

LM>FMAINT

Accesses the FMAINT user fileset definition and maintenance module.

Command Mode Syntax

```
LM>FM[AINT]
```

Parameters

None

Operation

LM>FMAINT terminates the LMAINT module and invokes the FMAINT module.

Example

Access the FMAINT module by typing:

```
LM>FMAINT
```

Related Information

See Chapter 2, "User Fileset Commands"

LM>HELP

Accesses online help for information on using LMAINT.

Command Mode Syntax

```
LM>HE[LP] [ command ]
```

Parameters

command	Any LMAINT command name.

Operation

HELP provides you with the LIBRARIAN online help. If you specify an LMAINT command, you will be given information for that command. If you do not specify any parameters, HELP provides you with an overview of LMAINT and the LMAINT commands.

Example

Obtain help on using the **LM>SORT** command by typing:

```
LM>HELP SORT
```

LM>LIST

Displays the filenames contained in a listfile.

Menu Mode

Select the **Listfiles...List** option from the **Tools** menu. A dialog appears allowing you to specify a listfile name.

Command Mode Syntax

LM>LIST *filename*

  [**FMT**]

Parameters

filename	Any valid listfile.
FMT	Allows you to view the list in a different format.

Operation

LM>LIST displays the filenames contained in the specified listfile on a page-by-page basis (36 files per page). Additionally, you can view the filenames in one of two formats by selecting the **FMT** option.

Examples

List the contents of a listfile named MYFILE by typing:

```
LM>LIST MYFILE
```

Related Information

See **LM>REPORT**

LM>OUTPUT

Outputs a list of filenames to a new or existing listfile.

Menu Mode

Select the **Listfiles...Output** option from the **Tools** menu. A dialog appears allowing you to specify a listfile name, files, selection criteria, revision criteria, and options.

Command Mode Syntax

```
LM>OUTPUT filelist TO listfilename

     [ ;ALL ]
     [ ;APPEND ]
     [ ;AUTHORIZE stepname ]
                  (  <  )
                  (  <= )
     [ ;EXPDATE   {  =  }  expdate ]
                  (  >= )
                  (  >  )
     [ ;FILECODE = filecode ]

                  (  <  )
                  (  <= )  MODDATE (filename) ]
     [ ;MODDATE   {  =  }  moddate ]
                  (  >= )  TIMESTAMP (filename) ]
                  (  >  )
     [ ;MODIFIED ]
     [ ;OLDNAME ]
     [ ;OWNER user ]
     [ ;REPLACE ]
     [ ;RESETONZERO ]
     [ ;SIMULATE stepname ]
     [ ;SYSTEM ]
     [ ;UNMODIFIED ]
     [ :UNTRACKED ]
     [ ;USE stepname ]
```

Parameters

filelist	A list of files, as described in *How to Refer to Files* at the beginning of Chapter 1, "Commands".
listfilename	The name of the listfile to which qualifying filenames should be written. It can be any valid local filename. If the file already exists, it must be a valid listfile.
	Listfiles are assigned a filecode of 55 (or 56 when the **SYSTEM** parameter is specified).
ALL	Indicates that **LM>OUTPUT** should include all files found, regardless of whether or not they are being tracked by LIBRARIAN.

LM>OUTPUT (continued)

Parameters, continued

APPEND	Appends the qualifying filenames to those already found in the listfile. Takes precedence over the default mode assigned to the file by using the **LM>ALTER** command.
AUTHORIZE	Creates a list of files that would be authorized for a particular step.
EXPDATE	Indicates that files selection is based on the expiration date assigned by LIBRARIAN.
expdate	Any valid date, expressed in accordance with the Date Format established on the System Profile (SP) screen. TODAY is a valid mnemonic for the current system date.
FILECODE	Indicates that files selection is based on the specified filecode, where *filecode* is any valid (numeric) filecode.
SYSTEM	Indicates that the filenames written to the listfile should include the system ID. By default, filenames do not include the system name. If **SYSTEM** is specified, filenames are written in the format: *system:filename* Listfiles that are created with a **SYSTEM** specification are assigned a filecode of 56 rather than 55.
MODDATE	Indicates that selection of files is based on the Date Modified (and optionally Time Modified) value stored in the file label. *moddate* is any valid date, expressed in accordance with the Date Format established on the SP screen. TODAY is a valid mnemonic for the current system date.
MODDATE (*filename*)	Selection is based on a comparison made with the Date Modified value of the specified filename.
TIMESTAMP	Selection is based on a comparison made with both the Date Modified and Time Modified values of the specified filename. *filename* is any valid filename, expressed in the format: *system:filename*
MODIFIED	Selects only those files modified since they were created by the LIBRARIAN program. The current timestamp in the file label is compared with the timestamp in the database.
OLDNAME	Indicates that the filename appearing in the listfile should be the file's original name, before it was retained (for retained files only).
OWNER	Selects files that the specified user owns.
user	Any valid LIBRARIAN user.
REPLACE	Does not append new filenames to the existing listfile. Takes precedence over the default mode assigned to the file by the **LM>ALTER** command. The opposite of **APPEND**.

LM>OUTPUT (continued)

Parameters, continued

RESETONZERO	Instructs **LM>OUTPUT** to replace an existing listfile with an empty listfile whenever no files qualify. By default, an empty fileset will not result in an empty listfile. The existing listfile is not replaced.
SIMULATE	Creates a list based on destination files authorized for a particular step. Similar to **USE** and **AUTHORIZE**.
UNMODIFIED	Selects only those files that were not modified since they were created by LIBRARIAN.
UNTRACKED	Selects only files not currently being tracked by LIBRARIAN.
USE	Creates a list based on the destination location defined for a step, including any refined destinations. *stepname* is the name of the step in the following format: *step*[.*route*[.*appl*]] Unlike the **SIMULATE** parameter, **USE** includes files regardless of whether or not they would be authorized.

Operation

LM>OUTPUT serves several purposes. First, it provides a way of creating an indirect store file (see the MPE's STORE command) from an existing LIBRARIAN fileset. This indirect store file can be updated later to include documentation notes about the files it references.

Second listfiles created by **LM>OUTPUT**, can in turn, be used as input to any valid LIBRARIAN command preceded with the exclamation point (!) or caret (^). For example, a listfile called MYFILE can be used as input to a user-defined step called CHECKOUT as follows:

```
>CHECKOUT !MYFILE
```

Examples

The following example creates a listfile containing the filenames found in a project fileset.

MPE

```
LM>OUTPUT %P1-FILES TO P1FILES.STORFILE
```

UNIX

```
LM>OUTPUT %P1–FILES TO ~/store/listfiles/proj1files
```

The following example transforms the filenames found in the project fileset into the names they would contain after a DISTRIBUTE step was performed.

MPE

```
LM>OUTPUT %P1-FILES TO P1FILES.STORFILE;SIMULATE DISTRIBUTE
```

UNIX

```
LM>OUTPUT %P1–FILES TO ~/store/list/proj1files ;SIMULATE; DISTRIBUTE
```

LM>OUTPUT (continued)

Parameters, continued

Append to a listfile the files involved in your last LIBRARIAN transaction by typing:

```
LM>OUTPUT * TO MYFILES.MYGROUP;APPEND
```

```
LM>OUTPUT ** TO myfiles; APPEND
```

The following example creates a listfile containing all files modified since the listfile itself was last modified. It includes any file, even those that are not tracked by LIBRARIAN. It then recreates the listfile if no files qualify.

```
LM>OUTPUT @.SOURCE TO MODFILES; MODDATE>TIMESTAMP(MODFILES);&
ALL;RESETONZERO
```

```
LM>OUTPUT src/* TO modfiles; MODDATE>TIMESTAMP(modfiles);ALL;RESETONZERO
```

Related Information

See **LM>DOCUMENT**

LM>REPORT

Displays the documentation notes and filenames contained in the listfile.

Menu Mode

Select the **Listfiles...Report** option from the **Tools** menu. A dialog appears allowing you to specify a listfile.

Command Mode Syntax

LM>REPORT *filename*

Parameters

filename	Any valid listfile.

Operation

LM>REPORT displays information pertinent to the specified listfile, including creation and modification dates. In addition, any associated documentation notes are also formatted and displayed on a page-by-page basis.

Filenames contained in the listfile are optionally displayed.

Examples

The following example reports the documentation notes and filenames contained in the listfile called MYFILE.

```
LM>REPORT MYFILE
```

Related Information

See **LM>DOCUMENT**

LM>SORT

Sorts the filenames stored in the listfile.

Menu Mode

Select the **Listfiles...Sort** option from the **Tools** menu. A dialog appears allowing you to specify a listfile and options.

Command Mode Syntax

```
LM>S[ORT] filename
    [ ;NODUPLICATES ]
```

Parameters

filename	Any valid listfile.
NODUPLICATES	Eliminates duplicate filenames from sorted output.

Operation

LM>SORT sorts the contents of a listfile by filename in ascending order.

Example

The following example sorts a listfile called MYFILE and eliminates duplicate filenames.

```
LM>SORT MYFILE;NODUPLICATES
```

SHOWLOG Commands 4

SHOWLOG is a comprehensive and flexible tool for generating transaction reports for meeting a variety of audit needs. Additionally, SHOWLOG provides a transaction memo editing function. It allows users to inspect transaction log records selectively in a variety of formats.

SHOWLOG has its own set of commands to set up selection criteria, establish report format and sort sequence, and process the report online or offline.

This chapter covers the following topics:

- Accessing SHOWLOG
- Transaction Codes on SHOWLOG Reports
- SHOWLOG Commands

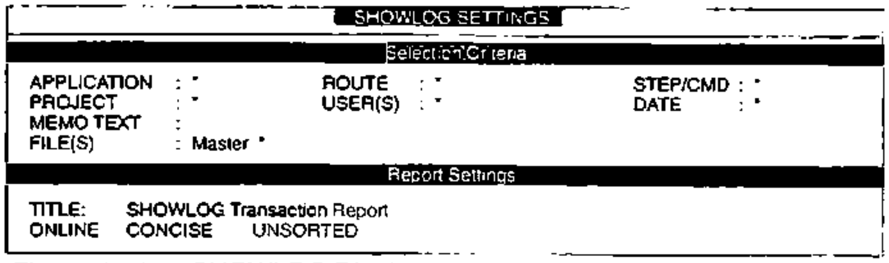
Accessing SHOWLOG

You can access the SHOWLOG module in the following ways:

1. Type **SHOWLOG** at the LIBRARIAN > prompt.
2. Select **Info** from the main menu followed by Log. Then, select **SHOWLOG** to access the SHOWLOG module.

The default selection criteria and report settings are displayed, followed by the SHOWLOG> prompt. From this prompt you can issue any SHOWLOG command until you enter **SHOWLOG>EXIT**.

Figure 4–1 shows the SHOWLOG display after entering the SHOWLOG module.



```
                              SHOWLOG SETTINGS
                             Selection Criteria
APPLICATION  : *          ROUTE    : *              STEP/CMD : *
PROJECT      : *          USER(S)  : *              DATE     : *
MEMO TEXT    :
FILE(S)      : Master *
                              Report Settings
TITLE:   SHOWLOG Transaction Report
ONLINE   CONCISE    UNSORTED
```

Figure 4–1. SHOWLOG Display

Transaction Codes on SHOWLOG Reports

Table 4–1 lists the codes that appear on transaction reports generated from the LIBLOG database (i.e., RTS10, RTD10, RTD40, and SHOWLOG reports).

Table 4–1. LIBLOG Transaction Codes

Transaction Code	Command	Transaction Code	Command
CO	COMPRESS	SC	SECURE
CP	COPY	SR	SCAN + REPLACE
DC	DECOMPRESS	ST	SET
LO	LOCK	TO	TOUCH
ME	MEMO	UL	UNLOCK
MV	MOVE	UP	UPDATE
OR	ORPHAN	XC	XCOMPRESS
OV	OVERLAY	XD	XDECOMPRESS
PF	PERFORM *(step)*	XM	XMOVE
PU	PURGE	XP	XPURGE
RE	RESET	XR	XRENAME
RL	RELEASE	XT	XTOUCH
RN	RENAME	XY	XCOPY
RS	RESTORE		

SHOWLOG Commands Summary

SHOWLOG commands allow users to define selection criteria and to set up report parameters. Use SHOWLOG commands to:

- define transaction selection criteria
- choose a report format
- change the output destination
- set up the report sort sequence
- save and retrieve report settings
- generate customized transaction reports
- select subsets of extracted transactions

Table 4–2 lists the SHOWLOG commands and their functions. Page references help you locate the detailed command descriptions.

Table 4–2 SHOWLOG Command Summary

Command	Function	Page
Selection Criteria		
SHOWLOG>SE(LECT)	Extracts only those transaction records which match the specified selection criteria.	4–17
Report Format		
SHOWLOG>FO(RMAT)	Changes report format.	4–7
SHOWLOG>LI(ST)	Creates a listfile containing names of files involved in selected transactions.	4–12
SHOWLOG>OU(TPUT)	Sets the report output disposition to offline or online.	4–13
Sort Sequence		
SHOWLOG>SO(RT)	Sets up the reports sort sequence.	4–23
Report Settings		
SHOWLOG>GE(T)	Processes commands from a file.	4–9
SHOWLOG>SA(VE)	Saves current report settings and selection criteria to a file.	4–16
Generate Reports		
SHOWLOG>GO	Generates reports using current report selection criteria and settings.	4–10
Subsets		
SHOWLOG>SUB(SET)	Selects a subset of currently extracted transactions for reporting.	4–24
SHOWLOG>UN(DO)	Resets the current subset.	4–26
Other Commands		
SHOWLOG>EX(IT)	Exits the SHOWLOG module, and returns to the LIBRARIAN prompt.	4–5
SHOWLOG>FL(USH)	Deletes all log records associated with extracted transactions.	4–6
SHOWLOG>HE(LP)	Accesses the online help for information about using SHOWLOG.	4–11
SHOWLOG>RED(O)	Edits the previous command entry.	4–14
SHOWLOG>RES(ET)	Resets selection criteria and/or report settings to default values.	4–15
SHOWLOG>SH(OW)	Refreshes the display of selection criteria and report settings.	4–22
SHOWLOG>TI(TLE)	Sets a title to appear on all pages of a report.	4–25

SHOWLOG Commands

This section contains descriptions and syntax for all SHOWLOG commands. The SHOWLOG command descriptions include:

Syntax	How the command is entered from the command line prompt.
Parameters	Detailed description of each command parameter.
Operation	Basic function and description of the command.
Example(s)	Example(s) of the command.
Related Information	Location of related information.

SHOWLOG>EXIT

Exits the SHOWLOG module and returns to the LIBRARIAN prompt.

Syntax

SHOWLOG>EX[IT]

Parameters

None

Operation

SHOWLOG>EXIT returns to the LIBRARIAN prompt.

Examples

Exit SHOWLOG and return to the LIBRARIAN prompt by typing:

```
SHOWLOG>EXIT
```

SHOWLOG>FLUSH

Deletes all log records associated with extracted transactions.

Syntax

```
SHOWLOG>FL[USH]
```

Parameters

None

Operation

SHOWLOG>FLUSH is available only to users with LIBRARIAN Manager capability. When issuing this command, you are prompted for a user and password.

Note All LIBRARIAN users and passwords are case-sensitive.

If a subset is currently active, only the subset of extracted transactions is deleted.

Examples

Flush all extracted log transaction records from the LIBLOG database by typing:

```
SHOWLOG>FLUSH
```

Related Information

See **FLUSHLOG**

SHOWLOG>FORMAT

Changes report format.

Syntax

```
SHOWLOG> [ FO[RMAT] ] ⎧ ALL       ⎫
                      ⎪ CONCISE   ⎪
                      ⎪ DETAIL    ⎪
                      ⎪ MEMO      ⎪
                      ⎨ MEMO EDIT ⎬
                      ⎪ RTD10     ⎪
                      ⎪ RTD40     ⎪
                      ⎪ RTS10     ⎪
                      ⎩ SUMMARY   ⎭
```

Parameters

ALL	Changes the report format to **ALL**. The **ALL** format includes summary data, memo text, and detailed data for qualifying transaction records.
CONCISE	Changes the report format to **CONCISE** (default). The **CONCISE** format is a single line summary for each transaction. Reference numbers for subset selection are available with this format by pressing CONTROL-Y. This toggles between dates and reference numbers for each display page of the **CONCISE** format.
DETAIL	Changes the report format to **DETAIL**. Transaction summary and detail information is included. If master file is defined as the primary sort key with the **SORT** command, then a different layout is used.
MEMO	Changes the report format to **MEMO**. The **MEMO** format includes concise transaction summary information and memo text.
MEMO EDIT	Changes the report format to **MEMO** and enables the memo editing facility. After each transaction summary and memo text is displayed, you are asked if you want to edit the memo text. To edit the memo, you must supply the password of the user who performed the transaction. Once the password is validated, you can edit any of that user's memos without being prompted again. You can replace or append a memo. Editing memo text is similar to adding a memo during a LIBRARIAN transaction.
RTD10	Changes the report format to RTD10 (Transaction Detail by Date, Time). For more information on this report, refer to Chapter 6, "Reports." When transaction records are extracted, they are sorted and sent to the RTD10 report program. The output is set automatically to OFFLINE, and the sort sequence is set automatically to DATE, TIME.
RTD40	Changes the report format to RTD40 (Transaction Detail Report by File, Date, Time). For more information on this report, refer to Chapter 6, "Reports." When transaction records are extracted, they are sorted and sent to the RTD40 report program. The output is set automatically to OFFLINE, and the sort sequence is set automatically to MASTERFILE, DATE, TIME.

SHOWLOG>FORMAT (continued)

Parameters, continued

- If you use the default sort, the master filename appears in the first column and the time is omitted.
- If you sort by DATE, the transaction time and date appears .

RTS10	Changes the report format to RTS10 (Transaction Summary Report by Date, Time). For more information on this report, refer to Chapter 6, "Reports." When transaction records are extracted, they are sorted and sent to the RTS10 report program. The output is set automatically to OFFLINE, and the sort sequence is set automatically to DATE, TIME.
SUMMARY	Changes the report format to **SUMMARY**. The **SUMMARY** format includes a three-line summary of each transaction.

Operation

SHOWLOG>FORMAT changes the currently active report format.

If your extract selection criteria and sort sequence have not changed since generating the last report, you can use this command in conjunction with the **GO** command. This causes the formats to alternate without waiting for a new extract or sort.

Generate a report in the selected format by using **GO**.

You can omit the word FORMAT and execute this command by specifying the parameter alone.

Examples

Choose the **ALL** report format by typing:

```
SHOWLOG>FORMAT ALL
```

Choose the **MEMO** format with the memo editing facility enabled by typing:

```
SHOWLOG>MEMO EDIT
```

Related Information

See **SHOWLOG>GO**
RTD10, RTD40, and RTS10 reports in Chapter 6, "Reports"

SHOWLOG>GET

Processes the commands in a file.

Syntax

```
SHOWLOG>GE[T] xeqfile
```

Parameters

xeqfile	The name of a file consisting of SHOWLOG commands. You must have READ access to the file.

Operation

Use **SHOWLOG>GET** to process a **SHOWLOG>SAVE** file with stored settings, or to process commands in a file you created by using an editor. **SHOWLOG** displays each command in the file as it is executed. If an error is encountered, or if you press CONTROL-Y, processing of the file can be terminated.

Examples

Restore settings that were saved earlier with the **SAVE** command in a file called SETTINGS by typing:

```
    SHOWLOG>GET SETTINGS
```

Related Information

See **SHOWLOG>SAVE**

SHOWLOG>GO

Generates reports using current report selection criteria and settings.

Syntax

```
SHOWLOG>GO
```

Parameters

None

Operation

SHOWLOG>GO initiates report processing:

- If selection criteria are new or have changed, an extract occurs. Use CONTROL-Y at any time during the extract to view the number of records that have been extracted so far. You have the option of terminating the extract.
- If a new extract has been performed or the sort sequence has changed, the extract file is sorted. Use CONTROL-Y to check the sort status and to discontinue, if desired.

Examples

Process the currently defined report by typing:

```
    SHOWLOG>GO
```

SHOWLOG>HELP

Accesses the online help for information on using SHOWLOG.

Syntax

```
SHOWLOG>HE[LP] [ command ]
```

Parameters

command	Any **SHOWLOG** command name, or one of the **SELECT**, **OUTPUT**, or **FORMAT** parameters.

Operation

When you issue **SHOWLOG>HELP** and specify a SHOWLOG command name, you are provided with information for that command. Without parameters specified, **HELP** provides you with an overview of SHOWLOG commands.

Examples

Obtain help on using the **SORT** command by typing:

```
SHOWLOG>HELP SORT
```

SHOWLOG>LIST

Creates a listfile containing the names of files involved in selected transactions.

Syntax

```
SHOWLOG>LI[ST] { MASTERS   } TO listfile
               { FROMFILES }
               { TOFILES   }

    [ ;APPEND ]
    [ ;SYSTEM ]
```

Parameters

MASTERS	Lists master filenames to the listfile.
FROMFILES	Lists source filenames to the listfile.
TOFILES	Lists destination filenames to the listfile.
listfile	Any valid filename within your login account.
APPEND	Appends records to an existing listfile.
SYSTEM	Includes the system names in the listfile.

Operation

When you issue **SHOWLOG>LIST**, the files involved in selected transactions (MASTER, FROM, or TO) are written to the specified listfile.

Examples

The following example appends destination filenames to a listfile called MYLIST.

```
SHOWLOG>LIST TOFILES TO MYLIST; APPEND
```

SHOWLOG>OUTPUT

Sets the report output disposition to offline or online.

Syntax

```
SHOWLOG>[ OU[TPUT] ]  { ON[LINE]  }
                      { OF[FLINE] }
```

Parameters

OFFLINE	Redirects SHOWLOG output to the LP device (i.e., :FILE SHLOGOUT;DEV=LP).
ONLINE	Redirects SHOWLOG output to the $STDLIST device (i.e., :FILE SHLOGOUT=$STDLIST). **ONLINE** is the default.

Operation

SHOWLOG>OUTPUT sets up a file equation for the formal file designator SHLOGOUT. To override the file equation set up by **OUTPUT**, issue a file equation for LIBOUT (e.g., :FILE LIBOUT;DEV=LASER). LIBOUT always takes precedence over SHLOGOUT.

You may omit the word OUTPUT and execute this command by specifying only **ONLINE** or **OFFLINE**.

Examples

Set the report destination so that reports are sent to the line printer by typing:

```
SHOWLOG>OUTPUT OFFLINE
```

Set the report destination so that reports are sent on the STDLIST by typing:

```
SHOWLOG>ONLINE
```

SHOWLOG>REDO

Edits the last command entered.

Syntax

```
SHOWLOG>RED[O]
```

Parameters

None

Operation

Use **SHOWLOG>REDO** to correct errors in the last command issued. This command functions like the MPE's **REDO** command.

Whenever you issue a command with a syntax error, SHOWLOG automatically enters REDO mode. Bypass the Auto-Redo feature by pressing RETURN.

Examples

Edit the last command entered by typing:

```
    SHOWLOG>REDO
```

SHOWLOG>RESET

Resets selection criteria and/or report settings to default values.

Syntax

SHOWLOG>RES[ET] *item*

Parameters

item The report setting or selection item set to the default. Possible items to be reset include:

AL[L]	All report settings and selection criteria to default values.
AP[PLICATION]	Current application selection to all applications.
DA[TES]	Current date selection to all dates.
FI[LES]	MASTER, FROM, and TO files selection to all files.
LA[ST]	Last transaction setting.
PR[OJECT]	Project selection to all projects.
RO[UTE]	Route selection to all routes.
SE[LECT]	All selection criteria.
SO[RT]	Sort sequence to unsorted.
ST[EP]	Current step selection to all steps.
SU[BSET]	Resets the subset. The next report shows all transactions in the current extract file.
TI[TLE]	Report title to the default title.
US[ERS]	Current user selection to all users.

Operation

Use **SHOWLOG>RESET** to return settings and criteria to default values.

Using **RESET** with the **SUBSET** parameter has the same effect as issuing **UNDO**.

Examples

Reset the sort sequence to default (unsorted) by typing:

```
SHOWLOG>RESET SORT
```

Related Information

See **SHOWLOG>UNDO**
SHOWLOG>SELECT

SHOWLOG>SAVE

Saves the current report settings and selection criteria in a file.

Syntax

SHOWLOG>SA[VE] *xeqfile*

Parameters

xeqfile	Any valid filename within your login account.

Operation

SHOWLOG>SAVE builds a file with commands that can be used to restore the current report settings at a later time. If you specify an already existing file, you are asked if you would like to replace it. Subsets are not saved.

Use **SHOWLOG>GET** to execute commands in the SAVE file.

Examples

Save settings in a file called SETTINGS by typing:

```
SHOWLOG>SAVE SETTINGS
```

Related Information

See **SHOWLOG>GET**

SHOWLOG>SELECT

Extracts only those transaction records which match the specified selection criteria.

Syntax

```
SHOWLOG>[ SEL[ECT] ]
  { AP[PLICATION] appl                                              }
  { BRANCH                                                          }
  { [COMMAND] command                                               }
  { DA[TES] daterange [ ,daterange [ ,... ] ]                       }
  { FA[ILURES]                                                      }
  { FR[OM] filelist [ TO filelist ] [ ,filelist [ TO filelist ] [,...] ] }
  { INPROGRESS                                                      }
  { LA[ST]                                                          }
  { MA[STERS] masterfile [ ,masterfile [ ,... ] ]                   }
  { NE[W]                                                           }
  { OR[PHANS]                                                       }
  { PR[OJECT] project                                               }
  { RO[UTE] route                                                   }
  { ST[EP] step                                                     }
  { TEXT memotext                                                   }
  { TO filelist [ FROM filelist ] [ ,filelist [ FROM filelist ] [,...] ] }
  { US[ERS] user [ ,user [ ,... ] ]                                 }
```

Parameters

APPLICATION *appl* Specifies application to be matched when extracting transaction summary records. Use four asterisks (****) to indicate transactions involving more than one application. Only one application can be specified. A warning is displayed if the application is not currently defined. By changing the application, any values for route, step, and project are reset. If you don't specify an application, all applications qualify for the report.

BRANCH Selects files that were created on a branch.

COMMAND *command*

Indicates the LIBRARIAN command to be matched when extracting transaction summary records. Selects transaction records for a particular command:

COMPRESS	**RENAME**
COPY	**RESET**
DECOMPRESS	**SCAN**
LOCK	**SECURE**
MAIL	**SET EXPDATE**
MEMO	**SET MODE**
MOVE	**SET LOCKWORD**
OVERLAY	**TOUCH**
PERFORM	**UNLOCK**
PURGE	**UPDATE**

Parameters, continued

DATES *daterange*	Selects transaction records for a particular date or date range. Multiple date ranges are permitted. The keyword **TODAY** can be used for the current date. The date range parameter can be in these forms:

Single Date	Date Range
xx/xx/xx	xx/xx/xx-xx/xx/xx
TODAY	xx/xx/xx-xx/xx/xx
	–xx/xx/xx
	xx/xx/xx–

Enter dates in the format defined in the System Profile (SP) screen. Because the hyphen (–) indicates a date range, it cannot be used as a separator; instead, use a slash (/).

Each **SELECT DATE** command appends to a list of selected dates. Use **RESET DATES** to start a new list of dates.

FAILURES Selects transactions that failed to complete or that failed during file operation.

FROM *filelist* Selects transaction detail records for file operations associated with a particular **FROM** file location or locations (and optionally associated **TO** file locations using **FROM** *filelist* **TO** *filelist*). Each **SELECT FROM** command appends to a list of selected files. Use **RESET FILES** to begin a new list of file specifications.

The filelist can be in any of these forms:

- Direct reference by filename
- Direct reference by logical fileset
- Indirect reference by secondary location

See *How to Refer to Files* at the beginning of Chapter 1, "Commands" for detailed information about referencing files using filelists.

LAST Extracts only the last transaction performed. **LAST** is available for the current LIBRARIAN session. Other selection criteria are ignored. **LAST** is useful for inspecting the results of a transaction when **QUIET ON** is in effect, or when results have scrolled off the screen. In addition, **LAST** can be used to generate printed documentation of your transactions.

INPROGRESS Selects files checked out in transactions that used the **INPROGRESS** parameter.

SHOWLOG>SELECT (continued)

Parameters, continued

MASTERS	Selects transaction detail records for file operations associated with a particular master file or files. Each **MASTER** command appends to a list of valid file specifications. Use **RESET FILES** to begin a new list of file specifications. A *master file* can take one of the following forms: • Direct reference by filename • Direct reference by fileset
NEW	Extracts transaction records only for files newly introduced to LIBRARIAN during the transaction.
ORPHANS	Extracts transactions records only for files orphaned during the transaction.
PROJECT	Selects transaction records for a particular project. Because project names are unique, SHOWLOG is able to set the corresponding application. Only one project name can be specified. An error occurs if the project name is not defined or if a specified application does not match the project application. If a slash (/) is used rather than a project name, transactions not connected to a project will be selected.
ROUTE	Selects transaction records associated with a particular route. If the route is unique, the corresponding application is also set. Only one route can be specified. A warning is displayed if the route is not defined. An error occurs if the route is ambiguous or if the route is not part of the current application. When a new route is specified, the value for step is reset.
STEP	Selects records for transactions associated with a particular stepname. If the stepname is unique, the corresponding application and route are also set. Only one stepname can be specified. A warning is displayed if the stepname is not documented. An error occurs if the step is ambiguous or the step is not part of the current application/route.
TEXT	Searches memo text for the string specified and reports the corresponding transactions.
TO *filelist*	Selects transaction detail records for file operations associated with a particular **TO** file location(s), and optionally associated **FROM** file locations using **FROM** *filelist* **TO** *filelist*. Each **SELECT TO** command appends to a list of selected files. Use **RESET FILES** to begin a new list of file specifications. The *filelist* parameter can take any of the forms described earlier for the **FROM** *filelist* parameter.
USER *user*	Selects transactions performed by a particular user(s). Each **USER** command appends to the list of selected users. Use **RESET USER** to start a new list of users. Warnings are issued if you specify an undocumented user.

SHOWLOG>SELECT (continued)

Operation

SHOWLOG>SELECT establishes the selection criteria that SHOWLOG uses to extract transaction records. **SHOWLOG>SELECT** generates a report based on these criteria by using **GO**. Once records have been extracted, they remain in effect until you change any of the selection criteria.

You can omit the word **SELECT** and execute this command by specifying only the selection parameters.

If you specify a project fileset, then project activity dates are set automatically as selection criteria.

Examples

Extract only transactions associated with the FIN application by typing:

```
SHOWLOG>APPLICATION FIN
```

Extract only transactions involving multiple applications by typing:

```
SHOWLOG>APP ****
```

Extract only transactions performed on September 30, 1991 by typing:

```
SHOWLOG>DATE 9/30/91
```

Extract only transactions performed before October 01, 1991, and between November 01, 1991 and today by typing:

```
SHOWLOG>DATE -10/01/91, 11/01/91-TODAY
```

The following example extracts only transaction detail records where the FROM file location was either the JCL or PRG groups of the FINANCE account, and the TO file location in both cases was the PROD group.

MPE
```
SHOWLOG>FROM @.JCL.FINANCE TO @.PROD.FINANCE, @.PRG.FINANCE TO
@.PROD.FINANCE
```

UNIX
```
SHOWLOG>FROM /finance/jcl/* TO /finance/prod/*, /finance/prg/* TO /finance/prod/*
```

The following example extracts only transaction detail records where the FROM file was a secondary copy of a file in the FINANCE fileset.

MPE
```
SHOWLOG>FROM %FINANCE AT @.@.@.@
```

UNIX
```
SHOWLOG>FROM %FINANCE AT *:/*
```

Select the last transaction performed only by typing:

```
SHOWLOG>LAST
```

Select only transactions associated with the PAYROLL project by typing:

```
SHOWLOG>PROJECT PAYROLL
```

Select only transactions associated with the DEVEL route by typing:

```
SHOWLOG>ROUTE DEVEL
```

SHOWLOG>SELECT (continued)

Examples, continued

Set a flag to extract transaction detailed records for new files by typing:

```
SHOWLOG>SELECT NEW
```

Select only orphaned files by typing:

```
SHOWLOG>SELECT ORPHANS
```

Select only new files, where the file operation failed by typing:

```
SHOWLOG>NEW
SHOWLOG>FAILURES
```

Select only transactions associated with the CHECKOUT step by typing:

```
SHOWLOG>STEP CHECKOUT
```

Select only transactions performed by users CHRIS, PAT or JEAN by typing:

```
SHOWLOG>USERS CHRIS, PAT, JEAN
```

Alternatively, select these transactions by typing:

```
SHOWLOG>USER CHRIS
SHOWLOG>USER PAT
SHOWLOG>USER JEAN
```

Related Information

See **SHOWLOG>RESET**

SHOWLOG>SHOW

Refreshes the display of selection criteria and report settings.

Syntax

```
SHOWLOG>SH[OW]
```

Parameters

None

Operation

Use **SHOWLOG>SHOW** to refresh the display of report settings and selection criteria. This command is useful when the display has been overwritten or when you need to review lengthy lists of criteria.

Examples

Refresh the screen after someone has sent you a message by typing:

```
SHOWLOG>SHOW
```

SHOWLOG>SORT

Sets up the sort sequence for a report.

Syntax

SHOWLOG>SO[RT] [-] *sortkey* [, [-] *sortkey* [,...]]

Parameters

–	Uses descending sort order. If this parameter is omitted, ascending order is used.
sortkey	Any of the following sort keys.

AP[PLICATION]	Sorts by application.
DA[TES]	Sorts by date.
FR[OMFILE]	Sorts by FROM file.
MA[STERFILE]	Sorts by master file.
PR[OJECT]	Sorts by project.
RO[UTE]	Sorts by route.
ST[EP]	Sorts by ste.
TI[ME]	Sorts by time.
TO[FILE]	Sorts by TO file.
US[ERS]	Sorts by user.

Operation

SHOWLOG>SORT sets the sort sequence for SHOWLOG reports. Transaction summary reports (CONCISE, SUMMARY, MEMO, etc.) cannot include file sorts since files do not appear in these formats. The RTD10, RTD40, and RTS10 sort sequences cannot be changed.

Examples

Sort the report by application, route, step, date and time (descending) by typing:

```
SHOWLOG>SORT AP, RO, ST, -DA, -TI
```

Sort the report by user by typing:

```
SHOWLOG>SORT USER
```

SHOWLOG>SUBSET

Flags a subset of currently extracted transactions for reporting.

Syntax

```
SHOWLOG>SUB[SET] refnum [ -refnum ] [,...]
```

Parameters

refnum	The transaction reference number. Ranges of reference numbers can be specified by using a hyphen (–).

Operation

SHOWLOG>SUBSET marks a subset of the transaction extract file for refined reports. Obtain reference numbers by using the concise report format and press CONTROL-Y. Each **SUBSET** command adds to the current subset. Use the **UNDO** or **RESET SUBSET** command to return to the original set of extracted transactions.

Note When you change the sort sequence, the subset is preserved even though the reference numbers change. Without parameters, **SUBSET** lists the reference numbers in the current subset.

Examples

View a refined report consisting of transactions numbered 1-4, 22, 66, and 132 through 180 by typing:

```
    SHOWLOG>SUBSET 1-4, 22, 66, 132-180
```

Related Information

See **SHOWLOG>UNDO**
RESET SUBSET

SHOWLOG>TITLE

Sets a title to appear on all pages of a report.

Syntax

```
SHOWLOG>TI[TLE] title
```

Parameters

title	A title, up to 50-characters in length, centered at the top of each page of a report.

Operation

SHOWLOG>TITLE lets you define your own title for your customized SHOWLOG report.

Examples

Change the default title for a report by typing:

```
SHOWLOG>TITLE File Operations Performed by Finance Programmers
```

SHOWLOG>UNDO

Resets the current subset.

Syntax

```
SHOWLOG>UN[DO]
```

Parameters

None

Operation

SHOWLOG>UNDO resets the subset. The next report generated uses all of the transactions in the extracted file (same as the **RESET SUBSET** command).

Examples

Return to the original set of extracted transactions by typing:

```
    SHOWLOG>UNDO
```

Related Information

See **SHOWLOG>RESET SUBSET**
SHOWLOG>SUBSET

Screens 5

This chapter describes in detail all of the LIBRARIAN screens. It includes the following topics:

- Screen Summary
- User Capability Requirements for Screen Access
- Accessing LIBRARIAN Screens
- Using LIBRARIAN Screens
- Screen Descriptions

Screen Summary

Table 5–1 lists the LIBRARIAN screens by their functions. Page numbers help you locate the detailed screen descriptions in this chapter.

Table 5–1 LIBRARIAN Screens

Screen Code	Screen Title	Page
System Data		
CE	Compress Exclusions	5–13
SP	System Profile	5–62
Network Data		
NC	Network Configuration	5–37
SY	Systems	5–85
SS	System-to-System Table	5–69
Users		
UC	User Capabilities	5–89
US	Users	5–91
Files		
AF	Auto Filesets	5–9
FA	File Access	5–16
FC	Fileset Components	5–19
FF	Files in Filesets	5–21
FI	File Inquiry	5–24
FS	Filesets	5–29
PF	Pending Master Files	5–42
Applications/Routes/Steps		
AP	Applications	5–11
CP	Composite Presteps	5–14
FV	Forward Versioning	5–32
PP	Pending Production Areas	5–51
RT	Routes	5–57
SA	Step Authorizations	5–59
SR	Step Refinements/Exceptions	5–66
ST	Steps	5–71
STO*	Step Options	5–76
Project Data		
PA	Project Authorizations	5–40
PI	Project Inquiry	5–45
PJ	Projects	5–48
PS	Project Status Change	5–54
Miscellaneous		
LP**	Long Pathname	5–34

* This screen is available only through the Steps (ST) screen.

** This screen can be accessed only through the AF, FA, FF, FI, FV, PF, PP, SR, and ST screens.

User Capability Requirements for Screen Access

LIBRARIAN permits access to the screens at these five levels:

- LIBRARIAN Manager (L):
  Accesses all screens and records stored in a database.
- Application Manager (A):
  Accesses the data entry screens and records associated with their own applications.
- Project Managers (P):
  Accesses the data entry screens and records associated with their own projects.
- General Users:
  Accesses their own User (US) records, the File Inquiry (FI) screen, and the Pending File (PF) screen (if LIBRARIAN is configured to allow PF access).
- Rule administrators (R):
  Accesses the data entry screens for application related data. Users with this capability cannot create users, authorize steps, or assign special capabilities.

Required user capabilities for accessing each screen are listed in the Screen Descriptions section of this chapter, as well as in the online help.

Accessing LIBRARIAN Screens

There are two ways to access the LIBRARIAN screens:

1. At the LIBRARIAN prompt >, invoke the LIBRARIAN screens by entering the two-letter screen code.

   For example, proceed directly to the Steps screen (ST) by typing:

   ```
   >ST
   ```

2. From the Main menu bar, select **Admin** followed by **Screens**. The following options appear:

   - **Config...** Accesses the screens to review, add, change, or delete system and network data in the database.
   - **Users...** Accesses the screens that identify a user and define the user's capabilities.
   - **Files...** Accesses the screens to define filesets and file attributes.
   - **Steps...** Accesses the screens to add and tune routes and steps.
   - **Projects...** Accesses the screens to define and maintain projects.

   Select the appropriate screen from one of these menus.

Using LIBRARIAN Screens

Moving Between Screens

In the upper-left corner of each screen is the selection field. To move directly from one screen to another, enter a two-letter screen code in the selection field and press ENTER.

Moving Between Fields

Move from one field to another field on a screen by pressing TAB.

Enter Key

After entering data in the fields on a screen, press the ENTER key to add new data.

Adding Data

After entering data in the fields on a screen and pressing ENTER, the data entry program edits and verifies the data. Valid data is added to the database, and a status message is displayed. If the data entered is invalid, an error message will be displayed.

Finding Data

To find a record, enter data in the key fields and press the **F1** (FIND) function key. The key fields for each screen are listed in the screen descriptions.

There are three ways to find records with the FIND function key:

- If the key fields uniquely identify a single record, FIND retrieves that record.
- If the key fields identify multiple records, **F1** retrieves the first record that matches those fields. Subsequent FINDs locate other records that may match those fields.
- If you do not make changes after a FIND operation, pressing **F1** again retrieves the next record in a database or in a chain, depending on the last FIND operation.

To find a record when you do not know any of the key data or to review all records, use **F6** (FIRST REC) to access the first record in the dataset. Then, use **F1** to serially access subsequent records in the set.

Using LIBRARIAN Screens, continued

Carrying Data Forward

When using the screens, LIBRARIAN automatically carries forward key data values from one screen to the next. This eliminates the need to enter the values again. The data values that LIBRARIAN carries forward include:

Application	Fileset	Long Pathnames
Route	User	
Stepname	Project name	

For example, if you are reviewing the Fileset Components (FC) screen and then you proceed to the Auto Filesets (AF) screen, the fileset name is carried forward and displayed in the Fileset field.

Changing Data

To change the data in a record, first use the **F1** function key to find it. Use the **TAB** key to move to the field(s) that you want to change. After typing new values in the field(s), press **F2** (CHANGE) function key to enter the changes into the database.

Deleting Data

To delete a record, first use **F1** to find it. Then, delete the record by pressing **F3** (DELETE). Screens that let you delete multiple records prompt you to press the F3 (DELETE) function key again to confirm.

Using Online Help

The online help provides an overview of individual screens and detailed descriptions of their fields. To review the information about the screen you are currently using, press F5 (HELP). For help on a single field, type a question mark (?) in the field and then press F5. You should clear the question mark from one field before using it for another field.

Breaking to MPE/UNIX

To break from LIBRARIAN screens to MPE or UNIX temporarily, press **F7** followed by **F5** (**BREAK**). To return to the screens from MPE issue the **RESUME** command, and from UNIX type **exit**.

Exiting from LIBRARIAN Screens

You can exit from any screen by pressing the **F8** function key (EXIT).

Using Function Keys

Use the function keys to add, change, or delete data. Each screen has two sets of function keys, as listed in Table 5–2. On each set, **F7** (**NEXT FKEYS**) toggles between the two sets of function keys.

Using LIBRARIAN Screens, continued

Table 5–2 Standard LIBRARIAN Screens Function Keys

Set 1		Set 2	
Key	**Function**	**Key**	**Function**
F1	FIND	F1	PRINT
F2	CHANGE	F2	(NOT USED)
F3	DELETE	F3	(NOT USED)
F4	REFRESH or LONG PATHNAME	F4	REFRESH
F5	HELP	F5	BREAK
F6	FIRST REC	F6	(NOT USED)
F7	NEXT FKEYS	F7	NEXT FKEYS
F8	EXIT	F8	EXIT

For some screens, these standard functions are not appropriate. In these cases, the standard function keys are not available or are replaced with special functions. Special functions are described in the Operation section for each detailed screen description.

Function Keys: Set 1

The following list describes in detail the first set of function keys:

F1 (FIND)	Retrieves records associated with the key fields on a screen.
F2 (CHANGE)	Changes a record. First retrieve the data with **F1** (FIND) and then change it with **F2.**
F3 (DELETE)	Deletes a record. First retrieve the data with **F1** (FIND) and then press **F3** to delete the record.
F4 (REFRESH)	Restores data as it appeared when you first retrieved it. Use this key if you want to restore the original display before changing the database.
or	
F4 (LONG PATHNAME)	Accesses the Long Pathname screen, where you can enter or view pathnames longer than 64 characters.
F5 (HELP)	Accesses online help for the screen you are currently using.
F6 (FIRST REC)	Accesses the first record in a database for the screen.
F7 (NEXT FKEYS)	Changes to the second set of function keys.
F8 (EXIT)	Exits screen.

Function Keys: Set 2

The following list describes in detail the second set of function keys:

F1 (PRINT)	Prints the contents of the current screen to the line printer.
F2	Not used.
F3	Not used.
F4 (REFRESH)	Restores data as it appeared when you first retrieved it. Use this key if you want to restore the original display before changing the database.
F5 (BREAK)	Breaks to MPE or UNIX. To return from MPE, temporarily issue the **RESUME** command, and from UNIX type **exit**.
F6	Not used.
F7 (NEXT)	Changes to the first set of function keys.
F8 (EXIT)	Exits screen.

Note

If you have several similar records to enter, use the following sequence to minimize data entry effort:

1. Type the first record and then hit the ENTER key to add to the database. The data remains displayed on the screen.
2. Next, type the data in the fields which are different and hit the ENTER key to add the subsequent records to the database.

Screens Descriptions

The remainder of this chapter provides detailed descriptions of each LIBRARIAN screen, presented in alphabetical order by screen code. Each screen description includes the following topics:

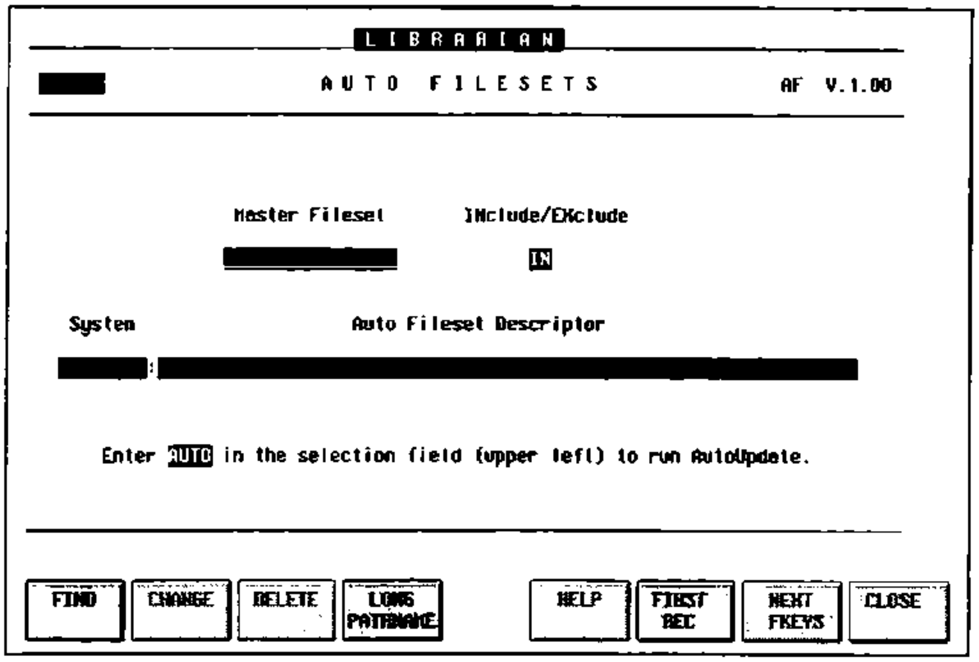
Screen Name and Code	Formal screen name and the associated screen code are shown in bold at the beginning of each screen description.
Menu Access	The menu from which you can access the screen.
Screen Security	User level of capability required.
Files Impacted	Datasets affected by add, change, and delete operations.
Screen layout	Illustration of the screen.
Key Fields	List of key fields.
Description	Basic function of the screen.
Explanation of Fields	Detailed description of each field on the screen.
Operation	Special operation information about the screen. This topic is included only for screens with special operations.

AUTO FILESETS AF

The Auto Filesets (AF) screen is where you define and maintain descriptors to automatically keep master filesets up–to–date.

Menu Access	**Admin...Screens...Files...AF Auto Filesets**
Screen Security	LIBRARIAN Manager, Rule Administrator
Files Impacted	D-AUTO-FILESET

Screen Layout



```
                        LIBRARIAN
          AUTO FILESETS                      AF V.1.00

            Master Fileset      INclude/EXclude
                                      IN

  System              Auto Fileset Descriptor
         :

      Enter AUTO in the selection field (upper left) to run AutoUpdate.
```

Key Fields

Master Fileset

Description

Before you can load Auto Fileset descriptors, you must define the fileset in the Filesets (FS) screen. Each descriptor uses specific values or wildcards to describe a general file location that is either included or excluded from the fileset.

Whenever the Auto Fileset Update (AUTOUPDP) program runs (by typing AUTO in the selection field of any screen or through the **AUTOUPDATE** command or Admin menu option), it explodes these descriptors into a list of valid files. The program adds previously undefined files to the fileset records in the database.

Auto Fileset descriptors are also used to update fileset definitions automatically when the **AUTOUPDATE** parameter is specified on secondary-to-master steps.

Explanation of Fields, continued

Master Fileset
Required. Length 16.

The name of the fileset. The fileset must be previously defined by the Filesets (FS) screen, or automatically by the Applications (AP) screen.

INclude/EXclude
Required. Length 2.

A flag indicating if the files identified by the descriptor should be included or excluded from the fileset. Use INclude to identify valid file locations for the fileset. Use EXclude to identify locations that should be excluded. The default is IN for include.

System
Optional. Length 8.

The LIBRARIAN system ID where the file(s) reside. If you do not specify a system, LIBRARIAN defaults to the current system.

Auto Fileset Descriptor
Required. Length 64 (255 for long pathname).

The general location of files associated with this fileset. Wildcards that are consistent with MPE/iX or UNIX conventions can be used.

To enter an Auto Fileset descriptor that is longer than 64 characters, press F4 (LONG PATHNAME). The Long Pathname (LP) screen appears to let you enter a string of characters up to 255. For more information, refer to the LP screen.

APPLICATIONS AP

The Applications (AP) screen is where you define and maintain application records.

Menu Access

Admin...Screens...Steps...AP Applications

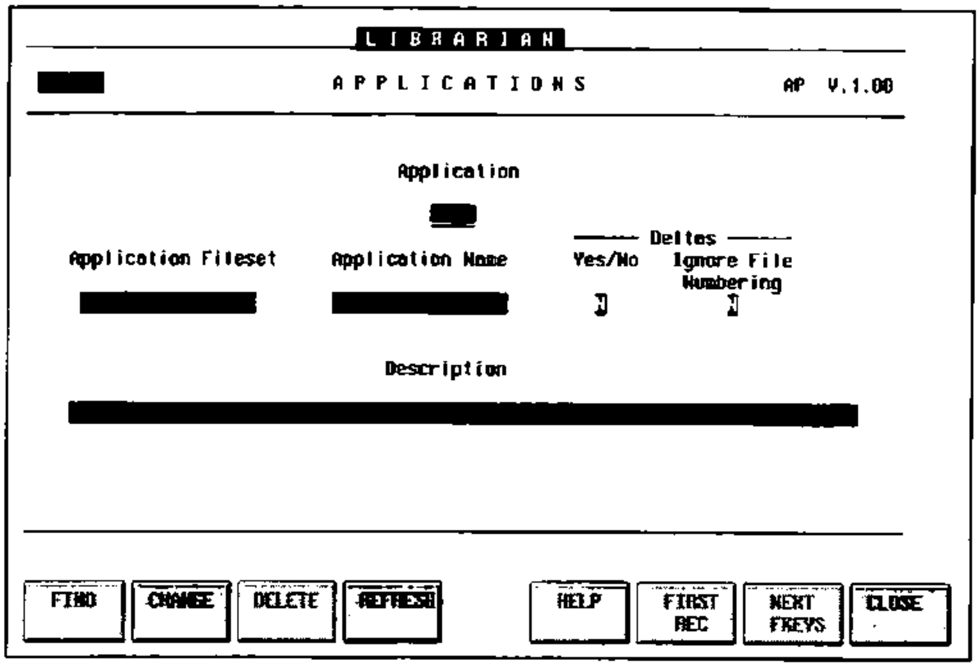
Screen Security

LIBRARIAN Manager, Rule Administrator, and Application Manager. Application Managers can only access the records for their applications.

Files Impacted

M-APPLICATION, M-FILE-SET

Screen Layout



Key Fields

Application

Description

An application is the highest organizational unit in LIBRARIAN. Applications consist of files and standard file movement rules (defined as routes and steps).

The AP record uniquely identifies an application. The highest level fileset of the application is defined on this screen. If the fileset has not been previously defined, it is automatically added.

Explanation of Fields

Application
Required. Length 4.

Explanation of Fields, continued

A unique identifier for the application. The application identifier can include alphabetic, numeric, hyphen (-), and underscore (_) characters.

Application Fileset
Required. Length 16.

Logical name of the fileset that includes all filesets and files for the application. If you have not previously defined the fileset on the Filesets (FS) screen, it will automatically be defined.

Application Name
Required. Length 16.

A longer name for the application for documentation only.

iX+ **Deltas (Yes/No)**
Required. Length 1.

Specifies whether LIBRARIAN creates delta files or generation files when retaining old versions of fixed text files. Legal values are:

- Y Instructs LIBRARIAN to store revisions as deltas. During checkin, only the changes between revisions are stored. For binary files, all prior revisions are kept as generation files.
- N Instructs LIBRARIAN to store previous generations in their entirety (can be compressed if the flag is set on the System Profile (SP) screen).

Note The ANNOTATE option of the **COPY, PERFORM,** and **PRINT** commands and **MERGE** feature can only be used if deltas are being stored. This shows the change history of a text file as comments within the text.

MPE **Deltas (Ignore File Numbering).**
Required. Length 1.

Specifies whether file numbering is significant when determining deltas. This field is applicable only when you specified Y in the Deltas Yes/No field. Legal values are:

- Y Instructs LIBRARIAN to ignore file numbering for source files.
- N Instructs LIBRARIAN not to ignore file numbering for source files.

Description
Optional. Length 72.

A description of the application.

Operation

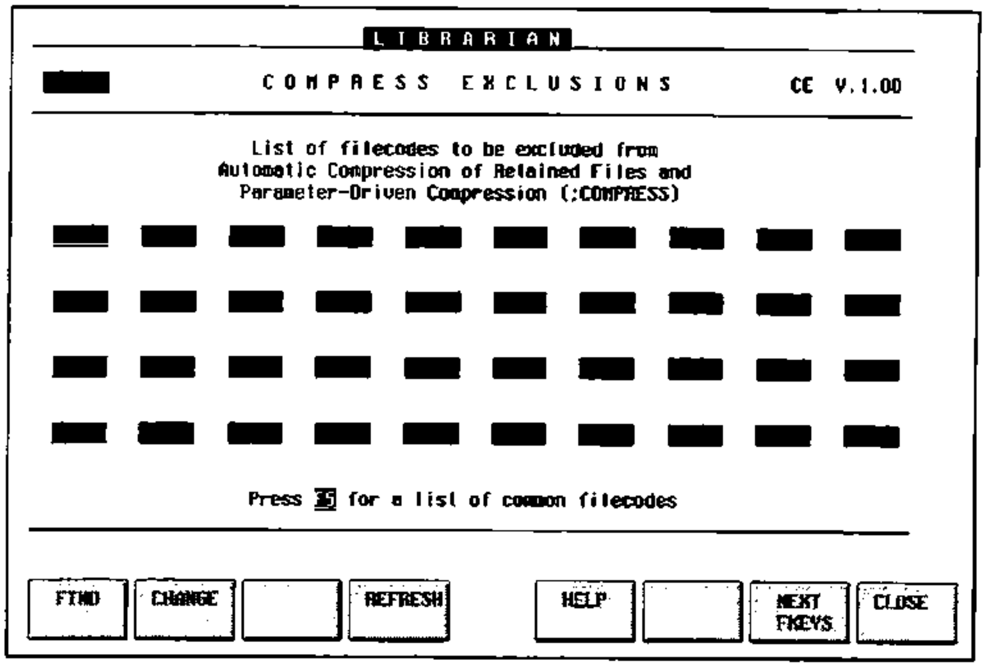
When you use the DELETE function on this screen, the entire application, including fileset definitions and movement rules, are deleted. You will be prompted to press DELETE again to confirm mass deletion.

COMPRESS EXCLUSIONS CE

The Compress Exclusions (CE) screen is where you identify the types of files to be excluded from automatic compression.

Menu Access	Admin...Screens...Config...CE Compress Exclusions
Screen Security	LIBRARIAN Manager
Files Impacted	D-NO-COMPRESS

Screen Layout



```
                         LIBRARIAN
              COMPRESS  EXCLUSIONS              CE  V.1.00

            List of filecodes to be excluded from
          Automatic Compression of Retained Files and
           Parameter-Driven Compression (;COMPRESS)

           Press [illegible] for a list of common filecodes
```

Key Fields

None

Description

File compression takes place when issuing LIBRARIAN commands that use the **COMPRESS** parameter.

List the MPE filecodes and/or mnemonics for the types of files that should not automatically be compressed.

Note that the entries on this screen do not prevent files from being compressed by the **COMPRESS** command.

Operation

The DELETE and FIRST RECORD functions are not available on this screen.

COMPOSITE PRESTEPS CP

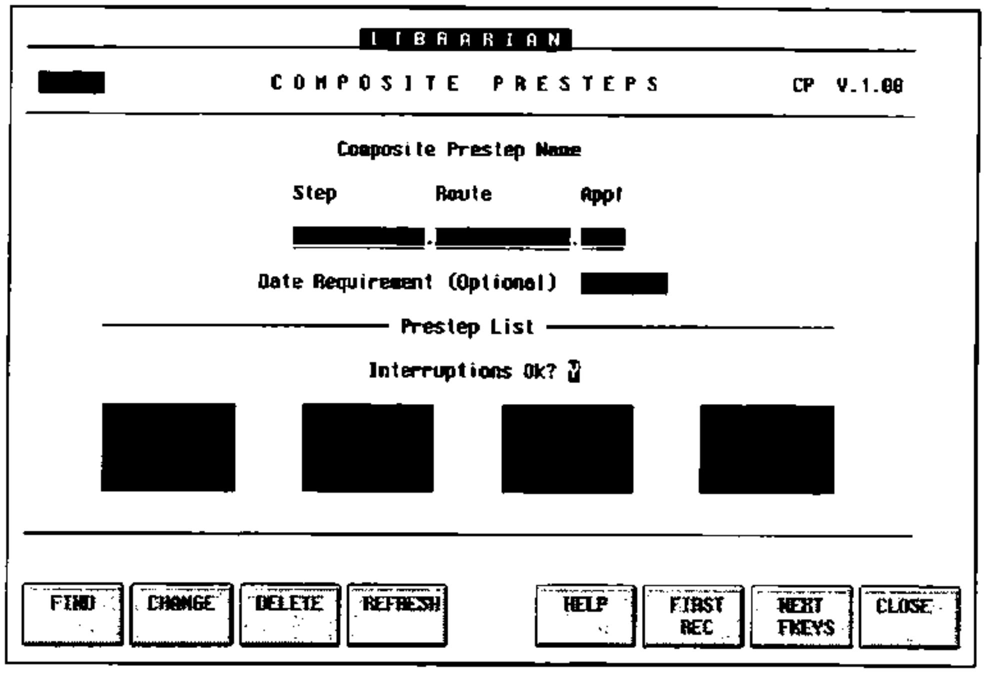
The Composite Presteps (CP) screen is where you assign a composite prestep name to refer to a collection of steps.

Menu Access Admin...Screens...Steps...CP Composite Presteps

Screen Security Application Managers, Rule Administrator

Files Impacted D-PRESTEPS

Screen Layout



```
LIBRARIAN
COMPOSITE PRESTEPS                CP V.1.00
Composite Prestep Name
Step        Route        Appl
Date Requirement (Optional)
Prestep List
Interruptions Ok?
```

Key Fields

Composite Prestep Name

Description

You can also use the optional Date Requirement field to create a prerequisite for a step that prevents the step from being performed prior to a particular date.

A composite prestep name that is used as a prestep or an alternate prestep on the Steps (ST) screen. The order in which the presteps that make up the composite are performed is not important. The steps listed must be previously defined in the Steps (ST) screen.

Each composite prestep must include either a date requirement or a minimum of one step.

Explanation of Fields

Composite Prestep Name
The name for this group of steps consisting of the following subfields:

> **Step**
> Required. Length 12.
>
> Name for this composite prestep. The step name can include alphabetic, numeric, hyphen (-), and underscore (_) characters.
>
> **Route**
> Required. Length 12.
>
> Identifier of the route to which the step belongs. You must have previously defined the route on the Routes (RT) screen.
>
> **Appl**
> Required. Length 4.
>
> The application to which the route and step belong. You must have previously defined the application on the Applications (AP) screen.

Date Requirement
Optional. Length 8.

A date in this field indicates that the composite prestep is not satisfied until this date, even if all of the listed steps have been performed.

Interruptions Ok?
Optional. Length 1.

A flag that indicates whether intervening steps are permitted when steps in the composite prestep are performed. Intervening steps are steps that are not part of the composite prestep list. Legal values are:

Y	Interruptions are acceptable.
N	Interruptions are not acceptable.

For example, if three approvals comprise a composite prestep, and a file is rejected prior to the third approval, all approvals would be required again if no interruptions are allowed.

Prestep List

A list containing a maximum of 16 steps, each of which must be completed in any order prior to the step for which the composite is a prerequisite. The steps must be previously defined on the Steps (ST) screen and be part of the same route.

FILE ACCESS FA

The File Access (FA) screen is where you review and change the access control, default access mode, and/or the language for a master file.

Menu Access

Admin...Screens...Files...FA File Access

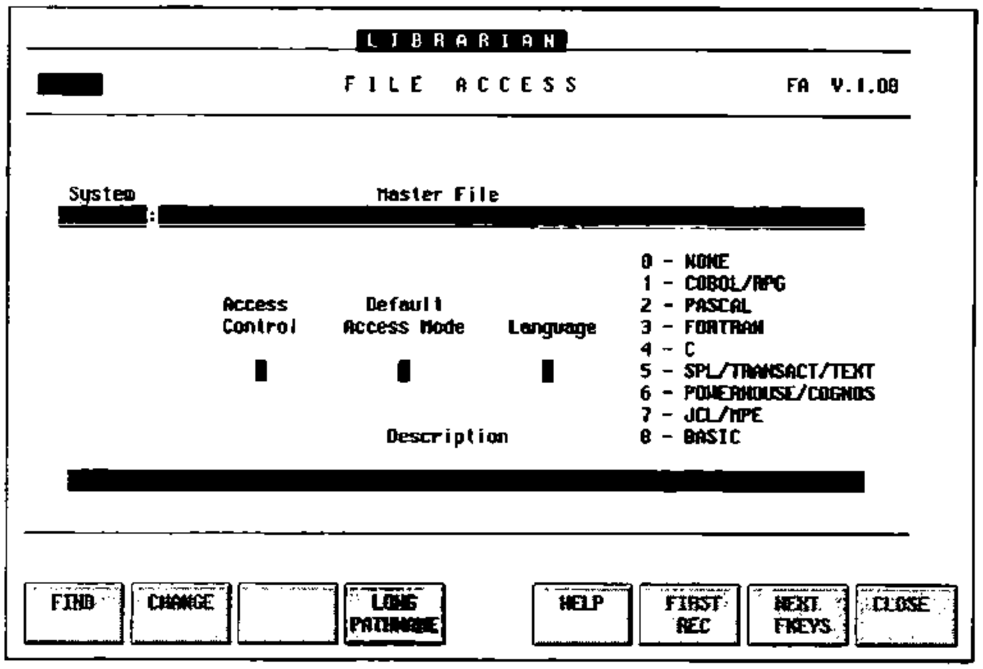
Screen Security

LIBRARIAN Manager, Rule Administrator, and Application Manager. Application Managers can access records only for files in their applications.

Files Impacted

D-FILE

Screen Layout



```
                          L I B R A R I A N
            F I L E   A C C E S S                    FA V.1.08

 System                    Master File
        :
                                              0 - NONE
                                              1 - COBOL/RPG
        Access      Default                   2 - PASCAL
        Control     Access Mode   Language    3 - FORTRAN
                                              4 - C
                                              5 - SPL/TRANSACT/TEXT
                                              6 - POWERHOUSE/COGNOS
                                              7 - JCL/MPE
                    Description               8 - BASIC
```

Key Fields

Master File

Description

The master files that you retrieve on this screen were originally loaded into the database from the Files in Filesets (FF) screen, a step that introduces new files, or the Auto Fileset Update (AUTOUPDP) program.

Explanation of Fields

System
Optional. Length 8.

The system where the file is located. If you do not specify a system, LIBRARIAN uses the current system as the default.

Master File
Required. Length 64 (255 on Long Pathname screen).

The name of a file that is part of an application library.

To enter a filename that is longer than 64 characters, press F4 (**LONG PATHNAME**). The Long Pathname (LP) screen appears, letting you enter a maximum of 255 characters. For more information, refer to the LP screen.

Access Control
Required. Length 1.

The access control level that determines the number of read and write mode copies allowed at one time. Legal values are:

X Exclusive	The master file cannot be copied at all (default access must also be read mode).
R Read	Any number of read mode copies allowed. However, no secondary files can have write mode access (default access must also be read mode).
S Serial-Write	Any number of read copies, but only one write-mode copy, allowed at a time.
M Multi-Write	Any number of copies in any mode allowed at one time (not recommended).

Default Access Mode
Required. Length 1.

The default access mode assigned to files if not specified when performing a step. Legal values are:

R Read	Secondary copies default to read mode access.
W Write	Secondary copies default to write mode access.

Language
Required. Length 1.

A flag indicating the programming language for a source code file. The **ANNOTATE** option available on the **COPY**, **PERFORM**, and **PRINT** commands reconstruct the source code with embedded comments. The language attribute determines how these comments appear, consistent with the syntax for that language.

Explanation of Fields, continued

The language setting is also used for creating comments regarding merge conflicts when you use the **MERGE** option on a checkout step. This annotation is also embedded in the source code. The following languages are supported:

Language	Comment Syntax
COBOL/RPG	* comment (* in column 7)
PASCAL	{ comment }
FORTRAN	* comment (* in column 1)
C	/* comment */
SPL/TRANSACT/TEXT	<< comment >>
POWERHOUSE/COGNOS	; comment
JCL/MPE	COMMENT comment
BASIC	REM comment

Description
Optional. Length 72.

A description of the file for documentation purposes.

Operation

The ADD and DELETE function keys are not available on this screen. If you want to add or delete records, use the Files in the Filesets (FF) screen, or the LIBRARIAN **PURGE** command.

FILESET COMPONENTS

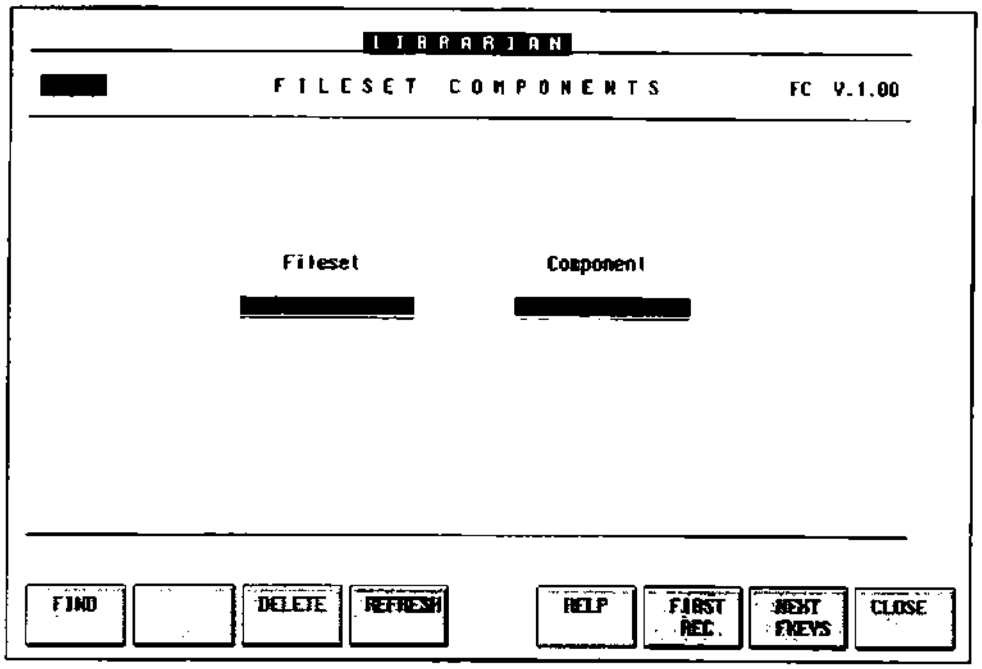
The Fileset Components (FC) screen is where you define a fileset hierarchy for an application.

Menu Access
Admin...Screens...Files...FC Fileset Components

Screen Security
LIBRARIAN Manager, Rule Administrator

Files Impacted
D-FSET-COMPONENT

Screen Layout



Key Fields

Fileset, Component

Description

A fileset component is a subset of a higher–level fileset. Create a separate record for each component. A fileset component can, in turn, include other components (other FC records). The fileset names that you use in this screen must have been previously defined on the Filesets (FS) screen.

Explanation of Fields

Fileset
Required. Length 16.

A fileset name. This fileset name must have been previously defined on the Filesets (FS) screen.

FILESET COMPONENTS (continued)

Explanation of Fields, continued

Component
Required. Length 16.

A subset of the above fileset. The fileset name must have been previously defined on the Filesets (FS) screen.

Operation

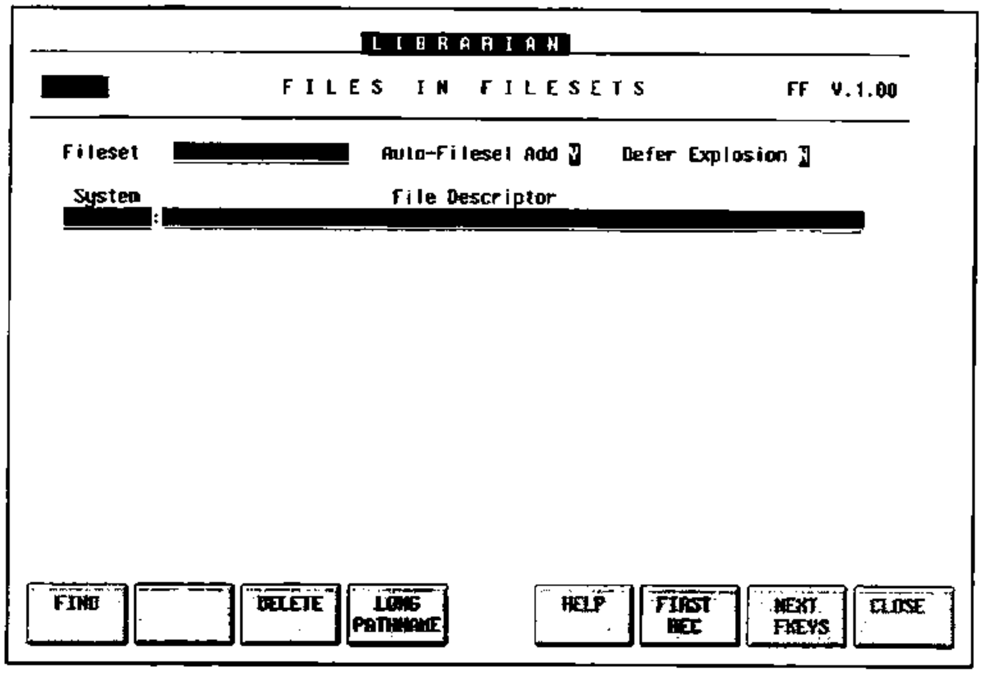
To change a record, you must first delete the record and then add a replacement record.

FILES IN FILESETS **FF**

The Files in Filesets (FF) screen is where you specify files that make up a master fileset.

Menu Access	**Admin...Screens...Files...FF Files In Filesets**
Screen Security	LIBRARIAN Manager, Rule Administrator
Files Impacted	D-FSET-FILE D-FILE D-AUTO-FILESET

Screen Layout



Key Fields

Fileset, File Descriptor

Description

The physical location is described by a descriptor in the format [*system:*] *file.group.account* (MPE) or [*system:*] /[*path.../*] *file* (UNIX). Use wildcards to identify a set of files. Enter separate records to identify other sets of files.

The descriptor is exploded to create database records for all files that match the descriptor.

You can save the file descriptor automatically as an Auto Fileset descriptor. This enables the fileset to be automatically updated by the Auto Fileset Update (AUTOUPDP) program when new files are introduced. For more information, refer to the Auto Filesets (AF) screen.

A file can belong to more than one fileset.

FILES IN FILESETS (continued)

Explanation of Fields

Fileset
Required. Length 16.

The name for a set of files, as defined on the Filesets (FS) screen.

Auto-Fileset Add
Required. Length 1.

A flag to indicate if the fileset descriptor will be added as an Auto Fileset (AF) record. Legal values are:

Y	Adds the descriptor as an AF record.
N	Does no add the descriptor as an AF record (default).

Defer Explosion
Required. Length 1.

This flag indicates when to perform the explosion for this fileset. Legal values are:

Y	Defers explosion until the Auto Fileset Update program runs (Auto Fileset flag must also be Y).
N	Explodes the fileset immediately (default).

System
Optional. Length 8.

The system where the file(s) are located. If you do not specify a system, LIBRARIAN uses the current system as the default.

File Descriptor
Required. Length 64 (255 on Long Pathname screen).

The general location of files to be added to the fileset. To enter a descriptor that is longer than 64 characters, press **F4** (LONG PATHNAME). The Long Pathname (LP) screen appears where you can enter a file descriptor of up to a maximum of 255 characters. For more information, refer to the LP screen. Wildcards consistent with MPE/iX and UNIX conventions are accepted.

Operation

F2 Function Key
(CHANGE) is not available on this screen.

ENTER Key
When you add a new FF record with ENTER, the file descriptor is exploded and database records are created automatically. If SHOW FILES is enabled on this screen, LIBRARIAN displays the files as they are being added. (See SHOW FILES, below.) As each file in the exploded set of files is processed, LIBRARIAN displays the filename and the status of each file. LIBRARIAN displays up to ten files at a time with prompts to continue or stop the transaction. After all files are processed, you are prompted prior to clearing the screen in preparation for the next transaction.

Operation, continued

If you set the Auto Fileset Add flag to Y, the descriptor is automatically added as a Auto Fileset descriptor record. In addition, if you set the Defer Explosion flag to Y, the fileset on this screen will not be exploded until you run the Auto Fileset Update (AUTOUPDP) program.

F3 (DELETE)
The delete operation removes files from the fileset.

To delete a group of records, enter wildcards in the Filename field. All records matching the specified pattern are deleted. Before LIBRARIAN deletes the records, you must press F3 a second time to confirm your deletion request.

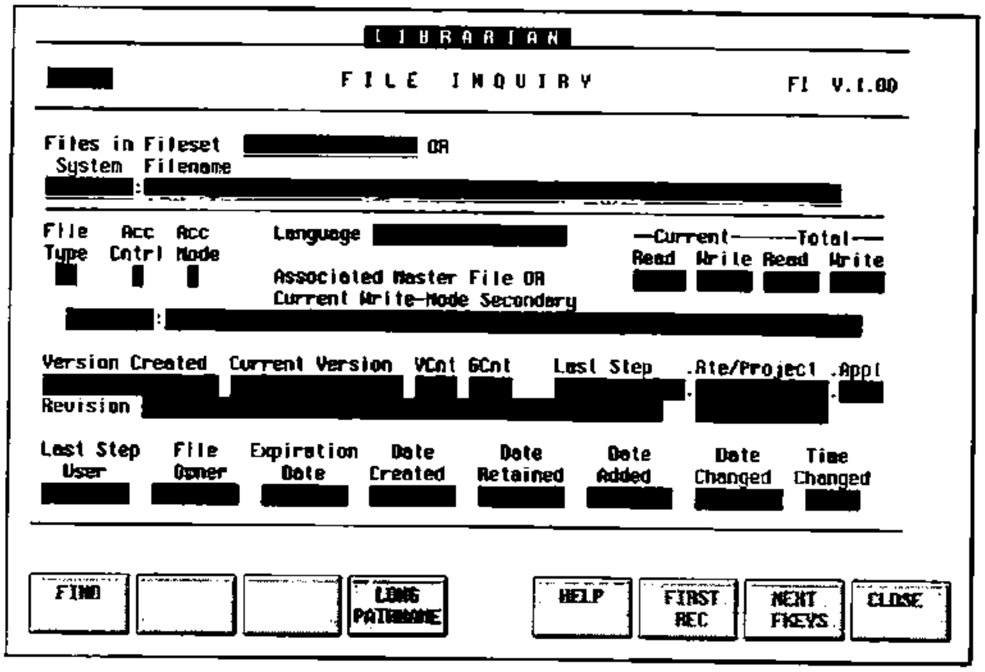
F2 on Set 2 (SHOW FILES)
When an entry is made on this screen and the fileset is exploded, LIBRARIAN normally displays the filenames and the status of each file (added, changed, etc.). Use **F2** (on **Set** 2) to suppress or re-enable this display function.

FILE INQUIRY

From the File Inquiry (FI) screen, you can obtain information about a specific file or files in a fileset.

Menu Access	Info...Files...FI File Inquiry
Screen Security	Any user
Files Impacted	None (online report)

Screen Layout



Key Fields

Fileset, File

Description

The information is retrieved from the database and cannot be changed from this screen.

Explanation of Fields

Files in Fileset
Required. Length 16

The fileset name, as defined on the Filesets (FS) screen.

Explanation of Fields, continued

System
Required. Length 8.

The system where the file is located. If you do not specify a system, LIBRARIAN uses the current system as the default.

Filename
Required. Length 64 (255 on Long Pathname screen).

The name of a master library file.

To enter a filename that is longer than 64 characters, press **F4** (LONG PATHNAME). The Long Pathname (LP) screen appears to let you enter a name up to 255 characters long. Pressing F4 displays the Filename field on the LP screen.

File Type
Display. Length 2.

Legal values are:

M	Master
S	Secondary
GM	Retained master
GS	Secondary
CM	Copy of master
CS	Copy of secondary

Acc Cntrl
Display. Length 1.

The default copy and update capability for all members of this fileset. Legal values are:

X Exclusive	The master file cannot be copied at all. (Default access must also be read mode).
R Read	Any number of read mode copies allowed. However, no secondary files can have write mode access. (Default access must also be read mode).
S Serial-Write	Any number of read copies, but only one write–mode copy, allowed at a time.
M Multi-Write	Any number of copies in any mode allowed at one time (not recommended).

Explanation of Fields, continued

Acc Mode
Display. Length 1.

Acc Mode
Display. Length 1.

The access mode for this file (or default, if a master file). Legal values are:

R Read	The secondary file has read mode access and cannot replace the master file.
W Write	The secondary file has write mode access and can replace the master file.

Associated Master File Or Current write mode Secondary
Display. Length 64 (255 on Long Pathname screen).

For a secondary file, the value in this field is the location of its master file. And, for a master file, the value in this field is the current write mode secondary file. This value is truncated if it is greater than 64 characters.

To view the Associated Master File or Current write mode Secondary field if it is larger than 64 characters, press F4 (Long Pathname), then press F1 (FIELD TOGGLE) in the LP screen. For more information, refer to the LP screen.

Current and total read/write
Display. Length 4 of 4.

The current number and total number of secondary copies with read and write accesses.

Version Created
Display. Length 16.

The version to which this file originally belonged.

Current Version
Display. Length 16.

The latest version of which this file was a part.

Language
Display. Length 22.

The programming language set for the file used for annotating source code.

VCnt
Display. Length 4.

The number of times the master file was replaced after creating the base version for the file.

GCnt
Display. Length 4.

The total number of generations of the master file that have existed over time.

Last Step
Display. Length 12.

Explanation of Fields, continued

The name of the last step performed on this file. If the file is a master, this value is the last step for the write mode secondary of the file.

Route/Project
Display. Length 12.

The route and project associated with the last step performed on this file.

Appl
Display. Length 4.

The application associated with the last step performed on this file.

Revision
Display. Length 48.

The revision ID for this file.

Last Step User
Display. Length 8.

The user who performed the last step for this file. If the file is a master, this is the user who performed the last step on a write mode secondary of the file.

File Owner
Display. Length 8.

The user who created the file.

Expiration Date
Display. Length 8.

The expiration date for this file.

Date Created
Display. Length 8.

The date LIBRARIAN created the file.

Date Retained
Display. Length 8.

The date when LIBRARIAN retained the file.

Date Added
Display. Length 8.

The date when the file record was first added to the database.

Date Changed
Display. Length 8.

The date when the file record was last modified.

Time Changed
Display. Length 5.

The time the file record was last modified.

Operation

If you want to retrieve the records for each file in a fileset, type a fileset name and press F1. Each time you press F1, the next file in the fileset is retrieved.

If you want to find the record for a specific file, first omit the fileset and then supply a filename and press F1.

FILESETS

The Filesets (FS) screen is where you define a fileset and the default characteristics for its files.

Menu Access

Admin...Screens...Files...FS Filesets

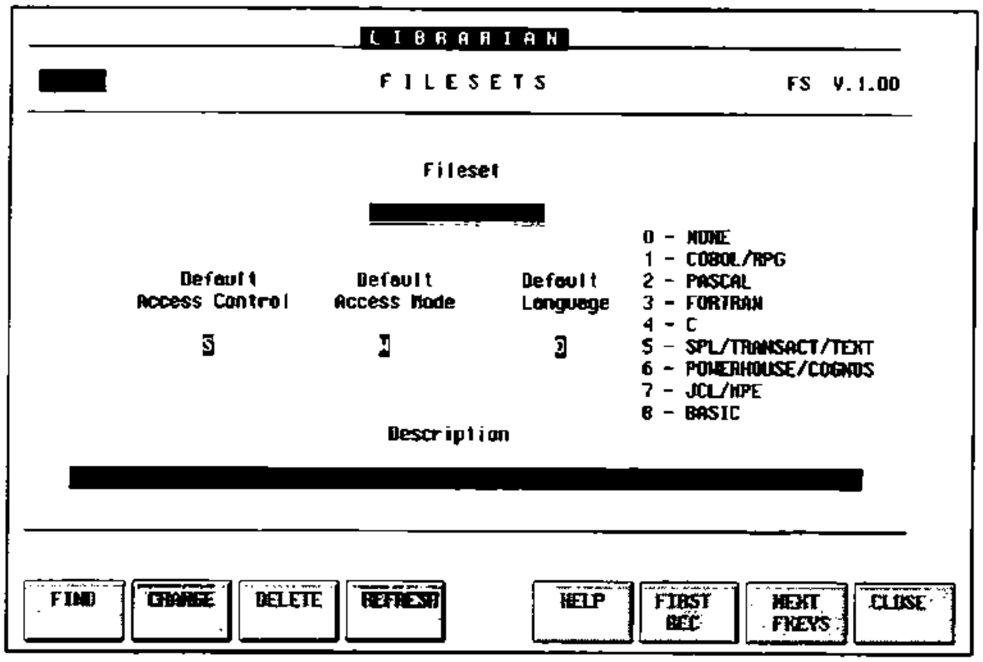
Screen Security

LIBRARIAN Manager, Rule Administrator

Files Impacted

M-FILE-SET

Screen Layout



```
                              L I B R A R I A N

                              F I L E S E T S                      FS  V.1.00

                                  Fileset

                                                      0 - NONE
                                                      1 - COBOL/RPG
       Default            Default         Default     2 - PASCAL
    Access Control      Access Mode       Language    3 - FORTRAN
                                                      4 - C
          5                  1               3        5 - SPL/TRANSACT/TEXT
                                                      6 - POWERHOUSE/COGNOS
                                                      7 - JCL/MPE
                                                      8 - BASIC
                                Description
```

Key Field

Fileset

Description

You can define one or more fileset components with the Fileset Components (FC) screen. Additionally, you can specify physical file members with the Files in Filesets (FF) and Auto Filesets (AF) screens.

The default values you define on this screen are assigned to all master files that are added as direct members of this fileset. You can use the File Access (FA) screen to specify different access attributes on a file-by-file basis, once files have been loaded with default values.

The default access mode determines if a read or a write mode is assigned to a copy of a master file.

The access control code determines how many read and write mode copies of a master file are allowed at one time.

Explanation of Fields

Fileset
Required. Length 16.

The name of the fileset. The fileset name can include alphabetic, numeric, hyphen (-), and underscore (_) characters.

Default Access Control
Required. Length 1.

The access control level are to all files in this fileset by default. Legal values are:

X Exclusive	The master file cannot be copied at all. (Default access must also be read mode).
R Read	Any number of read mode copies allowed. However, no secondary files can have write mode access. (Default access must also be read mode).
S Serial-Write	Any number of read copies, but only one write–mode copy, allowed at a time.
M Multi-Write	Any number of copies in any mode allowed at one time (not recommended).

Default Access Mode
Required. Length 1

The default access mode to be assigned to all files in this fileset. Legal values are:

R Read	The default for secondary files is read mode access.
W Write	The default for secondary files is write mode access.

Default Language
Required. Length 1.

The language setting is also used for creating comments regarding merge conflicts when you use the **MERGE** option on a checkout step. This annotation is also embedded in the source code. The following languages are supported:

Language	Comment Syntax
COBOL/RPG	* comment (* in column 7)
PASCAL	{ comment }
FORTRAN	* comment (* in column 1)
C	/* comment */
SPL/TRANSACT/TEXT	<< comment >>
POWERHOUSE/COGNOS	; comment
JCL/MPE	COMMENT comment
BASIC	REM comment

Description.
Optional. Length 72.

A description of the fileset.

Operation

Because F3 (DELETE) automatically deletes all component relationships and file members, you are prompted to confirm the deletion. Component filesets are deleted unless they are also components of another fileset. Physical file members of the main and component filesets will be deleted from the database if they are not members of other filesets. Files are not physically purged from the system.

FORWARD VERSIONING

FV

The Forward Versioning (FV) screen is where you identify alternate locations to search for a file if it is not found in the master location.

Menu Access

Admin...Screens...Steps...FV Forward Versioning

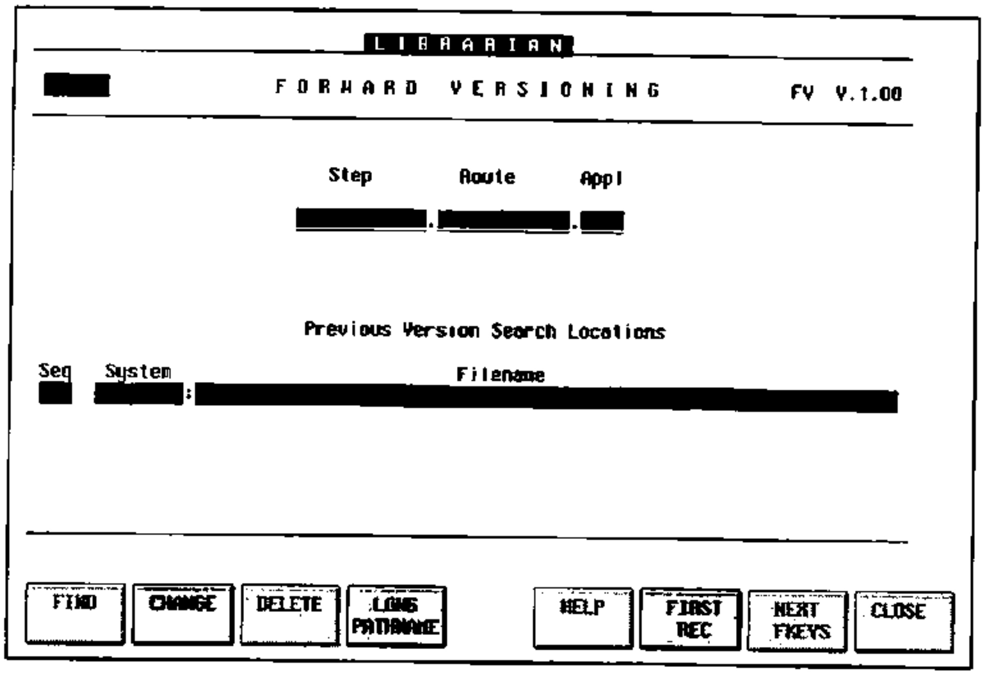
Screen Security

LIBRARIAN Manager, Rule Administrator, and Application Managers. Application Managers can only access the records for steps in their applications.

Files Impacted

D-FORWARD-VER

Screen Layout



```
                              LIBRARIAN
          FORWARD VERSIONING                    FV V.1.00

                     Step      Route     Appl
                   ________.________.___

               Previous Version Search Locations
Seq  System                  Filename
___  ________:_________________________________________________
```

Key Fields

Step, Route, Application

Description

By defining forward versioning rules, you can bring files forward from a previous version of an application to a new version in development when needed. Forward versioning applies when checking out files that do not exist yet in the location defined for the step. The alternate search locations you define on the Forward Versioning (FV) screen are typically locations for previous versions of the application. Alternate locations are checked in the sequence defined on this screen.

When a file is not found in the current (or new) version location defined for the checkout step, but is found in one of the locations defined on the Forward Versioning (FV) screen, it is introduced to LIBRARIAN as a pending master for the current (new) version in the application library. For more information on Forward Versioning , refer to Chapter 7, "Versions" in the *LIBRARIAN/iX Administrator's Guide.*

Explanation of Fields

Step
Required. Length 12.

The name of the master-to-secondary step to which the forward versioning criteria apply. Step must have already been defined on the Steps (ST) screen.

Route
Required. Length 12.

The name of the route to which the step belongs.

Appl
Required. Length 4.

The name of the application to which the step belongs.

Seq
Required. Length 3.

The sequence for searching for files.

System
Required. Length 8.

The system to search. The default is your current login.

Filename
Required. Length 64 (255 on Long Pathname screen).

The general location (using wildcards) to search for a file that is not found in the primary search location for the step.

To enter an UNIX filename that is longer than 64 characters, press **F4** (LONG PATHNAME). The Long Pathname (LP) screen appears to let you enter a maximum of 255 characters.

MPE Descriptors consist of *file.group.account* under UNIX. You can use the following wildcards:

- = Substitutes the contents of the element from the original search.
- @ Indicates zero or more alphabetic and/or numeric characters. Used alone and denotes all members of the set.
- # Indicates a single numeric character.
- ? Indicates a single alphabetic or numeric character.

UNIX Descriptors consist of */[path.../] file* under UNIX. You can use the following wildcards:

- = Substitutes the contents of the element from the original search.
- * Indicates zero or more of any characters. Used alone, denotes all members of the set.
- ? Indicates a single character.

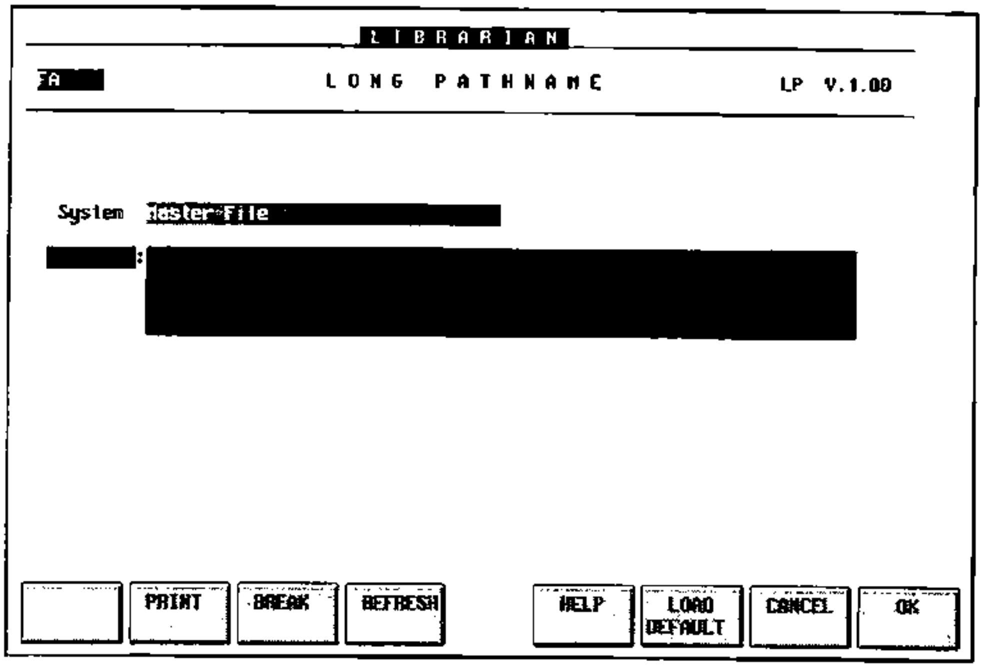
Warning If used, the equal sign (=) wildcard must be the only character for an element. You cannot use it in combination with any other wildcards and/or alphanumeric characters. All other wildcards can be used together.

LONG PATHNAME

The Long Pathname (LP) screen is where you can enter or view a pathname that is longer than 64 characters.

Menu Access	Not available directly from menu mode.
Screen Security	Any user. This screen can only be accessed from screens that have filename fields.
Files Impacted	None

Screen Layout



```
                         L I B R A R I A N
FA                      L O N G   P A T H N A M E          LP  V.1.00

  System  Master-File
```

Key Fields

None

Description

To access this screen, press **F4** (LONG PATHNAME) on applicable screens. Applicable screens include:

- Auto Filesets (AF)
- File Access (FA)
- Files in Filesets (FF)
- File Inquiry (FI)
- Forward Versioning (FV)
- Pending Master Files (PF)
- Pending Production Areas (PP)
- Step Refinements/Exceptions (SR)
- Steps (ST)

The screen code of the calling screen appears in the upper left box of the LP screen.

Note Usually, the box in the upper left corner of a screen is blank. This lets you access other screens by entering another screen code and pressing ENTER. In this screen, however, the screen code that appears is the screen from which you came, and is display only. You will always return to that screen when you are done.

Explanation of Fields

System
Optional. Length 8.

The name of the system from the field on the preceding screen.

Filename
Optional. Length 255.

The name of the file which has been expanded to 255 characters.

Operation

The following function keys have special meanings on the LP screen.

F1 (FIELD TOGGLE)

Use **F1** to toggle between fields if the calling screen has more than one filename field.

For example on the Steps (ST) screen, you can specify a source location and a destination location. Once in the ST screen, press F4 (LONG PATHNAME). Next, the Long Pathname (LP) screen appears. This lets you specify a source filename that is longer than 64 characters. After specifying a long pathname for the source filename, **F1** (FIELD TOGGLE) accesses the destination file field. Applicable screens include:

Operation, continued

- File Inquiry (FI)
- Pending Master Files (PF)
- Pending Production Areas (PP)
- Step Refinements/Exceptions (SR)
- Steps (ST)

F6 (LOAD DEFAULT)

Use **F6** to load the default stored in the system variable, OCSUSERPATH.

F7 (CANCEL)

Use **F7** to exit this screen without transferring the filename to the calling screen.

F8 (OK)

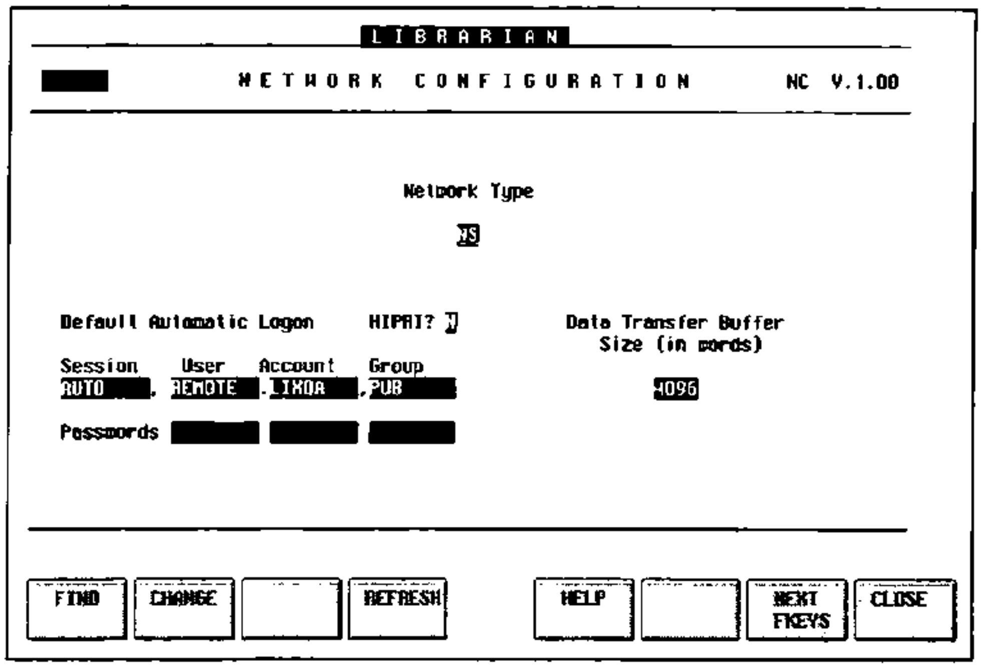
Use **F8** to transfer the filename to the calling screen. You will return to the original screen, and your long pathname will appear in the appropriate fields (truncated for display only, if necessary). The pathname will be carried forward in future invocations of the LP screen.

NETWORK CONFIGURATION NC

The Network Configuration (NC) screen is where you configure LIBRARIAN for network operations.

Menu Access	Admin...Screens...Config...NC Network Configuration
Screen Security	LIBRARIAN Manager
Files Impacted	D-NETWORK-CONFIG

Screen Layout



```
                         L I B R A R I A N
            N E T W O R K   C O N F I G U R A T I O N        NC  V.1.00

                              Network Type
                                  NS

Default Automatic Logon     HIPRI? Y        Data Transfer Buffer
                                              Size (in words)
Session    User    Account   Group
AUTO     , REMOTE . LIXOA   , PUB                  4096

Passwords

 FIND | CHANGE |  | REFRESH |   | HELP |  | NEXT FKEYS | CLOSE
```

Key Fields

None

Description

The data in the network configuration record defines defaults for LIBRARIAN networking operations in an MPE environment. If you are not using the product in a networked environment, you do not need to use this screen.

LIBRARIAN is shipped with initial values already loaded for this screen. You can change these values as needed for your own network environment.

The default automatic logon on this screen is used whenever LIBRARIAN logs on to a remote system. You can override this login for specific systems by creating a record on the Systems (SY) screen.

The maximum buffer size for the NS network type is 4096 words. The maximum buffer size for the (DS) network type is 1023 words. Often, X.25 users are restricted to a buffer size of 138 words (DS) or 1023 words (NS).

Explanation of Fields

Network Type
Required. Length 2.

The type of network you are using. Legal values are DS or NS. The default is NS.

Default Automatic Logon
The default login to use when transactions require automatic remote login. This login is initiated when a remote session has not been previously established. Unless there is an override for a specific system (as specified in the Systems (SY) screen), this login is used.

The automatic login consists of the following subfields:

Session
Required. Length 8.

The session name for automatic remote login. You can use the special wildcard !USERID in the Session field. The default is AUTO.

User
Required. Length 8.

The user name for automatic remote login. The default is REMOTE.

Account
Required. Length 8.

The account name for automatic remote login. The default is OCSLIB.

Group
Required. Length 8.

The group name for automatic remote login. The default is PUB.

Hipri
Required. Length 1.

Indicates if LIBRARIAN logs on to remote systems with HIPRI. A Y in this field indicates that remote logins use HIPRI. An N in this field indicates that remote logins use the default input priority.

Note In order to use **Hipri** for automatic remote logon to LIBRARIAN, MPE requires that the logon user and account have OP or SM capability.

Passwords
Optional. Length 3 fields of 8.

The user, account, and group passwords for the default automatic login. Passwords entered are encrypted in the database. The initial default automatic login does not include passwords.

NETWORK CONFIGURATION (continued) NC

Operation

The ADD, DELETE, and FIRST RECORD function keys are not available on this screen. There is only one NC record. You can use this screen to change pre-loaded values.

When you define the default automatic login, you can specify any login. In addition, you can use the special wildcard !USERID for the session. In this case, the current user ID is substituted as the session for remote login.

PROJECT AUTHORIZATIONS PA

Use the Project Authorizations (PA) screen to enter, review, or delete user authorizations for project activities.

Menu Access

Admin...Screens...Projects...PA Project Authorizations

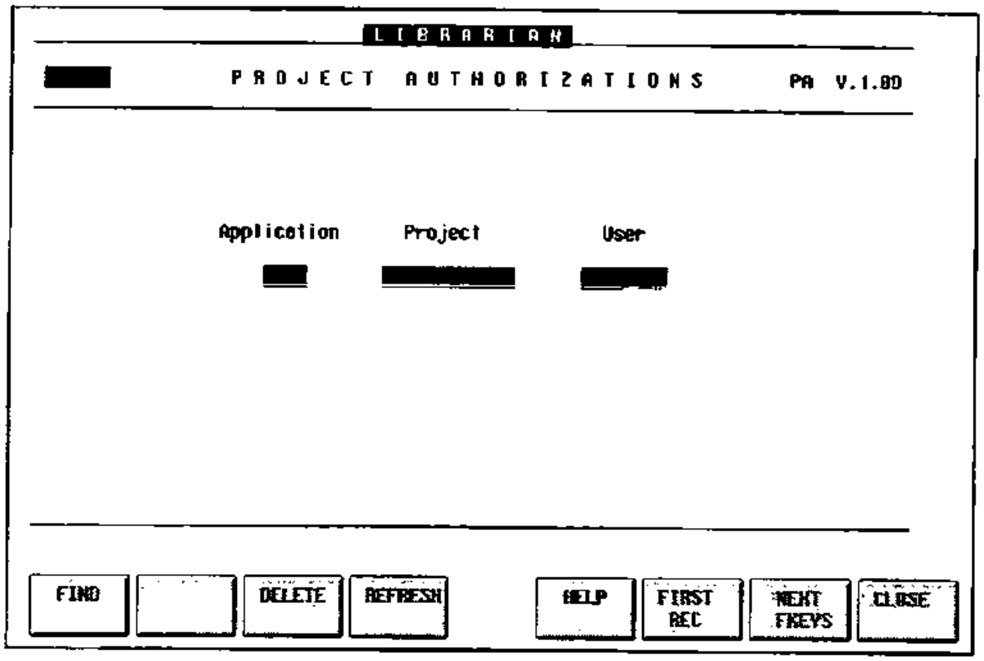
Screen Security

LIBRARIAN Manager, Rule Administrator, Application Manager, or Project Manager. Application Managers can only access the records for their applications. Project Managers can only access the records for their projects.

Files Impacted

D-USER-PROJECT

Screen Layout



Key Fields

Application, Project, User

Description

If the project definition requires project authorization, then users must have specific authorization on this screen. LIBRARIAN managers, Application managers, and Project managers do not require special project authorization for their projects.

PROJECT AUTHORIZATIONS (continued)

Explanation of Fields

Application
Required. Length 4.

The application to which this project belongs, as defined on the Projects (PJ) screen.

Project Name
Required. Length 12.

Name of the project as defined on the Projects (PJ) screen.

User
Required. Length 8.

User authorized to work on this project.

Operation

F2 (CHANGE) is not available on this screen. To change a project authorization, delete the record and then add a new record.

PENDING MASTER FILES

The Pending Master Files (PF) screen is where you can introduce new files to LIBRARIAN in a secondary location, and later check those files in to a specific master location.

Menu Access

Admin...Screens...Files...PF Pending Master FIles

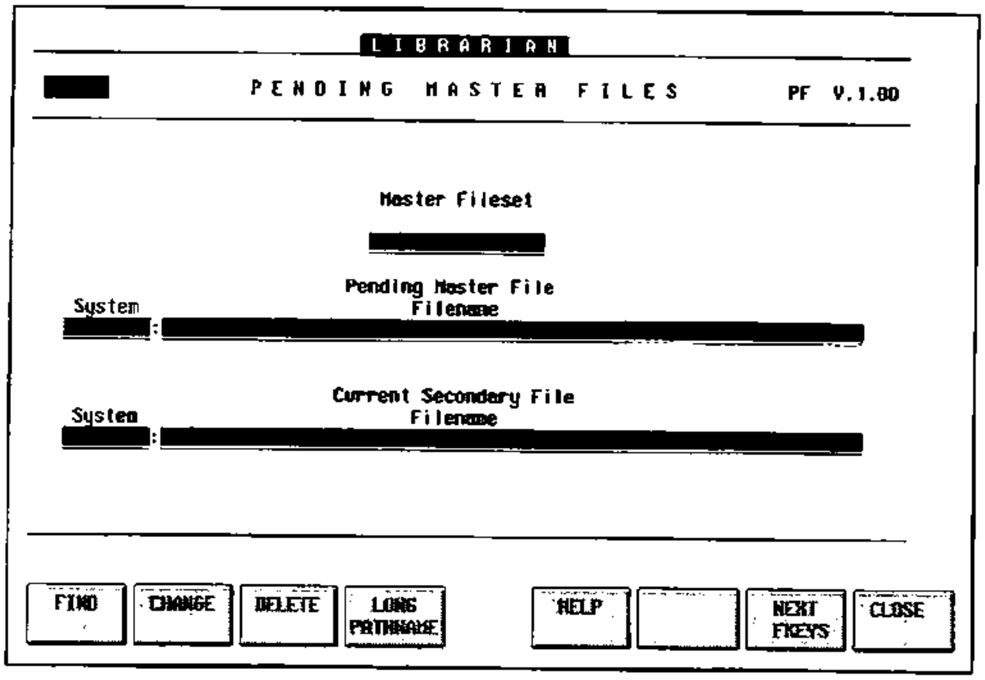
Screen Security

Access to this screen is controlled by the PF ACCESS flag on the System Profile (SP) screen. If PF ACCESS is set to Y, any user can access this screen. If PF ACCESS is set to N, only the LIBRARIAN manager, Rule administrator, or Application manager can access this screen.

Files Impacted

D-FSET-FILE
D-FILE

Screen Layout



Key Fields

Master Fileset, Pending Master File

Description

Define the current location of the secondary file and the eventual location of the master file. Identify the fileset to which it will belong.

The fileset you identify for the pending master file must have S or M default access control, permitting a write mode secondary.

Descriptions, continued

Use pending master files as a way to introduce new files into LIBRARIAN in a secondary location and later move those files to a predefined master location.

LIBRARIAN automatically creates PF records for secondary copies that you check out from, previous versions, and will move forward to a new version (as defined on the FV screen).

In addition, LIBRARIAN creates a PF record when you introduce a new file in a pending production area by applying the edit mask defined on the Pending Production Areas (PP) screen.

Explanation of Fields

Master Fileset
Required. Length 16.

The name of a fileset to which the pending master file will belong. This fileset must be previously defined in the Filesets (FS) screen. The default access control for the fileset (on the FS screen) must permit a write mode secondary file.

Pending Master System
Optional. Length 8.

System where the master file will reside. If you do not specify a system, the default is the current system.

Pending Master File
Required. Length 64 (255 on Long Pathname screen).

The location where the master file will eventually reside. This file cannot already be a member of a fileset or exist on disk.

To enter a pending master filename that is longer than 64 characters, press **F4** (LONG PATHNAME). The Long Pathname (LP) screen appears, allowing you to specify a name with a maximum of 255 characters.

Current Secondary System
Optional. Length 8.

System where the current secondary file resides. If you do not specify a system, the default is the current system.

Current Secondary File
Required. Length 64 (255 on Long Pathname screen).

The current development location of the pending master file. This file cannot already be defined to LIBRARIAN (as a master, secondary, or retained file). This file will be recorded as a write mode secondary of the pending master file.

To enter a current secondary file that is longer than 64 characters, press **F4** (LONG PATHNAME). The Long Pathname (LP) screen appears, allowing you to specify a name with a maximum of 255 characters. After you press F4 (LONG PATHNAME), the Pending Master File field appears on the LP screen. To go to the Current Secondary File field, press **F1** (TOGGLE) on the LP screen. For more information, refer to the LP screen.

Operation

To delete a record before moving it to the master file location, first find the record with F1 and then delete it with F3.

If you decide that you no longer want a file to be a master file, but the file has been moved to the master file location, use the Files in Filesets (FF) screen to delete its association with the fileset.

PROJECT INQUIRY

The Project Inquiry (PI) screen is where you can get information about projects related to an application.

Menu Access

Admin...Screens...Projects...PI Project Inquiry

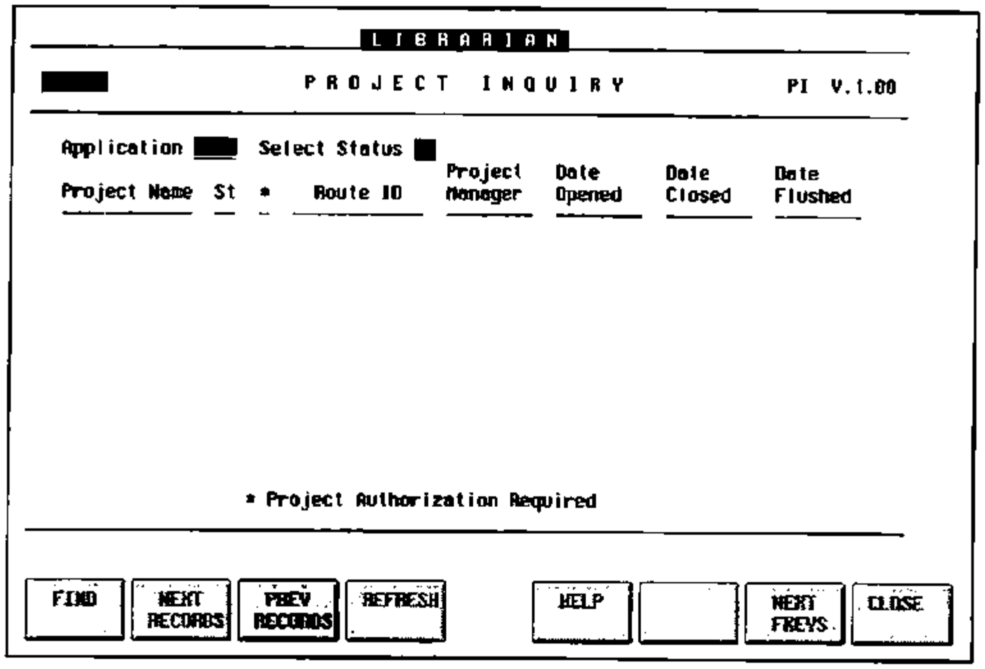
Screen Security

None

Files Impacted

D-PROJECTS

Screen Layout



```
                         L I B R A R I A N
                      P R O J E C T   I N Q U I R Y                 PI  V.1.00

  Application      Select Status
                                          Project   Date     Date     Date
  Project Name  St  *    Route ID         Manager   Opened   Closed   Flushed

                   * Project Authorization Required
```

Key Fields

Application

Description

By using this screen, you can select projects to review based on their status.

PROJECT INQUIRY (continued)

Explanation of Fields

Application
Required. Length 4.

Application to which this project belongs, as defined on the Applications (AP) screen.

Select Status
Optional. Length 2.

Specify one of the following:

AL	All projects	CL	Closed to all steps
AA	All active projects	DC	Documented
AI	All inactive projects	FP	Flush Pending
AO	All open projects	FL	Flushed
AC	All closed projects	OP	Opened
CC	Closed to CHECKOUT steps	RO	Reopened

Project Name
Display. Length 12

Name of the project as defined on the Projects (PJ) screen.

St
Display. Length 2.

Status code for project. Displays one of the following:

CC	Closed to Checkout steps	CL	Closed to all steps
DC	Documented	FP	Flush Pending
FL	Flushed	OP	Opened
RO	Reopened		

*****
Display. Length 1.
An asterisk in this column indicates that project authorization is required.

Route ID
Display. Length 12.

The name of the route to which the project belongs.

Project Manager
Display. Length 8.

The LIBRARIAN user who is responsible for the project.

Explanation of Fields, continued

Date Opened
Display. Length 8.

The date the project was opened.

Date Closed
Display. Length 8.

The date the project was closed.

Date Flushed
Display. Length 8.

The date the project was flushed.

Operation

The CHANGE, DELETE, and FIRST RECORD functions are not available on this screen. To retrieve project records, enter the application. If you want to review projects based on their status, enter a selection in the Select Status field. Then, use **F1** (FIND)to retrieve the records.

F2 and F3 have the following special functions on this screen.

NEXT RECORDS (F2)

If the application contains more projects than can be presented on one screen, press **F2** to display the next page of projects.

PREV RECORDS (F3)

If the application contains more projects than can be presented on one screen, press **F3** to display the previous page of projects.

Projects on the PI screen are sorted in reverse date sequence (i.e., most recent projects are shown first).

PROJECTS

The Projects (PJ) screen is where you define projects.

Menu Access

Admin...Screens...Projects...PJ Projects

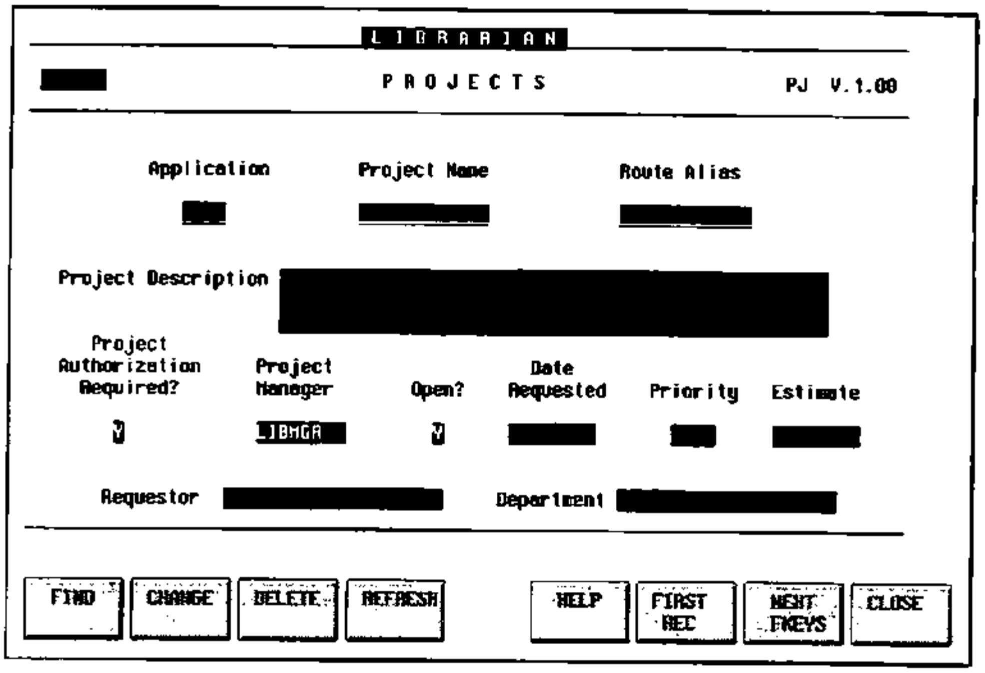
Screen Security

LIBRARIAN Managers, Rule Administrator, Application Managers, and Project Managers. Application Managers can only access the records for their applications. Project Managers can only access the records for their projects. All users can access this screen for inquiry.

Files Impacted

D-PROJECTS
M-FILE-SET
D-USER-FSET

Screen Layout



Application
Project Name
Route Alias
Project Description
Project Authorization Required? Y
Project Manager LIBMGR
Open? Y
Date Requested
Priority
Estimate
Requestor
Department

Key Fields

Application, Project Name

Description

Use this screen to define projects, review, or modify project records.

Explanation of Fields

Application
Required. Length 4.

The application name to which this project belongs.

Explanation of Fields, continued

Project Name
Required. Length 12.

The name of the project. This name is also assigned to the project fileset.

Route Alias
Required. Length 12.

The route for which this project is valid. Use the at sign (@) wildcard to indicate the project is valid for all routes in the application.

Project Description
Optional. Length 150.

A description of the project.

Project Authorization Required?
Required. Length 1.

A flag that indicates whether a user must be authorized to work on this project as defined in the Project Authorizations (PA) screen. Legal values are:

Y	Indicates that users must be authorized to reference this project (default).
N	Indicates that no authorization is required to reference this project.

Project Manager
Required. Length 8.

The user responsible for the project. The user must have Project manager, Application manager, or LIBRARIAN manager capability. The default is the current user.

Open?
Required. Length 1.

A flag to indicate whether the project should initially be open. Possible values include:

Y	Indicates that the initial project status is open (OP) (default).
N	Indicates that the initial project status is documented (DC).

Date Requested
Optional. Length 8.

For documentation only.

Priority
Optional. Length 4.

For documentation only.

Estimate
Length 8.

For documentation only.

Explanation of Fields, continued

Requestor
Length 20.

For documentation only.

Department
Length 20.

Operation

The project name defined on this screen can be referenced in a LIBRARIAN command instead of a route. If the project is not specified and projects are required for the route, then a menu of projects is displayed.

A project fileset is created with the same name as the project. This fileset is a private user fileset owned by the Project manager. If you do not want to require specific project authorization, change the Project Authorization Required field to N.

If you require project authorization, use the Project Authorizations (PA) screen to identify the authorized users.

If you want to use this project immediately, set the Open flag to Y. If you want to define the project for later use, set the flag to N.

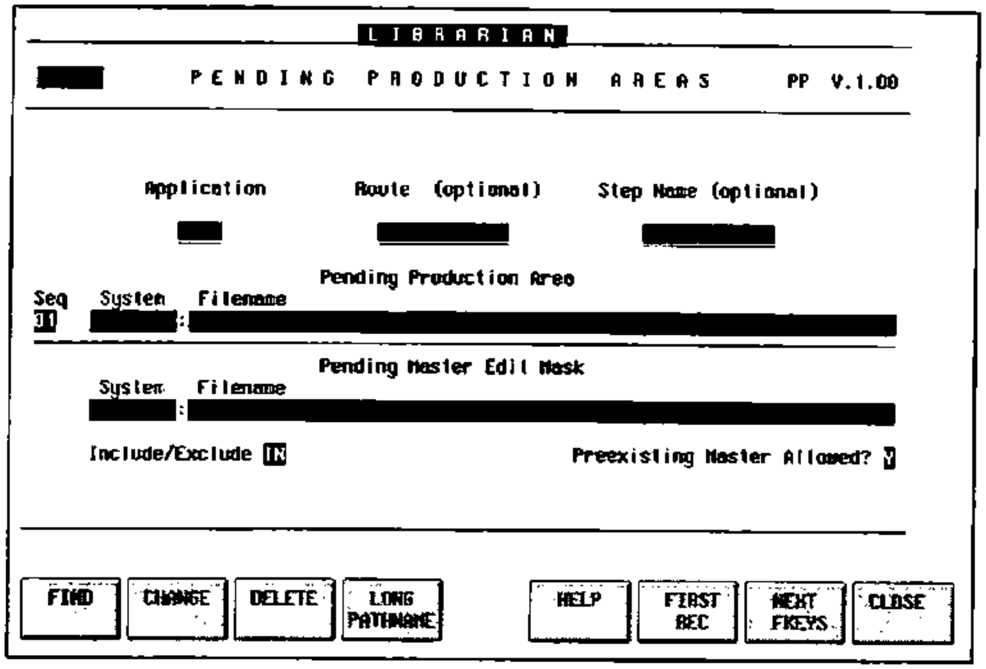
F3 (DELETE) is allowed on this screen only if the project is flushed. A project is set to the Flush-Pending state on the Project Status (PS) screen. A project is flushed by the FLUSHLOG utility program.

PENDING PRODUCTION AREAS PP

The Pending Production Areas (PP) screen is where you define general secondary locations where new files can be introduced.

Menu Access	**Admin...Screens...Steps...PP Pending Production Areas**
Screen Security	LIBRARIAN Manager, Rule Administrator, and Application Manager. Application Managers can only access the records for their applications.
Files Impacted	D-PENDING-AREA

Screen Layout



LIBRARIAN

PENDING PRODUCTION AREAS PP V.1.00

Application Route (optional) Step Name (optional)

Pending Production Area

Seq System Filename

01 :

Pending Master Edit Mask

System Filename

:

Include/Exclude IN Preexisting Master Allowed? Y

Key Fields

Application, Route, Step Name

Description

Use this screen to define where and when users can introduce new secondary files. You define the step, route, and/or application that users can use to introduce new files, and from what location. Each pending production area is defined in terms of areas to be included or excluded.

If a pending master editmask is provided, then a pending master record is created automatically. This occurs when a file is introduced through the pending production area. If no pending mask is provided, users are responsible for identifying the master file when they perform the step that introduces the new file.

Explanation of Fields

Application
Required. Length 4.

The application to which this pending production area applies, as defined on the Applications (AP) screen.

Route
Optional. Length 12.

The route to which this pending production area applies. If this field is left blank, the pending production area is valid for all routes in the application.

Step Name
Optional. Length 12.

The step to which this pending production area applies. If this field is left blank, the pending production area is valid for all steps in the route.

Seq
Required. Length 2.

Specifies the sequence that LIBRARIAN uses to check pending production areas. Valid sequence numbers range from 0 to 99. If you define multiple overlapping areas, use this field to identify the sequence in which you want LIBRARIAN to check.

Pending Production Area (System:Location)
Required. Length 8 for system; 64 for location (255 on Long Pathname screen)

The general location where new secondary files can be introduced. Wildcards consistent with MPE/iX and HP-UX conventions are accepted.

To enter a pending production area that is longer than 64 characters, press **F4** (LONG PATHNAME). The Long Pathname (LP) screen appears to let you specify a name with a maximum of 255 characters.

Pending Master Edit Mask (System:Filename)
Required. Length 8 for system; 64 for filename (255 on Long Pathname screen)

An edit mask is used to translate a pending production secondary filename into a pending master filename. This automatically creates a pending master record. For more information on edit masks, refer to *Edit Masks* at the beginning of Chapter 1, "Commands."

To enter a pending master edit mask that is longer than 64 characters, press **F4** (LONG PATHNAME). The Long Pathname (LP) screen appears to let you specify a name with a maximum of 255 characters. After pressing **F4** (LONG PATHNAME), the Pending Production Area field appears on the LP screen. To go to the Pending Master Edit Mast field, press **F1** (TOGGLE) on the LP screen. For more information, refer to the LP screen.

Include/Exclude
Required. Length 2.

A flag to indicate whether the specified area should be included or excluded from the files for this pending production area.

Operation, continued

Preexisting Master Allowed?
Required. Length 1.

A flag to indicate whether a new file can be introduced when it maps to the name of an already existing master file. Legal values are:

Y	Okay that a master exists in the pending master location.
N	Not okay that a master exists in the pending master location.

Operation

Creates a separate PP record for each area to be included or excluded from the definition of a pending production area. For example, if the area includes @.@.FIN and @.JOB.AP but excludes PR@.JOB.AP, use three PP records to define the pending production area.

PROJECT STATUS CHANGE

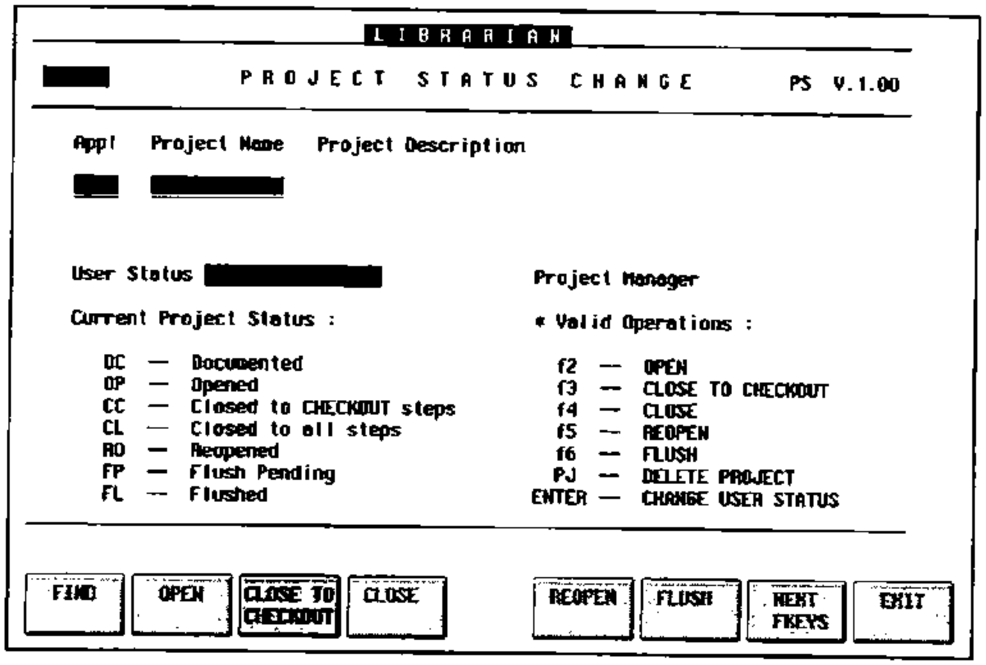
The Project Status Change (PS) screen lets you review and change the status of a project.

Menu Access Admin...Screens...Projects...PS Project Status Change

Screen Security LIBRARIAN Manager, Rule Administrator, Application Manager, or Project Manager. Application Managers and Project Managers can only access the records for their own applications.

Files Impacted D-PROJECTS

Screen Layout



```
                          L I B R A R I A N

              P R O J E C T   S T A T U S   C H A N G E        PS  V.1.00

  Appl   Project Name   Project Description



  User Status                                  Project Manager

  Current Project Status :                     * Valid Operations :

     DC  —  Documented                           f2   —  OPEN
     OP  —  Opened                               f3   —  CLOSE TO CHECKOUT
     CC  —  Closed to CHECKOUT steps             f4   —  CLOSE
     CL  —  Closed to all steps                  f5   —  REOPEN
     RO  —  Reopened                             f6   —  FLUSH
     FP  —  Flush Pending                        PJ   —  DELETE PROJECT
     FL  —  Flushed                            ENTER  —  CHANGE USER STATUS
```

Key Fields

Appl, Project Name

Description

Use this screen to review and modify the status of a project.

Explanation of Fields

Appl
Required. Length 4.

The application to which the project belongs.

Explanation of Fields, continued

Project Name
Required. Length 4.

The name of the project as defined on the Projects (PJ) screen.

Project Description
Display. Length 150.

The description of the project.

User Status
Optional. Length 16.

A free–form user–defined status value.

Project Manager
Display. Length 8.

The LIBRARIAN user who is the Project Manager.

Current Project Status
Display. Length 12.

The current status of the project.

Operation

HELP for this screen is available through F6 on function key set 2. The ADD, CHANGE, and FIRST RECORD functions are not available.

Add a user defined status value by pressing ENTER.

Special function keys on this screen modify project status; the current project status reflects any change you make. The operations available for a project depend on it current status. Operations that are currently valid appear on the screen with an asterisk (*). Descriptions of the special function keys are below:

OPEN

Use **F2** to open the project if the status is documented (DC).

CLOSE–TO–CHECKOUTS

Use **F3** to close a project to checkout (master–to–secondary) steps.

CLOSE

Use **F4** to close an opened or reopened project.

Note LIBRARIAN rejects attempts to close projects with associated write mode secondaries.

Operation, continued

REOPEN

Use F5 to reopen a project that you closed or made "flush pending" (but not flushed).

FLUSH

Use F6 to designate a project as flush pending. When the FLUSHLOG utility runs, it flushes the project fileset and all transaction records that refer to the project.

ROUTES

The Routes (RT) screen is where you define routes.

Menu Access

Admin...Screens...Steps...RT Routes

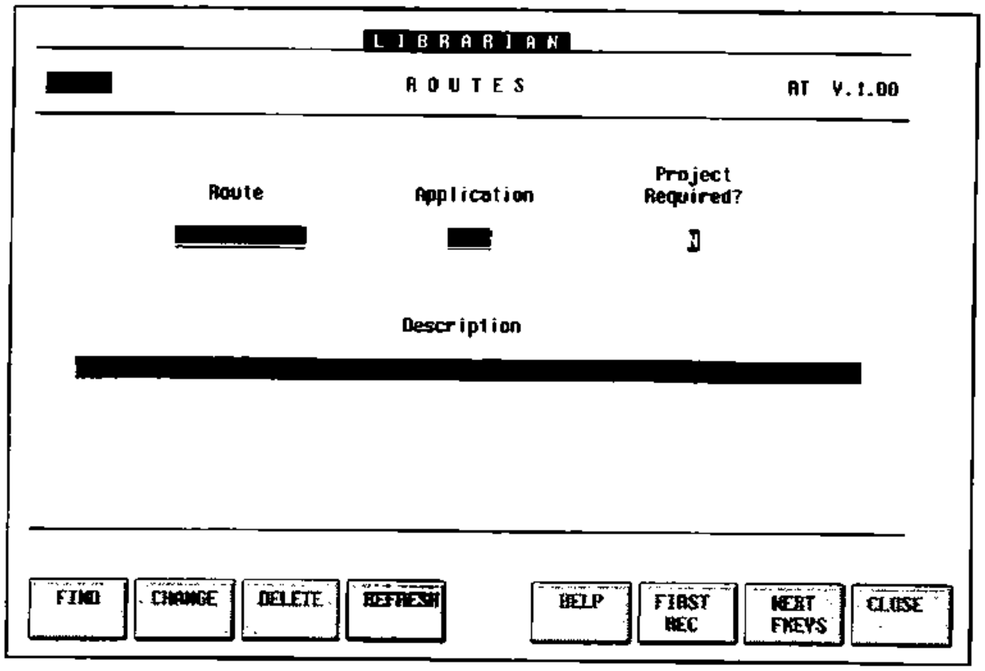
Screen Security

LIBRARIAN Manager, Rule Administrator, and Application Manager. Application Managers can only access the records for routes associated with their own applications.

Files Impacted

D-ROUTE

Screen Layout



Key Fields

Route, Application

Description

A route is a collection of steps for copying and moving files in a standard sequence. Define each step separately on the Steps (ST) screen.

The route and application uniquely identify the route. Different applications can use the same route name.

The Project Required field lets you specify whether transactions in the route must be associated with a project.

ROUTES (continued)

Explanation of Fields

Route
Required. Length 12.

The identifier of the route. The route can include alphabetic, numeric, hyphen (-), and underscore (_) characters. The route must be unique within its application.

Application
Required. Length 4.

The application to which this route belongs.

Project Required?
Required. Length 1.

A flag indicating if step transactions in this route must be associated with a project. Legal values are:

Y	Indicates that projects must be identified.
N	Indicates that no project identification is required (default).

Description
Optional. Length 72.

A description of the route.

Operation

Before you can delete the record for a route, use the Steps (ST) screen to delete the records for the steps associated with the route.

STEP AUTHORIZATIONS

The Step Authorizations (SA) screen is where you authorize users to perform a step.

Menu Access

Admin...Screens...Steps...SA Step Authorizations

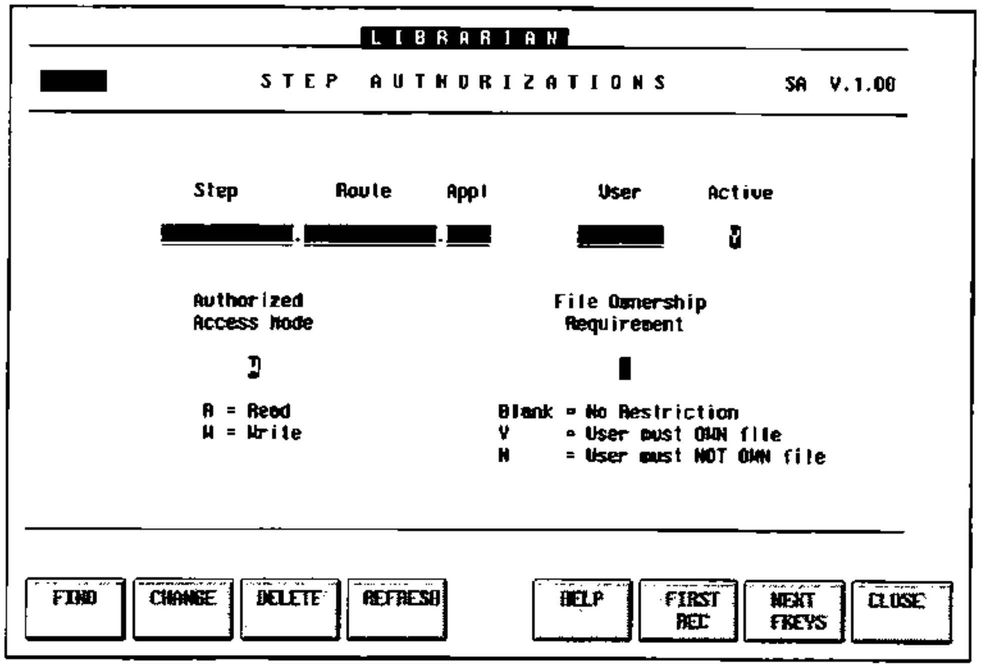
Screen Security

LIBRARIAN Manager and Application Manager. Application Managers can only access the records for steps associated with their own applications.

Files Impacted

D-USER-STEP

Screen Layout



```
                          L I B R A R I A N
          S T E P   A U T H O R I Z A T I O N S        SA  V.1.00

     Step        Route     Appl        User     Active

     Authorized                      File Ownership
     Access Mode                      Requirement

     A = Read                    Blank = No Restriction
     W = Write                   Y     = User must OWN file
                                 N     = User must NOT OWN file
```

Key Fields

Step, Route, Appl, User

Description

You can authorize any number of users to perform a step. Create a separate record for each user/step authorization. In addition, you can use this record to specify the type of copies the user can obtain with the step, if different from the step definition.

Use the Active flag to suspend and reinstate a user's access to the step at any time, without deleting the record from the database.

You do not need to authorize LIBRARIAN managers and Application managers to perform steps. The LIBRARIAN manager can perform all steps. Application managers can perform all steps associated with their own applications.

Explanation of Fields

Step
Required. Length 12.

The name of the step. You must have already defined the step on the Steps (ST) screen. You can use an at sign (@) to authorize a user for all steps within a route.

Route
Required. Length 12.

The unique identifier of the route to which this step belongs. You must have already defined the route on the Routes (RT) screen. You can use an at sign (@) to authorize a user for all steps within an application.

Appl
Required. Length 4.

The application to which the route and step belong. You must have already defined the application on the Applications (AP) screen.

User
Required. Length 8.

The user authorized to perform this step. You must have already defined the user on the Users (US) screen.

To authorize all users for a step, change the Authorization Required field to N on the Steps (ST) screen.

Active
Required. Length 1.

A flag that indicates if a user is currently authorized to perform this step. Legal values are:

Y	Indicates that the user is currently authorized to perform the step (default).
N	Indicates that the user is not currently authorized to perform the step.

Authorized Access mode.
Required. Length 1.

A flag that indicates the mode available to a user for files created by the step. Legal values are:

R Read	Indicates that the user can only obtain read mode copies.
W Write	Indicates that the user can obtain read or write mode copies.

Explanation of Fields, continued

File Ownership Requirement
Optional. Length 1.

A flag that indicates whether the user must own a file to be authorized to perform the step on it. Legal values are:

Blank	Indicates no file ownership restriction for the user.
Y	Indicates that the user must be the file owner.
N	Indicates that the user must not own the file.

Operation

You can authorize a user to perform all steps in a route by using the at sign (@) in the Step field. To select all routes in an application use the at sign (@) in the Route field as well.

SYSTEM PROFILE

The System Profile (SP) screen is where you define parameters that affect global LIBRARIAN operation.

Menu Access

Admin...Screens...Config...SP System Profile

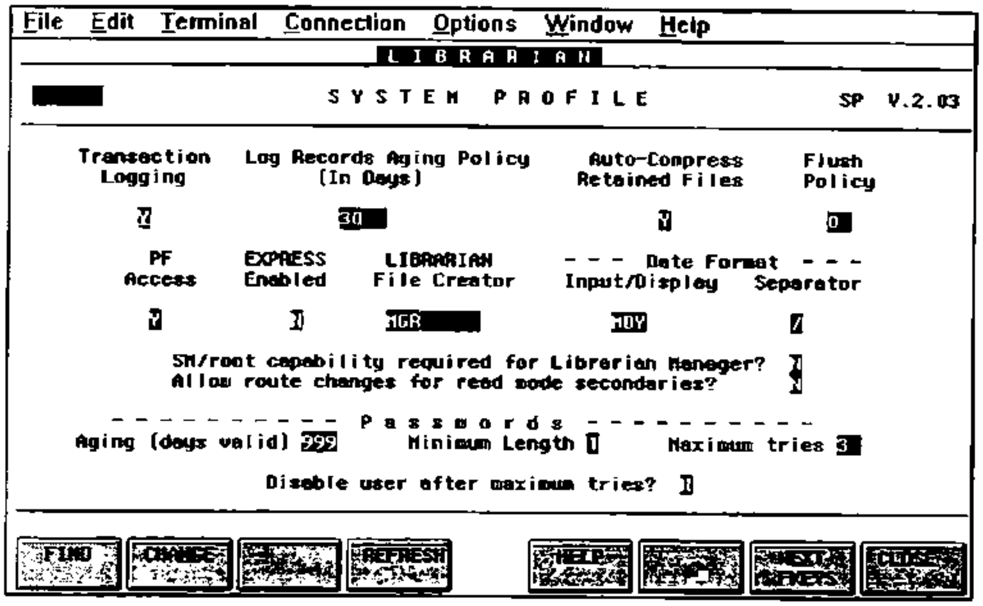
Screen Security

LIBRARIAN Manager

Files Impacted

D-SYSTEM-PROFILE

Screen Layout



```
                         L I B R A R I A N

                  S Y S T E M   P R O F I L E              SP  V.2.03

   Transaction   Log Records Aging Policy    Auto-Compress    Flush
    Logging           (In Days)              Retained Files   Policy

       Y                 30                       N             0

      PF       EXPRESS     LIBRARIAN     - - -  Date Format  - - -
    Access     Enabled    File Creator   Input/Display   Separator

       Y          N        MGR               MDY             /

          SM/root capability required for Librarian Manager?  Y
          Allow route changes for read mode secondaries?      N

   - - - - - - - - - - -  P a s s w o r d s  - - - - - - - - - -
  Aging (days valid) 999     Minimum Length 0      Maximum tries 3

                  Disable user after maximum tries?  N
```

Key Fields

None

Description

The system profile is a set of global parameters that the LIBRARIAN Manager maintains, controlling how LIBRARIAN operates. Includes items such as flush policy, aging policy, date formats, etc.

Explanation of Fields

Transaction Logging
Required. Length 1.

A flag to activate the LIBRARIAN Audit Trail Logging Facility. Legal values are:

Y	Indicates that Audit Trail Logging is active. LIBRARIAN command activity is tracked (default).
N	Indicates that Audit Trail Logging is inactive. LIBRARIAN command activity is not tracked.

Log Records Aging Policy (in Days)
Required. Length 4.

The number of days you want audit trail records to be saved before they are purged by FLUSHLOG. The FLUSHLOG utility uses this value to select associated records to be purged. This value applies to all transaction records except those identified with projects. Project transactions are retained until the project is flushed. Possible values are 1 to 9999. The default is 30 days.

Auto-Compress Retained Files
Required. Length 1.

A flag to indicate if retained files should be compressed automatically. This does not apply to files retained as deltas. Legal values are:

Y	Compresses files when they are retained (default).
N	Does not compress files when they are retained.

Flush Policy
Required. Length 2.

The maximum number of generations of a master file to be preserved, when running the FLUSH utility. Acceptable values range from 0 to 99. The default is zero generations.

PF Access
Required. Length 1.

A flag to let users access the Pending Master Files (PF) screen to add or modify a pending master file record. Legal values are:

Y	Indicates that any user can define Pending Master Files (default).
N	Indicates that only the LIBRARIAN manager can define Pending Master Files.

EXPRESS Enabled
Required. Length 1.

A flag to indicate whether batch file transactions will be scheduled automatically with the EXPRESS SUBMIT utility (only available if EXPRESS is installed). Legal values are:

Y	Schedules all batch requests by using the EXPRESS SUBMIT utility.
N	Does not schedule batch requests with the EXPRESS SUBMIT utility (default).

Explanation of Fields, continued

LIBRARIAN File Creator
Required. Length 8.

The default user that LIBRARIAN will use when pushing a file from one account to another. This user becomes the file creator for the new file. The default is MGR.

Date Format Input/Display
Required. Length 3.

The format for all dates used by LIBRARIAN. Legal values are:

MDY	Month, Day, Year order (default)
DMY	Day, Month, Year order
YMD	Year, Month, Day order

Date Format Separator
Required. Length 1.

The symbol used as a separator in all date formats. Legal values are:

/	(slash is the default)
-	(hyphen)
.	(period)

SM/root capability required for LIBRARIAN Manager?
Required. Length 1.

A flag to indicate whether LIBRARIAN Manager users need to login with MPE's SM or UNIX's root capability. Legal values are:

Y	Indicates that MPE/UNIX user must have SM/root capability.
N	Indicates that MPE/UNIX user is not required to have SM/root capability (default).

Note

If you attempt to change the "SM capability required" switch to "Y", and you do not have SM capability (MPE) or superuser privileges (UNIX), an error occurs. This protects you from inadvertently locking yourself out of LIBRARIAN.

Allow route changes for read mode secondaries?
Required. Length 1.

A flag to determine if you can copy or move read mode secondaries by using a step in a route different from the one originally used. By default, attempting to change routes results in a violation. Legal values are:

Y	Indicates that changing routes for read mode secondaries is allowed.
N	Indicates that changing routes for read mode secondaries is not allowed (default).

SYSTEM PROFILE (continued) SP

Explanation of Fields, continued

Aging (Days Valid)
Required. Length 3.

The number of days passwords are valid.

Passwords can expire between 1 and 500 days. The default expiration period is 999 days, indicating that user passwords will not expire.

Minimum length
Required. Length 8.

The minimum number of characters required for each LIBRARIAN user password.

You can set passwords to have a minimum length of 1 to 8 characters. The default minimum password length is 1 character.

Maximum tries
Required. Length 2.

The maximum number of times a user can attempt to enter a valid password.

If a user exceeds this number of attempts and the "Disable user after maximum tries?" field is set to "Y", LIBRARIAN will disable the user ID. The LIBRARIAN Manager will then have to re–activate the user on US screen.

You can set the maximum number of attempts to a value between 1 and 16. The default is 3 attempts.

Disable user after maximum tries?
Required. Length 1.

If this field is set to "Y", when the user exceeds the maximum number of attempts to enter a valid password, LIBRARIAN will disable the user ID by setting the active flag to "N". The LIBRARIAN Manager can re–activate the user on the US screen.

STEP REFINEMENTS/EXCEPTIONS SR

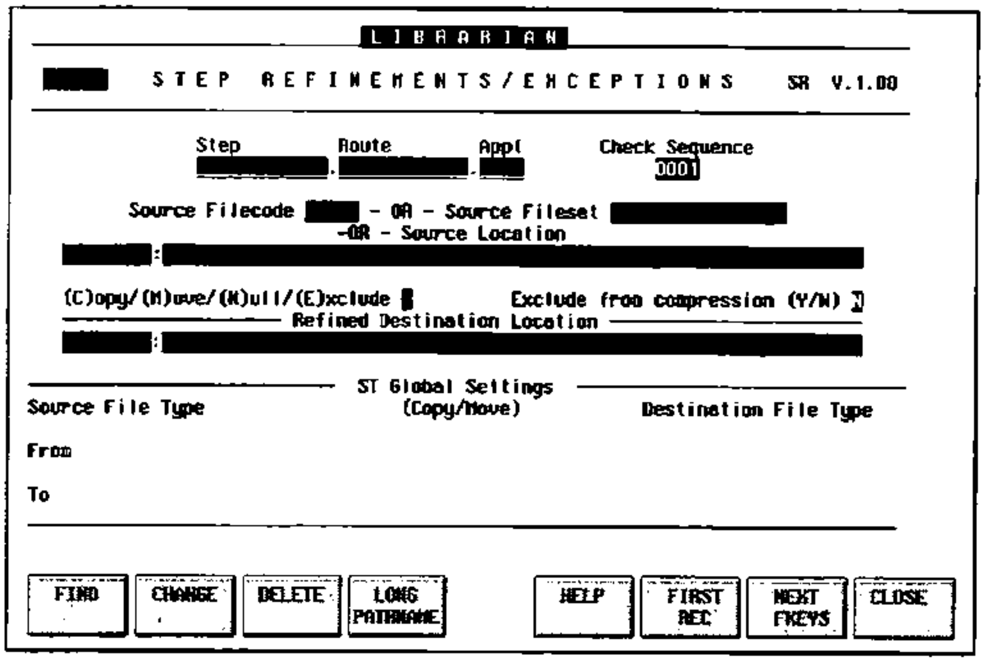
The Step Refinements/Exceptions (SR) screen is where you refine a step definition based on a file's name, fileset, or filecode.

Menu Access **Admin...Screens...Steps...SR Step Refinement/Exceptions**

Screen Security LIBRARIAN Manager, Rule Administrator, and Application Manager. Application Managers can only access the records for steps in their own applications.

Files Impacted D-REFINED-STEP

Screen Layout



```
                         LIBRARIAN
      STEP REFINEMENTS/EXCEPTIONS            SR V.1.00

          Step        Route      Appl     Check Sequence
                                                0001
     Source Filecode      - OR - Source Fileset
                     -OR - Source Location
         :
  (C)opy/(M)ove/(N)ull/(E)xclude      Exclude from compression (Y/N)
                 Refined Destination Location
         :
                      ST Global Settings
Source File Type         (Copy/Move)          Destination File Type
From
To
```

Key Fields

Step, Route, Appl

Description

You can refine or create exceptions for a step by defining different destinations and movement types for certain files. In addition, you can specify files to be excluded from the step.

Enter the step, route, application, and then use **F1** to load the global step definition from the Steps (ST) screen. Then, enter either a general source location using wildcards, filecode, or fileset to refine. Finally, specify a refined movement type and/or destination location.

If you specify N (null), the destination location must be the same as the source location. If you specify E (exclude), the destination location must be blank.

Description, continued

If you have more than one SR record for a step, specify the sequence for checking the definitions since a file could qualify for more than one refinement.

A Null refinement causes the step transaction to be recorded without moving/copying the file. An Exclude refinement causes the file to be excluded from the step, so that no log record is created for that file.

You can retrieve step refinements by specifying the file code, fileset, or source location and pressing F1.

Explanation of Fields

Step
Required. Length 12.

The name of the step being refined.

Route
Required. Length 12.

The route to which the step belongs.

Appl
Required. Length 4.

The application to which this route and step belong.

Check Sequence
Required. Length 4.

If you define multiple step refinements, use this field to identify the sequence to check the definitions.

MPE **Source Filecode**
Optional. Length 5.

Apply refinement/exception to files that have this filecode or mnemonic.

Source Fileset
Optional. Length 16.

Apply refinement/exception to files belonging to this fileset.

Source Location (System)
Optional. Length 8.

The system to which this step refinement/exception applies. If you do not specify a system, LIBRARIAN uses the current system as the default.

Source Location (Filename)
Optional. Length 64 (255 on Long Pathname screen).

The source file(s) to which this step refinement/exception applies. Wildcards consistent with MPE/iX and UNIX conventions are accepted.

To enter a source location that is longer than 64 characters, press **F4** (LONG PATHNAME). The Long Pathname (LP) screen appears to let you specify a name with a maximum of 255 characters.

Explanation of Fields, continued

(C)opy/(M)ove/(N)ull/(E)xclude
Required. Length 1.

The movement type override for qualifying files. Legal values are:

C	Copy
M	Move
N	Null
E	Exclude

Destination Location (System)
Optional. Length 8.

The system where the file should be created. If you do not specify a system, LIBRARIAN uses the current system as the default.

Destination Location (Filename)
Required. Length 64 (255 on Long Pathname screen).

The refined destination location or exception for qualifying files. You can use an edit mask to translate source filenames into destination filenames. For more information on edit masks, refer to *Edit Masks* at the beginning of Chapter 1, "Commands."

To enter a refined destination location that is longer than 64 characters, press **F4** (LONG PATHNAME). The Long Pathname (LP) screen appears to let you specify a name with a maximum of 255 characters. After you press **F4** (LONG PATHNAME), the Source Location field appears on the LP screen. To go to the Refined Destination Location field, press **F1** (TOGGLE) on the LP screen. For more information, refer to the LP screen.

Exclude from compression
Required. Length 1.

A flag allowing you to exclude files from compression. Legal values are:

Y	Indicates that files should be excluded.
N	Indicates that files should not be excluded (default).

SYSTEM-TO-SYSTEM TABLE

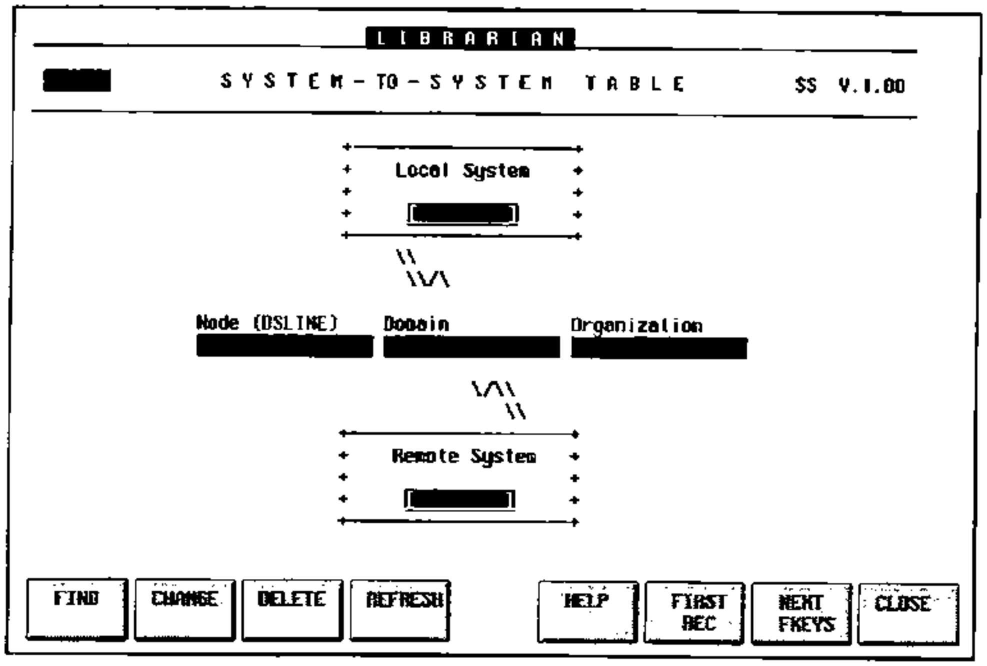
The System–To–System Table (SS) screen is where you identify the node used for communication between two specific systems.

Menu Access Admin...Screens...Config...SS System–To–System Table

Screen Security LIBRARIAN Manager

Files Impacted D-COM-LINKS

Screen Layout



Key Fields

None

Description

This information is necessary if the remote system is referenced by different node names, depending on the origin.

Explanation of Fields

Local System

Required. Length 8.

The local system name. If you do not specify a system, LIBRARIAN uses the current system as the default.

Node (DSLINE)

Required. Length 16.

The actual device ID used for communication between this local system and the remote system. You can use periods in node names to represent system identifiers with more than three elements.

Explanation of Fields, continued

Domain
Optional. Length 16.

The domain associated with the specified node (NS only).

Organization
Optional. Length 16.

The organization associated with the specified node (NS only).

Remote System ID
Required. Length 8.

The remote system name. If you do not specify a system, LIBRARIAN uses the current system as the default.

STEPS

The Steps (ST) screen is where you define file movement commands (rules).

Menu Access

Admin...Screens...Steps...ST Steps

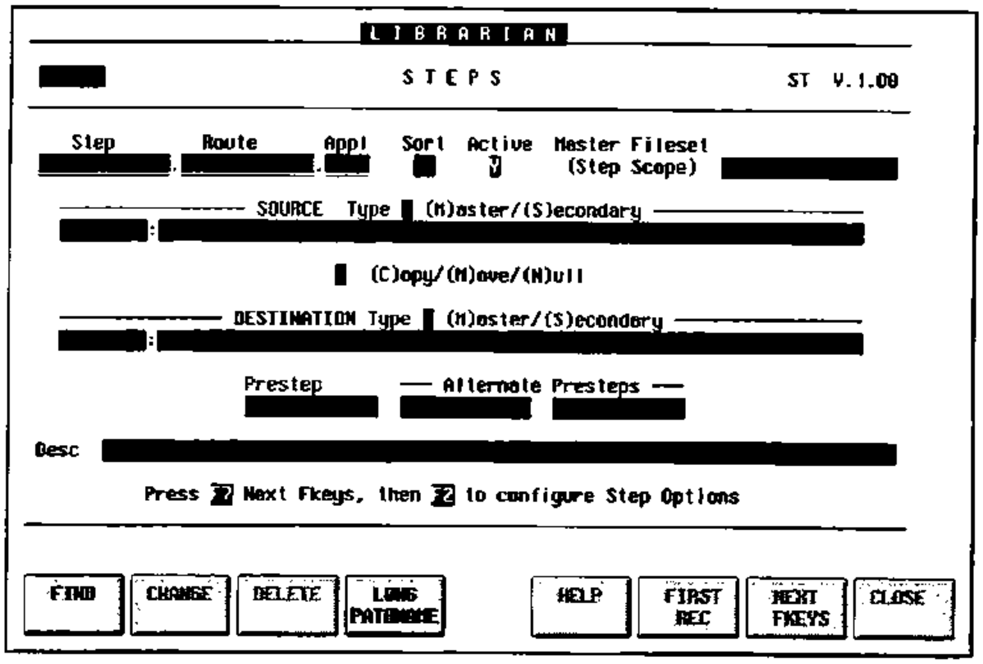
Screen Security

LIBRARIAN Manager, Rule Administrator, and Application Manager. Application Managers can only access the records in their own applications.

Files Impacted

D-STEP
D-SYSTEM-PROFILE

Screen Layout



Press F2 Next Fkeys, then F2 to configure Step Options

Key Fields

Step, Route, Appl

Description

A step is the movement of a file or group of files from a source location to a target location. Steps can also be used to record an external event, such as approval of a set of files. Each step is part of a defined route.

The information on this screen uniquely identifies the step. While defining a step, you identify it's application, route, and fileset to which the step applies. In addition, you can define presteps that users must perform before this step.

Description, continued

By using the Active flag on this screen, you can suspend and reinstate the step. You can change the flag from active to inactive at any time. This way, you can disable the step without deleting the record from the database or removing authorization of the step for all users.

Use the Step Refinements (SR) screen to further refine the definitions on this screen.

The source and destination file information on this screen include the file locations, the types of files, and the type of operation.

You can configure additional step options using the Step Options (STO) screen to further control the behavior of the step.

Explanation of Fields

Step
Required. Length 12.

The name of the step. The step name is a user-defined LIBRARIAN command. The step name can include alphabetic, numeric, hyphen (-), and underscore (_) characters.

Route
Required. Length 12.

The route to which the step belongs. You must have previously defined the route on the Routes (RT) screen.

Appl
Required Length 4.

The application to which the step belongs. You must have previously defined the application on the Applications (AP) screen.

Sort
Required. Length 2.

The number of the step you are defining. Used for reporting steps in a particular sequence within a route. The value can range from 1 to 99.

Active
Required. Length 1.

A flag indicating whether or not the step is currently active. Legal values are as follows:

Y	Indicates that the step is active. Authorized users can perform this step (default).
N	Indicates that the step is inactive. No users can perform this step.

Master Fileset (Step Scope)
Required. Length 16.

The fileset to which the step applies, as defined on the Filesets (FS) screen. This name must be the application fileset or any component. The step only applies to files within this fileset.

Explanation of Fields, continued

SOURCE LOCATION

Type
Required. Length 1.

Specifies whether the source file type is a master (M) or secondary (S).

System
Optional. Length 8.

The system where the file(s) reside. If you do not specify a system, LIBRARIAN uses the current system as the default.

Filename
Required. Length 64 (255 on the Long Pathname screen).

The source file(s) to which this step applies. Wildcards consistent with MPE/iX and HP-UX conventions are accepted.

When performing the step, any elements of the source location the user omits are filled–in using step values defined here.

Additionally, you can use three special LIBRARIAN wildcards in any element of the Source Location:

!USERID	The LIBRARIAN user performing the step.
!LOCAL	The user's local login group and/or account.
!LOGON	Current login path element.
!hpvar	Any MPE system variable that is prefixed by a !, e.g., !HPHGROUP. The value is determined when a user performs the step.

To enter a source location longer than 64 characters, press **F4** (LONG PATHNAME). The Long Pathname (LP) screen appears to let you specify a name with a maximum of 255 characters.

(Copy/Move/Null)
Required. Length 1.

The type of operation this step performs. Legal values are:

C	Copy	Copies the file without purging the source.
M	Move	Moves the file and purges the source. If the source file is a master file, the movement type cannot be M (Move).
N	Null	Performs a step with no movement. A null step (approval step) does not involve actual movement. Although the file is not actually moved, LIBRARIAN records the event and recognizes this as a prestep.

DESTINATION LOCATION

Type
Required. Length 1.

Specifies whether the destination file type is a master (M) or secondary (S).

STEPS (continued)

Explanation of Fields, continued

System
Optional. Length 8.

The system where the source file(s) are to be copied or moved. If you do not specify a system, LIBRARIAN uses the current system as the default.

Filename
Required. Length 64 (255 on the Long Pathname screen).

The destination location for files copied or moved from the above source location. You can use an edit mask to translate source filenames into destination filenames. For more information, refer to *Edit Masks* at the beginning of Chapter 1, "Commands."

When performing the step, any elements of the destination location which the user omits are filled–in using step values defined here.

Additionally, you can use the following LIBRARIAN wildcards in elements of the Destination Location:

!OWNER	Owner of the source file. Owner is the user who created the file. This is useful when you need to return files to the programmers who submitted them to a central location.
!USERID	The ID of the user performing the step.
!LOGON	Current login value.
!MSUSER	User who originally checked out the file.
=	Uses corresponding master name element.
!hpvar	Any MPE system variable that is prefixed by a !, e.g., !HPHGROUP. The value is determined when a user performs the step.

To enter a destination location longer than 64 characters, press **F4** (LONG PATHNAME). The Long Pathname (LP) screen appears to let you specify a name with a maximum of 255 characters. After pressing **F4**, the Source Location field appears on the LP screen. To go to the Destination Location field, press **F1** (TOGGLE) on the LP screen. For more information, refer to the LP screen.

To configure or view additional step options and step parameters, press **F7** (NEXT FKEYS) followed by **F2** (STEP OPTIONS). The Step Options (STO) screen appears.

Prestep
Optional. Length 12.

The name of a step that must be performed before this step. The prestep can be the preceding step in the route, or it can be the name of a composite prestep, as defined on the Composite Presteps (CP) screen.

STEPS (continued)

Explanation of Fields, continued

Alternate Presteps
Optional. Length 2 fields of 12 characters.

The name of one or two alternate steps to satisfy the prestep requirement for this step. If an alternate prestep is defined, the prestep requirement is satisfied when any of the presteps have been performed successfully.

Desc
Optional. Length 72.

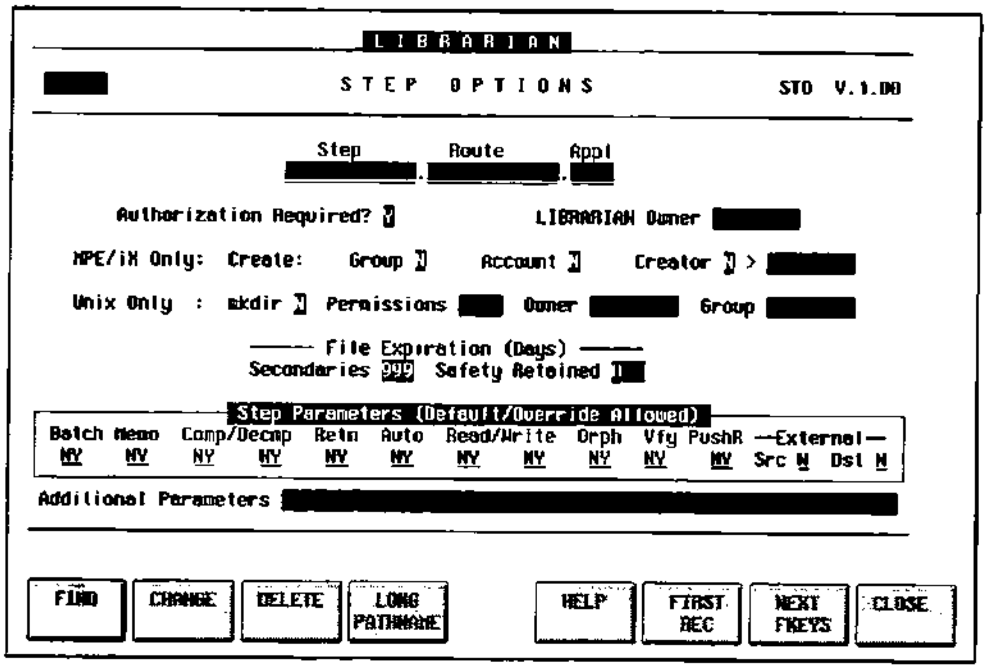
A description of the step.

STEP OPTIONS STO

The Step Options (STO) screen allows you to provide additional step options.

Menu Access	Not available directly from menu mode.
Screen Security	LIBRARIAN Manager, Rule Administrator, and Application Manager. Application Managers can only access the records in their own applications. The only way to access this screen is from the ST screen.
Files Impacted	D-STEP D-SYSTEM-PROFILE

Screen Layout



```
                              L I B R A R I A N

                          S T E P   O P T I O N S                 STO  V.1.00

                                 Step      Route     Appl
                                     .          .

        Authorization Required? N                LIBRARIAN Owner

     MPE/iX Only:  Create:    Group N    Account N    Creator N >

     Unix Only   :  mkdir N  Permissions      Owner          Group

                       ——— File Expiration (Days) ———
                       Secondaries 999  Safety Retained 0

                  Step Parameters (Default/Override Allowed)
   Batch Memo  Comp/Decmp  Retn  Auto  Read/Write  Orph  Vfy PushR —External—
    NY    NY    NY    NY    NY    NY    NY    NY    NY    NY    NY  Src N  Dst N

  Additional Parameters
```

Key Fields

Step, Route, Appl

Description

Use this screen to define additional step options from the Steps (ST) screen.

When a user requests a step that exists in more than one route and/or application, an "ambiguous step name" message is issued, and a menu of steps is displayed. This menu of steps is alphabetically sorted and only displays steps the user is authorized to use.

Explanation of Fields

Step
Required. Length 12.

The name of the step being defined.

Explanation of Fields, continued

Route
Required. Length 12.

The route to which the step belongs.

Appl
Required Length 4.

The application to which the step belongs.

Authorization Required?
Required. Length 1.

A flag indicating whether or not step authorization is required via the Step Authorization (SA) screen. Legal values are:

Y	Indicates that step authorization or a special capability is required (default).
N	Indicates that any user defined to LIBRARIAN can perform this step.

LIBRARIAN Owner
Required. Length 8.

The LIBRARIAN owner that should be assigned to files created by this step. The default is the user who performs the step.

MPE **Create: Group**
Required. Length 1.

A flag indicating if a new group should be created, if one does not exist in the destination account. Legal values are:

Y	Creates a group if one does not exist.
N	Does not create a group if one does not exist. If no group exists, the file movement will fail.

MPE **Create: Account**
Required. Length 1.

A flag indicating whether to create a new account, if one does not exist for the destination file. Legal values are:

Y	Creates an account if one does not exist.
N	Does not create an account if one does not exist. If no account exists, the file movement will fail.

Explanation of Fields, continued

Create: Creator
Required. Length 1.

A flag indicating whether to create a new MPE user, if the creator for the destination file does not exist. Legal values are:

Y	Creates a user if creator does not exist in the destination account.
N	Does not create a user if one does not exist.

⇒Creator
Optional. Length 8.

Assigns the specified user as the creator of the destination file. This field is especially useful when pushing files across account boundaries. The user does need not to exist in the destination account if the Create: Creator flag is set to Y.

You can use the following wildcards:

!USERID	Use the LIBRARIAN user as the creator.
!LOGON	Use the current MPE login user as the creator.
!KEEP	Use the creator of the file being replaced.

Mkdir
Required. Length 1.

A flag indicating if LIBRARIAN should create a new directory for the destination file if it does not already exist. Legal values are:

Y	Creates directory if it doesn't exist.
N	Does not create directory if it does not exist. If the directory does not exist, the file movement will fail.

Permissions
Optional. Length 3.

Assigns permissions to the files this step creates. (For more information, see the UNIX **chmod** command.)

Owner
Optional. Length 5.

Assigns the specified user as the owner of the files this step creates. It can be a specific UNIX user login or decimal user ID found in /etc/passwd.

Group
Optional. Length 5.

Allows you to specify the group for files this step creates. It can be a specific UNIX group name or decimal group ID found in /etc/group.

File Expiration (Days)

The expiration criteria for files this step creates. Includes separate values for read mode secondary files and retained files.

Explanation of Fields, continued

Secondaries
Optional. Length 3.

The expiration period (in days) for read mode secondary files this step creates. Possible values range from 0 to 999. A value of 0 indicates that the file expires on the same date it is created. A value of 999 indicates that the file does not expire. When a secondary file has expired, it can be flushed.

Safety Retained
Optional. Length 3.

The expiration period (in days) for files this step retains. Possible values range from 0 to 999. A value of 0 indicates that the file expires on the same date it is created. A value of 999 indicates that the file does not expire. When a file has expired, it can be flushed.

Step Parameters (Default/Override Allowed)

These flags define the default parameters when a user performs the step. They also indicate whether a user performing this step can override the defined default parameter value. See the **PERFORM** command for more information on parameters available when executing step commands.

Batch
Required. Length 2 fields of 1.

Default (first position)

A flag to indicate if the transaction should be performed in batch mode. Legal values are:

Y	Performs transaction in batch mode. If the EXPRESS Enabled flag in the System Profile (SP) screen is set to Y, the user is prompted to schedule the transaction (EXPRESS must be installed). If EXPRESS is not enabled, the transaction will be streamed directly through MPE.
N	Performs the transaction online (default).

Override (second position)

A flag to indicate if you can override the default **BATCH** value. Legal values are:

Y	Indicates that you can override the default **BATCH** value (default).
N	Indicates that all transactions must use the default **BATCH** value.

Memo
Required. Length 2 fields of 1.

Default (first position)

A flag to indicate if memo text is required when performing this step. Legal values are:

Y	Indicates that memo text is required. The program will prompt for text describing each transaction.
N	Indicates that memo text is not required (default).

Explanation of Fields, continued

Override (second position)

A flag to indicate whether you can override the default **MEMO** value. Legal values are:

Y	Indicates that you can override the default **MEMO** value (default).
N	Indicates that all transactions must use the default **MEMO** value.

Comp (Compress)
Required. Length 2 fields of 1.

Default (first position)

A flag to indicate if this step should automatically compress the new destination file. Legal values are:

Y	Indicates that LIBRARIAN should compress the destination file when performing this step. Specific types of files in the Compress Exclusions (CE) screen will not be compressed. If this flag is set to Y, DECMP must be set to N.
N	Indicates that LIBRARIAN should not compress destination files (default).

Override (second position)

A flag to indicate if you can override the default **COMPRESS** value. Legal values are:

Y	Indicates that you can override the default **COMPRESS** value (default).
N	Indicates that all transactions must use the default **COMPRESS** value.

Decmp (Decompress)
Required. Length 2 fields of 1.

Default (first position)

A flag to indicate if this step should decompress the new destination file which is created by this transaction automatically. Legal values are:

Y	Indicates that LIBRARIAN should decompress the destination file when performing this step.
N	Indicates that LIBRARIAN should not decompress the destination file (default).

Override (second position)

A flag to indicate if you can override the default **DECOMPRESS** value. Legal values are:

Y	Indicates that you can override the default **DECOMPRESS** value (default).
N	Indicates that all transactions must use the default **DECOMPRESS** value.

Explanation of Fields, continued

Retn (Retain)
Required. Length 2 fields of 1.

Default (first position)

A flag to indicate whether to retain destination files automatically. The files would otherwise be replaced by this step. Legal values are:

Y	Indicates that LIBRARIAN should retain files.
N	Indicates that LIBRARIAN should not retain files (default).

Override (second position)

A flag to indicate if you can override the default **RETAIN** value. Legal values are:

Y	Indicates that you can override the default **RETAIN** value (default).
N	All transactions must use the default **RETAIN** value.

Auto (Autoupdate)
Required. Length 2 fields of 1.

Default (first position)

A flag to indicate whether to automatically update filesets (Auto Fileset Update) with new files (for secondary-to-master steps only). Legal values are:

Y	Indicates that LIBRARIAN should update the fileset after the transaction. Destination files in the master location are compared to Auto Fileset descriptors in the Auto Filesets (AF) screen. New files that are not previously defined will be added to the appropriate filesets.
N	Indicates that LIBRARIAN should not update the filesets automatically (default).

Override (second position)

A flag to indicate if you can override the default **AUTOUPDATE** value. Legal values are:

Y	Indicates that you can override the default **AUTOUPDATE** value (default).
N	Indicates that all transactions must use the default **AUTOUPDATE** value.

Read
Required. Length 2 fields of 1.

Default (first position)

A flag to indicate whether to assign read mode access to files created with this step. Legal values are:

Y	Indicates that LIBRARIAN should automatically assign read mode access.
N	Indicates that LIBRARIAN should not automatically assign read mode access (default).

Explanation of Fields, continued

Override (second position)

A flag to indicate if you can override the default **READ** parameter value. Legal values are:

Y	Indicates that you can override the default **READ** value (default).
N	Indicates that all transactions must use the default **READ** value.

Write
Required. Length 2 fields of 1.

Default (first position)

A flag to indicate whether to automatically assign write mode access to files created with this step. Legal values are:

Y	Indicates that LIBRARIAN should automatically assign write mode access.
N	Indicates that LIBRARIAN should not automatically assign write mode access (default).

Override (second position)

A flag to indicate if you can override the default **WRITE** parameter value. Possible values are:

Y	Indicates that you can override the default **WRITE** value (default).
N	Indicates that all transactions must use the default **WRITE** value.

Orph (Orphan)
Required. Length 2 fields of 1.

Default (first position)

A flag to indicate whether to break the link between a master file and its secondaries automatically when performing this step. Orphaned secondaries are not tracked by LIBRARIAN after making the copy. Legal values are:

Y	Indicates that LIBRARIAN should orphan secondaries automatically.
N	Indicates that LIBRARIAN should not orphan secondaries automatically (default).

Override (second position)

A flag to indicate if you can override the default **ORPHAN** parameter value. Legal values are:

Y	Indicates that you can override the default **ORPHAN** value (default).
N	Indicates that all transactions must use the default **ORPHAN** value.

Explanation of Fields, continued

Vfy (Verify)
Required. Length 2 fields of 1.

Default (first position)

A flag indicating if LIBRARIAN should verify that files have not been modified since they were created by LIBRARIAN.

Y Indicates that LIBRARIAN should verify files automatically.

N Indicates that LIBRARIAN should not verify files automatically (default).

Override (second position)

A flag to indicate if you can override the default **VERIFY** parameter value. Legal values are:

Y Indicates that you can override the default **VERIFY** value (default).

N Indicates that all transactions must use the default **VERIFY** value.

PushR (Pushread)
Required. Length 2 fields of 1.

Default (first position)

A flag indicating if files in read mode should be allowed to replace a write mode secondary or a master file. When this is done, an exception flag is set. Legal values are:

Y Indicates that LIBRARIAN should allow read mode files to replace write mode files.

N Indicates that LIBRARIAN should not allow read mode files to replace write mode files (default).

Override (second position)

A flag to indicate if you can override the default **PUSHREAD** parameter value. Legal values are:

Y Indicates that you can override the default **PUSHREAD** value (default).

N Indicates that all transactions must use the default **PUSHREAD** value.

External
Required. Length 2 fields of 1.

A flag indicating whether to just record the movement for files that are located on systems inaccessible to LIBRARIAN. Users are responsible for transferring files using some other method than LIBRARIAN.

Src (Source)

A flag indicating whether source files should be recorded as moved, without LIBRARIAN physically moving or copying the files. It is used when files reside on a system not running LIBRARIAN.

Explanation of Fields, continued

Dst (Destination)

A flag indicating whether the destination files should be recorded as moved without LIBRARIAN physically moving or copying the files. It is used when files reside on a system not running LIBRARIAN.

Additional Parameters
Optional. Length 24.

A free-form set of parameters that are appended to the user request when performing the step. Use the syntax described for the parameters of the **PERFORM** command (e.g., **NOINPROGRESS, NOTIFY**) as described in Chapter 1, "Commands". Any conflict with user–supplied parameters, or syntax errors are reported at the time the step is performed.

Operation

You can access the Step Options (STO) screen only from the Steps (ST) screen. To do so, press **F7** (NEXT FKEYS) followed by **F2** (STEP OPTIONS). The Step Options (STO) screen appears.

Note Pressing **F8** (CLOSE) from the Step Options screen returns you to the Steps (ST) screen.

SYSTEMS SY

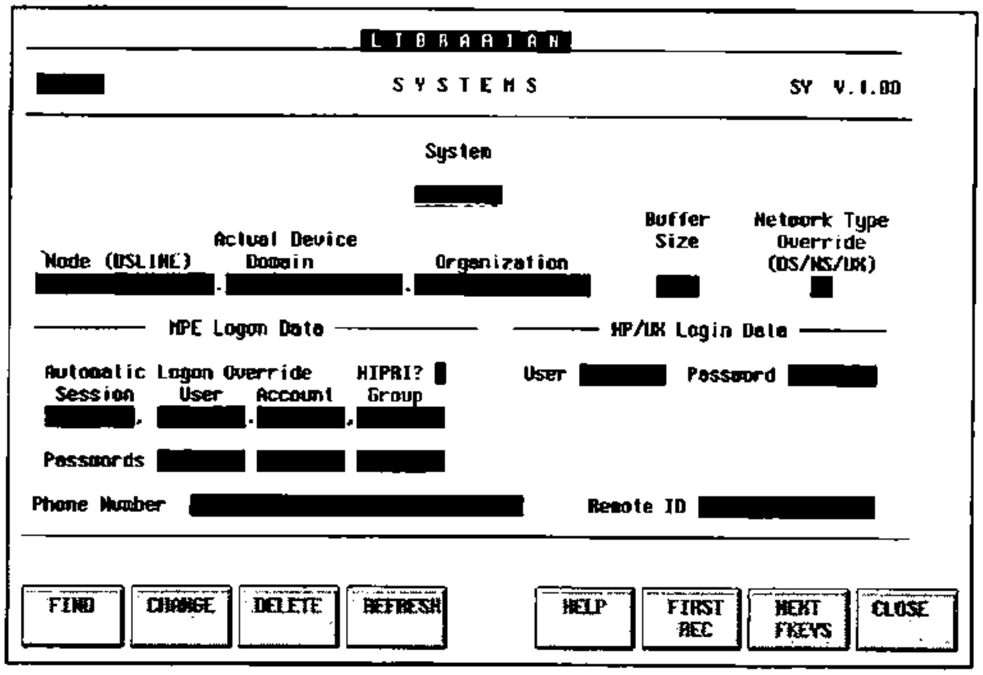
The Systems (SY) screen is where you define information about how LIBRARIAN should connect to systems in your network.

Menu Access Admin...Screens...Config...SY Systems

Screen Security LIBRARIAN Manager

Files Impacted D-SYSTEM-ID

Screen Layout



Key Fields

None

Description

Use this screen to define information about how to connect to a system, if it is different from the global definitions on the Network Configuration (NC) screen, or if the system is a UNIX machine.

Description, continued

Use this screen for any of the following MPE systems:

- LIBRARIAN's logical system name in the system configuration file (CONFIG.PUB.OCSLIB) is not the same as the actual MPE node name.
- The MPE system does not operate under the network values described on the Network Configuration (NC) screen.
- LIBRARIAN should be accessed by an automatic remote login different from the one shown on the Network Configuration (NC) screen.
- The MPE system has different passwords for the automatic remote login.
- The MPE system is accessed via a dial-DS link.

UNIX When the MPE server installs a UNIX client, it automatically creates a system record.

If a system is accessed differently by different systems, use the System-to-System Table (SS) screen to specify the node name to be used.

Explanation of Fields

System
Required. Length 8.

The system as it appears in the LIBRARIAN configuration file.

Actual Device

The actual DS/NS device name used to access this system. Domain and organization are used only for NS networks with multiple domain and organization values. Use these fields if the name in the system configuration file does not adequately define the device. It consists of the following subfields:

Node (DSLINE)
Optional. Length 16.

The actual device name for this system, if different from the LIBRARIAN configuration file.

Domain
Optional. Length 16.

The domain associated with the node (NS only).

Organization
Optional. Length 16.

The organization associated with the node (NS only).

Note You can use embeded periods with all **Node, Domain,** and **Organization** fields to handle network addresses that contain more than three parts.

Explanation of Fields, continued

Note If you have network node names with more than three elements, you can put more than one element in any of the Node, Domain, and Organization fields of the SY screen. For example, to enter the node name *abc.chicago.usa.com* on the SY screen, you can enter *abc* as the Node, *chicago.usa* as the Domain, and *com* as the Organization.

Buffer Size
Optional. Length 4.

The network communications buffer size for this system, if different from the value on the Network Configuration (NC) screen.

Network Override
Optional. Length 2.

The network type for this system, if different from the value on the Network Configuration (NC) screen. (NS/DS/UX)

MPE Logon Data

Automatic Logon Override

The login for remote transactions on this system, if different from the Network Configuration (NC) screen. If used, the value overrides the automatic login for this system only.

The special **!USERID** wildcard can be used in the Session field, so that automatic remote login reflects the LIBRARIAN user.

This field consists of the following subfields:

Session
Optional. Length 8.

The default session name for automatic remote login.

User
Required. Length 8.

The default user name for automatic remote login.

Account
Required. Length 8.

The default account name for automatic remote login.

Group
Required. Length 8.

The default group name for automatic remote login.

Explanation of Fields, continued

Passwords
Optional. Length 3 fields of 8.

The user, account, and/or group passwords for the automatic login.

MPE **HIPRI?**
Required. Length 1.

A flag indicating whether LIBRARIAN uses HIPRI to log on to this system. Legal values are:

Y Indicates HIPRI is used to login to the system.

N Indicates HIPRI is not used to login to the system.

Note In order to use **HIPRI** for automatic remote logon to LIBRARIAN, MPE requires that the logon user and account have OP or SM capability.

MPE **Phone Number**
Optional. Length 30.

The phone number used by the modem to establish a telephone connection. (Dial DS)

MPE **Remote ID**
Optional. Length 16.

A string that defines the valid ID sequence when attempting to establish a telephone connection. If the remote system does not send a valid ID sequence, the telephone connection is terminated (Dial DS).

UNIX **HP/UX Login Data**

User
Required. Length 8.

The default user name for automatic remote login.

Password
Optional. Length 8.

The user password for the remote login.

Operation

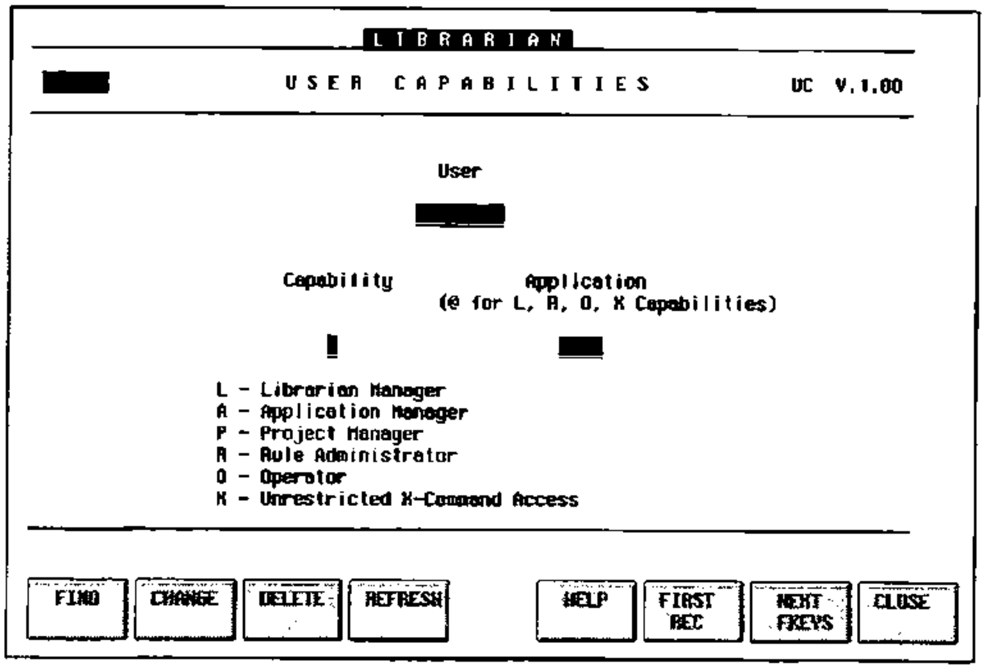
If you specify a login ID override, you can use the special !USERID wildcard for the session. The user's current user will be substituted as the session in the remote login.

USER CAPABILITIES UC

The User Capabilities (UC) screen allows you to provide users different levels of access.

Menu Access	Admin...Screens...Users...UC User Capabilities
Screen Security	LIBRARIAN Manager, and Application Manager. Application Managers cannot assign L, X, R, or O capability.
Files Impacted	D-USER-CAPS

Screen Layout



```
                         L I B R A R I A N
        U S E R   C A P A B I L I T I E S             UC  V.1.00

                              User

              Capability             Application
                               (@ for L, R, O, X Capabilities)

        L - Librarian Manager
        A - Application Manager
        P - Project Manager
        R - Rule Administrator
        O - Operator
        X - Unrestricted X-Command Access
```

Key Fields

User

Description

Use this screen to provide selected users with LIBRARIAN Manager, Application Manager, Project Manager, Rule Administrator, or Operator Capabilities. In addition, unrestricted use of the X commands can be granted on this screen.

UC records are not required for general LIBRARIAN users.

Explanation of Fields

User
Required. Length 8.

The unique identifier of the LIBRARIAN user. You must have previously defined the user on the Users (US) screen.

Capability
Required. Length 1.

A flag indicating a special capability for this user.

Application
Required. Length 4.

The name of the application for which the user has special capability, as defined in the Applications (AP) screen.

If the capability is L, X, R, or O, use the at sign (@) in this field to indicate all applications.

Application managers can only define users to have Project Manager capability.

Operation

LIBRARIAN Managers define libraries and have unrestricted access to all LIBRARIAN functions.

Project Managers can create new projects on the Projects (PJ) screen and update the status of their projects in the Project Status Change (PS) screen. Additionally, they can authorize users for their projects in the Project Authorization (PA) screen.

Additionally, these users are automatically authorized to work on any projects they manage. The users can manipulate related project filesets by using the FMAINT commands. Project Manager appears on standard project reports.

Rule Administrators can define libraries and file movement rules, but cannot perform LIBRARIAN file operations without authorization from a LIBRARIAN manager.

Operators can run the FLUSH and FLUSHLOG utilities. They can execute the LIBRARIAN **RESTORE** command for master files. Operator capability is not application-specific.

The *X–Command access* capability allows a user to perform commands on untracked files without enforcement of MPE security (see the X commands in Chapter 1, "Commands."

USERS US

The Users (US) screen is where you add or change user records.

Menu Access

Admin...Screens...Users...US Users

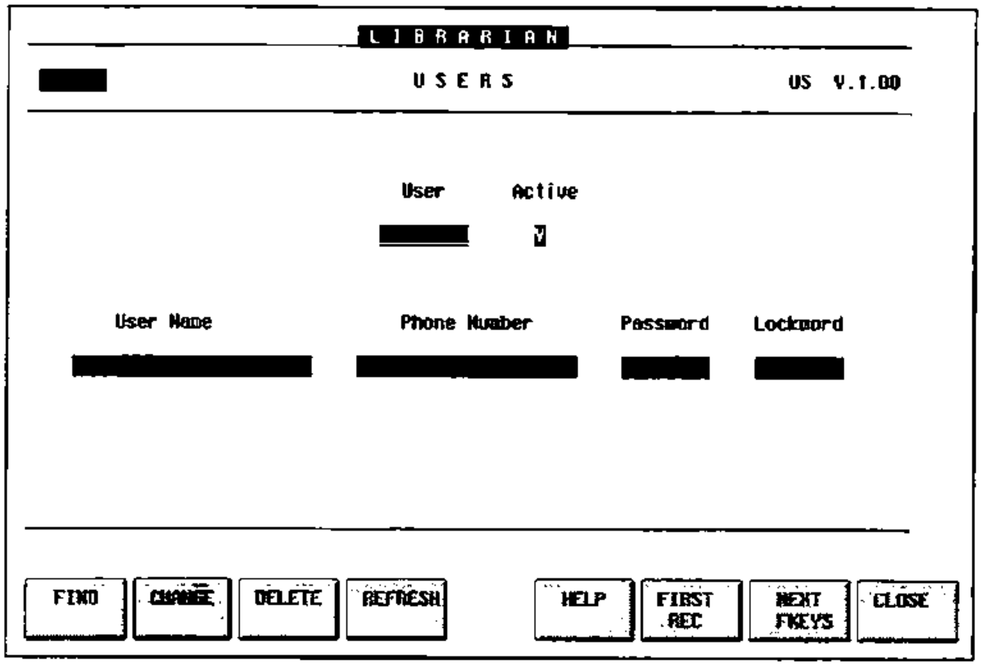
Screen Security

Any user. General system users and Application Managers can only access their own records. The LIBRARIAN Manager can access all records, and is the only person who has the capability to add new users. No one, including the LIBRARIAN Manager, can view the passwords of other users.

Files Impacted

M-USER

Screen Layout



Key Fields

User

Description

General system users can access this screen to add or change the data of their own records. They can update the information, but they cannot access another user's record. No one, not even a LIBRARIAN manager, is allowed to view another user's password.

The Active flag on this screen determines whether a user currently has access to LIBRARIAN. A LIBRARIAN manager can set the flag at any time to change the user's access to LIBRARIAN without deleting the record from the database.

Passwords can be changed on this screen by using the **USER** command, or from the **User** menu.

Explanation of Fields, continued

User
Required. Length 8.

The identifier of the LIBRARIAN user described in this record. The user can include alphabetic, numeric, hyphen (-), and underscore (_) characters.

Active
Required. Length 1.

A flag that indicates this user is currently an active user.

Legal values are:

- Y Indicates that the user is active. The user currently has access to the LIBRARIAN system (default).
- N Indicates that the user is inactive. The user does not currently have access to the LIBRARIAN system.

User Name
Optional. Length 22.

The full name of the user. Use this field for documentation only.

Phone Number
Optional. Length 20.

The phone number of the user. Use this field for documentation only.

Password
Required. Length 8.

The password for the user. This password is selected by the user and must be supplied each time this user accesses the LIBRARIAN system. Passwords are encrypted in the database and are only displayed on this screen when you press **F2** (**Set 2**).

MPE **Lockword**
Optional. Length 8.

The lockword for the user. LIBRARIAN assigns this lockword automatically to all secondary files created by this user. Lockwords are encrypted in the database and are only displayed on this screen when you press **F2** (**Set 2**).

Operation

When you access this screen, the password and lockword fields are not displayed. You can display them with the SHOW PASSWORD key. To hide them again, press HIDE PASSWORD. The SHOW PASSWORD and HIDE PASSWORD functions are turned on and off by **F2** (**Set** 2).

Note Additionally, you can change passwords and lockwords by using the **USER** command in LIBRARIAN, or the **Passwords** option of the **User** menu.

The DELETE (F3) key on the US screen invokes a mass delete of all associated data, including:

- step authorizations (SA)
- project authorizations (PA)
- user capabilities (UC)
- user fileset ownership
- messages queued to this user

The user is prompted to confirm the mass delete by pressing F3 a second time.

Reports 6

This chapter describes and provides samples of all LIBRARIAN reports. The following topics are discussed in this chapter:

- Summary of LIBRARIAN Reports
- Transaction Codes on Reports
- Generating Reports
- Redirecting Reports
- Report Headings
- Report Descriptions and Samples

Summary of LIBRARIAN Reports

Table 6–1 on the next page lists the LIBRARIAN reports categorized by type. The table lists the five-character report codes used to generate the reports. Additionally, page references help you locate detailed descriptions and samples for all reports.

Table 6–1. LIBRARIAN Reports by Type

Report Code	Report Title	Page
Step/Project Reports		
RAD10	Step Summary	6–8
RAD20	Step Detail	6–10
RAV10	Versions	6–14
RPJ10	Projects	6–33
RUP10	Project Authorizations	6–47
RUS10	Step Authorizations	6–48
Fileset and File Reports		
RAF10	Auto Filesets	6–13
RFD10	Fileset Status	6–17
RFD20	Master File Status	6–19
RFE10	Fileset Explosion	6–21
RFE20	Fileset Explosion (with descriptions)	6–23
RFX10	File Exceptions	6–29
RGF10	Generated Files	6–31
RPM10	Pending Master Files	6–35
RRH10	Revision History	6–37
RVD10	File Versions	5–85
RVT10	File Versions and Timestamps	6–52
RVT20	File Versions and Timestamp Exceptions	6–54
Transaction Log Reports		
RTD10	Transaction Detail (by date, time)	6–39
RTD40	Transaction Detail (by file, date, time)	6–41
RTS10	Transaction Summary (by date, time)	6–43
User Data Reports		
RUD10	Users	6–45
RUP10	Project Authorizations	6–47
RUS10	Step Authorizations	6–48
Flush Reports		
FLUSH	Flush Utility Detail	6–7
RFN10	Pre-Flush Notification (by files)	6–25
RFN20	Pre-Flush Notification (by user)	6–27

The File Inquiry (FI) screen and the **VERIFY** and **SHOWLOG** commands also function as online reports.

- The FI screen displays the status of any file used by LIBRARIAN.
- The **VERIFY** command displays/prints information about files and their versions.
- The **SHOWLOG** commands display/print transaction history data. For more information on SHOWLOG commands, refer to Chapter 4, "SHOWLOG Commands."

Transaction Codes on Reports

Table 6–2 lists the transaction codes that appear on reports generated from the LIBLOG database (i.e., RTD10, RTD40, RTS10, and SHOWLOG reports). Table 6–3 lists transaction subcategory codes for the **PERFORM, SET,** and **RESET** commands.

Table 6–2. Transaction Report Codes

Transaction Code	Command	Transaction Code	Command
CO	COMPRESS	SC	SECURE
CP	COPY	SR	SCAN with REPLACE
DC	DECOMPRESS	SE	SET
LO	LOCK	TO	TOUCH
ME	MEMO	UL	UNLOCK
MV	MOVE	UP	UPDATE
OR	ORPHAN	XC	XCOMPRESS
OV	OVERLAY	XD	XDECOMPRESS
PF	PERFORM	XM	XMOVE
PU	PURGE	XP	XPURGE
RE	RESET	XR	XRENAME
RL	RELEASE	XT	XTOUCH
RN	RENAME	XY	XCOPY
RS	RESTORE		

Table 6-3. LIBLOG Transaction Subcategory Codes

Subcode (or S C) is the subcategory of a transaction		
For **PERFORM**	C	Copy
	M	Move
	N	Null
	G	Generated (retained file)
For **SET**	M	Mode (Read/Write)
	L	Lockword
	E	Expiration Date
	O	Owner
For **RESET**	X	Exception
	T	Timestamp

Generating Reports

There are three ways to generate LIBRARIAN reports:

1. In menu mode, select **Info** from the main menu. Then, select the **Files...**, **Versions...**, **Rules...**, or **Log...** menu and choose the report you want to run.
2. In command mode, type the report code at the LIBRARIAN prompt and press RETURN. For example, generate the Transaction Summary Report (RTS10) by typing:

```
>RTS10
```

3. You can also stream the appropriate report jobstream (LIBJ*xxxx*.JOB.OCSLIB). For example, obtain the Transaction Summary Report (RTS10) by typing:

```
:STREAM LIBRTS10.JOB.OCSLIB
```

If you are using the EXPRESS or PRIVATE products on your system, you can use **STREAMER** to generate reports.

Redirecting Reports

Reports run from a UNIX client are generated on the MPE server. Outputs are sent to the default printer for the client using the UNIX **lp** command. You can set the environment variable LIBPRINT on a UNIX client to specify options for the **lp** command. For example, to print all the reports on a printer called "ocsprinter", you would put the following command in the profile file.

```
LIBPRINT = "-d ocsprinter"
export LIBPRINT
```

Offline reports are directed to the default LP device; however, you can redirect the output destination, if desired. Additionally, the default priority is assigned to LIBRARIAN reports, unless you override it.

There are two **FILE** commands available to override the output destination and priority:

1. To redirect all LIBRARIAN reports, issue a file equation for the formal file designator LIBOUT. For example,

   ```
   FILE LIBOUT;DEV=36,2
   ```

   The **FILE** command above directs every report to the defined device (in this case 36), with the defined output priority (in this case 2). The formal file designator for all files will be LIBOUT.

2. To redirect a specific LIBRARIAN report, issue a file equation for a formal file designator in the format LIBR??##, where *??##* corresponds to a specific report code. For example,

   ```
   FILE LIBRTS10;DEV=WIDELP,3
   ```

   The **FILE** command above directs the specified report (in this case, LIBRTS10) to the defined device (in this case, WIDELP), with the defined output priority (in this case, 3). This file equation overrides the file equation issued in the first example for this report.

Report Headings

The heading for each LIBRARIAN report contains the following information:

Report	The LIBRARIAN report code. It is also the formal file designator for the report.
Version	The version number of the program that generated the report.
System	The identifier of the system on which this report was generated. **System** is stored in the LIBRARIAN system configuration file.
Sort Sequence	The sequence in which report information is sorted.
Report Title	The LIBRARIAN report title.
Company Name	Your company name. The company name for all reports is taken from licence information entered during product authorization.
Page	The page number within the report. All reports begin with page number one.
Printed	The date the report was generated.
Time	The time the report was generated.

Report Descriptions and Samples

The following pages contain detailed descriptions of the LIBRARIAN reports, in alphabetical order by report code. Each report includes the following information:

Report Name and Code	Report title and five-character report code.
Sort Sequence	Sequence in which report information is sorted.
Files Accessed	Datasets/files accessed for the report.
When to Run	Recommended run frequency for the report.
Menu Access	The menu from which you can generate the report.
Description	Description of the information included in the report.
Sample	A sample of the report.
Field Descriptions	Description of the fields on the report.

FLUSH DETAIL REPORT

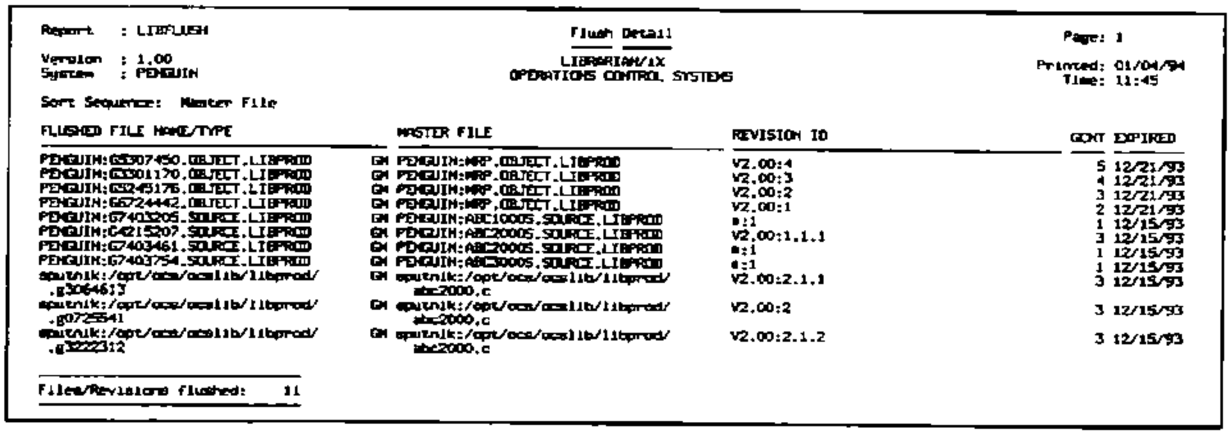
Sort Sequence	Master File
Files Accessed	D-APPL-VERSION D-FILE
When to Run	Automatically generated when running the FLUSH utility
Menu Access	**Admin...Flush**

Description

The FLUSH utility generates this report to identify all secondary files and revisions it purged. This report cross-references retained files with their master files. This enables you to determine which retained files were purged.

Report information includes the retained file name, the master file with which it is associated, revision identifier, and expiration date.

Sample



```
Report   : LIBFLUSH                              Flush Detail                                          Page: 1
                                                 ----- ------
Version  : 1.00                                  LIBRARIAN/iX                                   Printed: 01/04/94
System   : PENGUIN                         OPERATIONS CONTROL SYSTEMS                              Time: 11:45

Sort Sequence:  Master File

FLUSHED FILE NAME/TYPE                  MASTER FILE                          REVISION ID                GCNT EXPIRED
----------------------------------      ----------------------------------   ---------------------     ---- --------
PENGUIN:G5307450.OBJECT.LIBPROD         GM PENGUIN:MRP.OBJECT.LIBPROD        V2.00:4                      5 12/21/93
PENGUIN:G3301170.OBJECT.LIBPROD         GM PENGUIN:MRP.OBJECT.LIBPROD        V2.00:3                      4 12/21/93
PENGUIN:G5245176.OBJECT.LIBPROD         GM PENGUIN:MRP.OBJECT.LIBPROD        V2.00:2                      3 12/21/93
PENGUIN:G6724442.OBJECT.LIBPROD         GM PENGUIN:MRP.OBJECT.LIBPROD        V2.00:1                      2 12/21/93
PENGUIN:G7403205.SOURCE.LIBPROD         GM PENGUIN:ABC1000S.SOURCE.LIBPROD   *:1                          1 12/15/93
PENGUIN:G4215207.SOURCE.LIBPROD         GM PENGUIN:ABC2000S.SOURCE.LIBPROD   V2.00:1.1.1                  3 12/15/93
PENGUIN:G7403461.SOURCE.LIBPROD         GM PENGUIN:ABC2000S.SOURCE.LIBPROD   *:1                          1 12/15/93
PENGUIN:G7403754.SOURCE.LIBPROD         GM PENGUIN:ABC3000S.SOURCE.LIBPROD   *:1                          1 12/15/93
sputnik:/opt/ocs/ocslib/libprod/        GM sputnik:/opt/ocs/ocslib/libprod/  V2.00:2.1.1                  3 12/15/93
 .g3064613                                    abc2000.c
sputnik:/opt/ocs/ocslib/libprod/        GM sputnik:/opt/ocs/ocslib/libprod/  V2.00:2                      3 12/15/93
 .g0725541                                    abc2000.c
sputnik:/opt/ocs/ocslib/libprod/        GM sputnik:/opt/ocs/ocslib/libprod/  V2.00:2.1.2                  3 12/15/93
 .g3222312                                    abc2000.c

--------------------------------
Files/Revisions flushed:      11
--------------------------------
```

Field Descriptions

Flushed File Name	The name of the file flushed.
Type	The file type represented by the following codes:

S	Secondary
GM	Retained Master
GS	Retained Secondary
CM	Copies of Retained Master
CS	Copies of Retained Secondary

Master File	The name of the corresponding master file.
Revision ID	The revision identifier associated with the flushed file.
GCOUNT	The generation count (GCOUNT) for the file.
Expired	The date the flushed file expired.
Files/Revisions flushed	The total number of files flushed.

STEP SUMMARY REPORT RAD10

Sort Sequence

Application, Route, Step

Files Accessed

D-STEP	M-APPLICATION
D-SYSTEM-PROFILE	M-USER
D-USER-CAPS	

When to Run

On demand

Menu Access

Info...Rules...RAD10 Step Summary

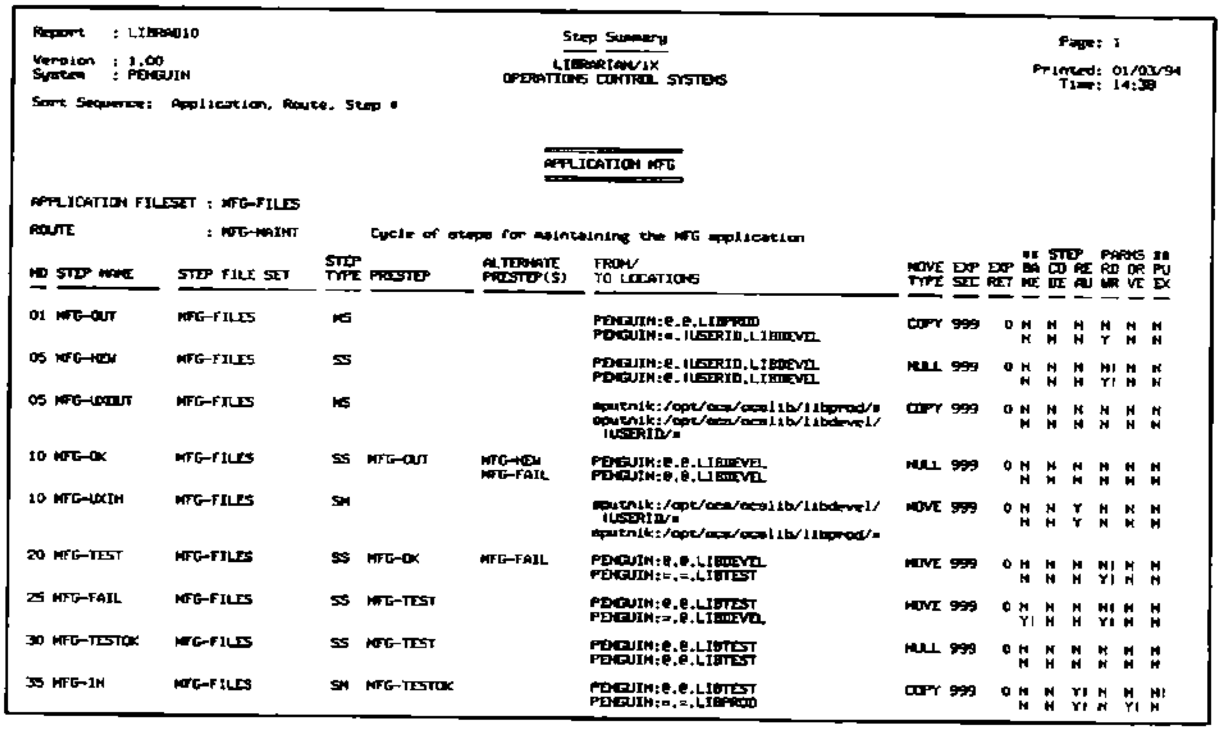
Description

This report provides an overview of the routes and steps in each application.

The summary information for each step in the route includes step type, prestep, alternate presteps, source and destination locations, movement type, file expiration, default parameters, and override restrictions.

In addition, the report provides general application information such as the name of the Application manager and phone number.

Sample



```
Report   : LIBRAD10                       Step Summary                               Page: 1
                                          ---- -------
Version  : 1.00                           LIBRARIAN/iX                           Printed: 01/03/94
System   : PENGUIN                   OPERATIONS CONTROL SYSTEMS                     Time: 14:38

Sort Sequence:  Application, Route, Step #


                                         ----------------
                                         APPLICATION MFG
                                         ----------------

APPLICATION FILESET : MFG-FILES

ROUTE               : MFG-MAINT       Cycle of steps for maintaining the MFG application

                                                                                                       ** STEP   PARMS **
                                    STEP                   ALTERNATE     FROM/                         MOVE EXP EXP BA CD RE RD OR PU
NO STEP NAME       STEP FILE SET    TYPE PRESTEP           PRESTEP(S)    TO LOCATIONS                  TYPE SEC RET ME DE AU WR VE EX
-- ---------       -------------    ---- -------           ----------    ------------                  ---- --- --- -- -- -- -- -- --

01 MFG-OUT         MFG-FILES         MS                                  PENGUIN:@.@.LIBPROD           COPY 999   0 N  N  N  N  N  N
                                                                         PENGUIN:=.!USERID.LIBDEVEL                 N  N  N  Y  N  N

05 MFG-NEW         MFG-FILES         SS                                  PENGUIN:@.!USERID.LIBDEVEL    NULL 999   0 N  N  N  N! N  N
                                                                         PENGUIN:@.!USERID.LIBDEVEL                 N  N  N  Y! N  N

05 MFG-UXOUT       MFG-FILES         MS                                  sputnik:/opt/ocs/ocslib/libprod/*  COPY 999   0 N  N  N  N  N  N
                                                                         sputnik:/opt/ocs/ocslib/libdevel/          N  N  N  N  N  N
                                                                          !USERID/*

10 MFG-OK          MFG-FILES         SS  MFG-OUT           MFG-NEW       PENGUIN:@.@.LIBDEVEL          NULL 999   0 N  N  N  N  N  N
                                                           MFG-FAIL      PENGUIN:@.@.LIBDEVEL                       N  N  N  N  N  N

10 MFG-UXIN        MFG-FILES         SM                                  sputnik:/opt/ocs/ocslib/libdevel/  MOVE 999   0 N  N  Y  N  N  N
                                                                          !USERID/*                                 N  N  Y  N  N  N
                                                                         sputnik:/opt/ocs/ocslib/libprod/*

20 MFG-TEST        MFG-FILES         SS  MFG-OK            MFG-FAIL      PENGUIN:@.@.LIBDEVEL          MOVE 999   0 N  N  N  N! N  N
                                                                         PENGUIN:=.=.LIBTEST                        N  N  N  Y! N  N

25 MFG-FAIL        MFG-FILES         SS  MFG-TEST                        PENGUIN:@.@.LIBTEST           MOVE 999   0 N  N  N  N! N  N
                                                                         PENGUIN:=.@.LIBDEVEL                       Y! N  N  Y! N  N

30 MFG-TESTOK      MFG-FILES         SS  MFG-TEST                        PENGUIN:@.@.LIBTEST           NULL 999   0 N  N  N  N  N  N
                                                                         PENGUIN:@.@.LIBTEST                        N  N  N  N  N  N

35 MFG-IN          MFG-FILES         SM  MFG-TESTOK                      PENGUIN:@.@.LIBTEST           COPY 999   0 N  N  Y! N  N  N!
                                                                         PENGUIN:=.=.LIBPROD                        N  N  Y! N  Y! N
```

Field Descriptions

Application Manager	Identifier for Application Manager.
Name	Name of Application Manager.
Phone	Phone number of Application Manager.
Application Fileset	Name of the fileset for the application.
Route	Name of the route.
No	Step number.
Step Name	Name of step.
Step File Set	Fileset associated with the step.
Step Type	Type of step:

MS	Master–to–secondary
SS	Secondary–to–secondary
SM	Secondary–to–master

Prestep	Name of the prestep(s), if any.
Alternate Prestep(s)	Name of the alternate prestep(s), if any.
From/To Locations	Source and destination locations for the step.
Move Type	Type of movement (copy, move, or null step).
Exp Sec	Number of days before read–mode secondaries expire.
Exp Ret	Number of days before safety–retained files expire.
Step Parms	For each parameter listed below, specifies whether setting is the default (Y/N), and whether overrides are prohibited (!). Refer to the **PERFORM** command in Chapter 1, "LIBRARIAN Commands", for more information about these parameters:

BA (BATCH)	**BATCH** default/override.
ME (MEMO)	**MEMO** default/override.
CO (COMPRESS)	**COMPRESS** default/override.
DE (DECOMPRESS)	**DECOMPRESS** default/ override.
RE (RETAIN)	**RETAIN** default/override.
AU (AUTOUPDATE)	**AUTOUPDATE** default/override.
RD (READ)	**READ** default/override.
WR (WRITE)	**WRITE** default/override.
OR (ORPHAN)	**ORPHAN** default/override.
VE (VERIFY)	**VERIFY** default/override.
PU (PUSHREAD)	**PUSHREAD** default/override.
EX (EXTERNAL)	**EXTERNAL** default/override.

STEP DETAIL REPORT RAD20

Sort Sequence Application, Route, Step

Files Accessed

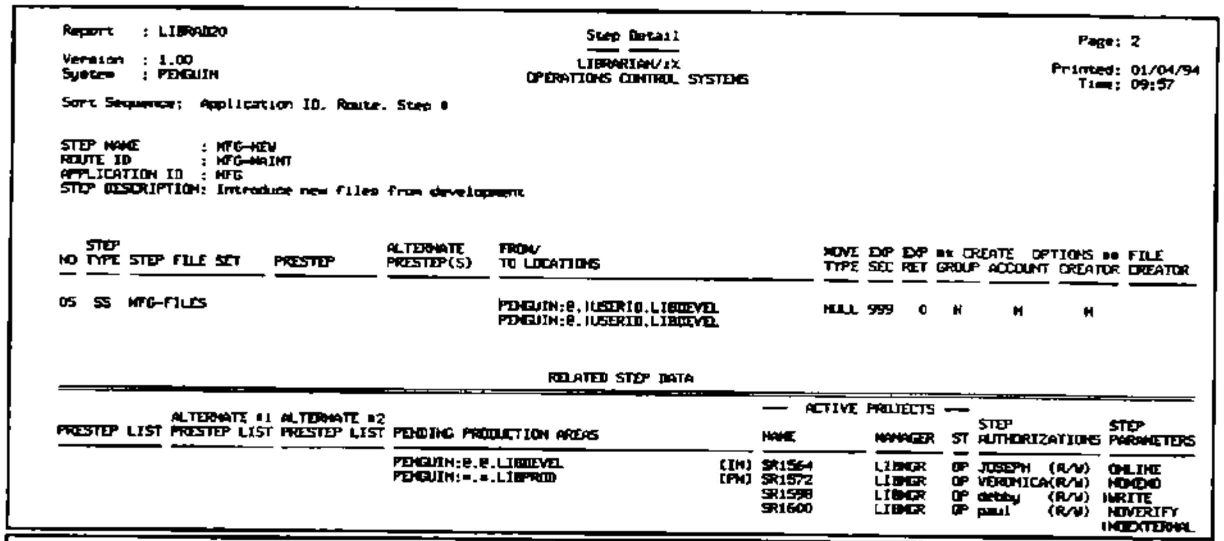
D-PENDING-AREA	D-REFINED-STEP	D-USER-CAPS
D-PRESTEPS	D-STEP	D-USER-STEP
D-PROJECTS	D-SYSTEM-PROFILE	

When to Run On demand

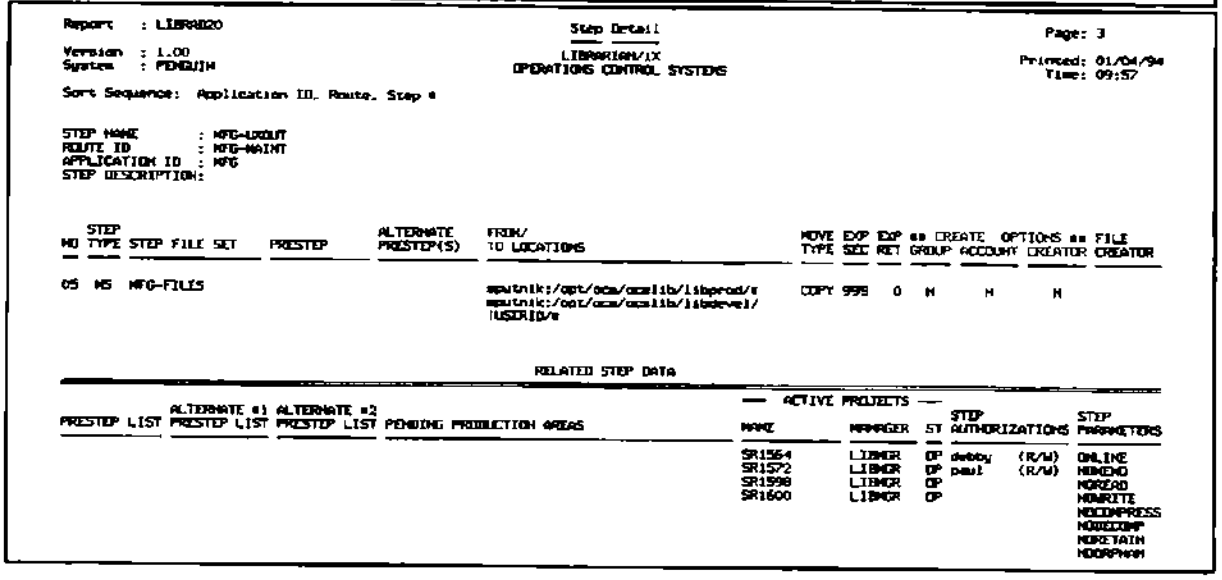
Menu Access **Info...Rules...RAD20 Step Detail**

Description Provides detailed information on the routes and steps in each application. The information for each step includes description, prestep, alternate prestep(s), source/target locations, movement type, step options, and related step data.

Sample



```
Report   : LIBRAD20                              Step Detail                                    Page: 2
                                                 -----------
Version  : 1.00                                LIBRARIAN/iX                             Printed: 01/04/94
System   : PENGUIN                      OPERATIONS CONTROL SYSTEMS                          Time: 09:57

Sort Sequence:  Application ID, Route, Step #

STEP NAME        : MFG-NEW
ROUTE ID         : MFG-MAINT
APPLICATION ID   : MFG
STEP DESCRIPTION: Introduce new files from development


   STEP                                ALTERNATE    FROM/                         MOVE EXP EXP ** CREATE  OPTIONS ** FILE
NO TYPE STEP FILE SET     PRESTEP      PRESTEP(S)   TO LOCATIONS                  TYPE SEC RET GROUP ACCOUNT CREATOR CREATOR
-- ---- -------------     -------      ----------   ------------                  ---- --- --- ----- ------- ------- -------

05  SS  MFG-FILES                                   PENGUIN:@.!USERID.LIBDEVEL    NULL 999  0   N       N       N
                                                    PENGUIN:@.!USERID.LIBDEVEL


                                              RELATED STEP DATA
=======================================================================================================================
                                                                               --- ACTIVE PROJECTS ---
             ALTERNATE #1 ALTERNATE #2                                                               STEP               STEP
PRESTEP LIST PRESTEP LIST PRESTEP LIST PENDING PRODUCTION AREAS               NAME       MANAGER  ST AUTHORIZATIONS PARAMETERS
------------ ------------ ------------ ------------------------               ----       -------  -- -------------- ----------
                                       PENGUIN:@.@.LIBDEVEL              [IN] SR1564     LIBMGR   OP JOSEPH   (R/W) ONLINE
                                       PENGUIN:=.*.LIBPROD               [PN] SR1572     LIBMGR   OP VERONICA(R/W)  NOMEMO
                                                                              SR1598     LIBMGR   OP debby    (R/W) NWRITE
                                                                              SR1600     LIBMGR   OP paul     (R/W) NOVERIFY
                                                                                                                    INOEXTERNAL
```



```
Report   : LIBRAD20                              Step Detail                                    Page: 3
                                                 -----------
Version  : 1.00                                LIBRARIAN/iX                             Printed: 01/04/94
System   : PENGUIN                      OPERATIONS CONTROL SYSTEMS                          Time: 09:57

Sort Sequence:  Application ID, Route, Step #

STEP NAME        : MFG-LOOUT
ROUTE ID         : MFG-MAINT
APPLICATION ID   : MFG
STEP DESCRIPTION:


   STEP                                ALTERNATE    FROM/                                  MOVE EXP EXP ** CREATE  OPTIONS ** FILE
NO TYPE STEP FILE SET     PRESTEP      PRESTEP(S)   TO LOCATIONS                           TYPE SEC RET GROUP ACCOUNT CREATOR CREATOR
-- ---- -------------     -------      ----------   ------------                           ---- --- --- ----- ------- ------- -------

05  NS  MFG-FILES                                   sputnik:/opt/ocm/ocmlib/libprod/*      COPY 999  0   N       N       N
                                                    sputnik:/opt/ocm/ocmlib/libdevel/
                                                    !USERID/*


                                              RELATED STEP DATA
=======================================================================================================================
                                                                          --- ACTIVE PROJECTS ---
             ALTERNATE #1 ALTERNATE #2                                                          STEP               STEP
PRESTEP LIST PRESTEP LIST PRESTEP LIST PENDING PRODUCTION AREAS          NAME       MANAGER  ST AUTHORIZATIONS PARAMETERS
------------ ------------ ------------ ------------------------          ----       -------  -- -------------- ----------
                                                                         SR1564     LIBMGR   OP debby    (R/W) ONLINE
                                                                         SR1572     LIBMGR   OP paul     (R/W) NOMEMO
                                                                         SR1598     LIBMGR   OP                NOREAD
                                                                         SR1600     LIBMGR   OP                NOWRITE
                                                                                                               NOCOMPRESS
                                                                                                               NODECOMP
                                                                                                               NORETAIN
                                                                                                               NOORPHAN
```

STEP DETAIL REPORT (continued) RAD20

Field Descriptions

Step Name	Name of the step.
Route ID	Name of the route.
Application ID	Name of the application.
Step Description	Description of the step.
No	Step number.
Step Type	Type of step.
Step File Set	Name of the fileset for the step.
Prestep	Name of the presteps, if any.
Alternate Prestep(s)	Name of the alternate presteps, if any.
From/To Locations	Source and destination locations for the step.
Move Type	Type of movement (copy, move, null step).
Exp Sec	Number of days before read–mode secondaries expire.
Exp Ret	Number of days before safety–retained files expire.
Create Options	
Group	Specifies creation of a group, when necessary (Y/N).
Account	Specifies creation of an account, when necessary (Y/N).
Creator	Specifies creation of the creator, when necessary (Y/N).
File Creator	The creator to assign to destination files.
Related Step Data	
Prestep List	Name of the prestep.
Alternate Prestep List #1	Name of the first alternate prestep.
Alternate Prestep List #2	Name of the second alternate prestep.
Pending Production Areas	Location of the pending production area(s) to introduce new files. Shows areas to include [IN], areas to exclude [EX], and pending master edit mask [PM].
Name	Name of the project.
Manager	Name of the project manager.
ST	Current status of the project (OP=open, DC=documented).
Step Authorizations	Name of persons authorized to use the step.
Step Parameters	Default parameters on the step (! indicates that no override is allowed).

Field Descriptions, continued

Step Refinements (if applicable)

Seq	Refinement/exception check sequence.
From Fileset	Fileset selection criterion.
FCODE	Filecode selection criterion.
From Location	Location (name) selection criterion.
To Location	Refined destination location
Move Type	Refined movement type:

EXCL	Exclude qualifying files from step.
COPY	Copy qualifying files rather than move.
MOVE	Move qualifying files rather than copy.
NULL	Record as null step for qualifying files.

Forward Versioning (Alternate Search Locations) (if applicable)

Sequence	Alternate search sequence number
From Location	Location to search if file(s) not found in step source location

AUTO FILESETS REPORT RAF10

Sort Sequence	Application, Level, Fileset	
Files Accessed	D-AUTO-FILESET D-FSET-COMPONENT D-SYSTEM-PROFILE	M-APPLICATION M-FILE-SET
When to Run	On demand	
Menu Access	**Info...Files...RAF10 Auto Filesets**	

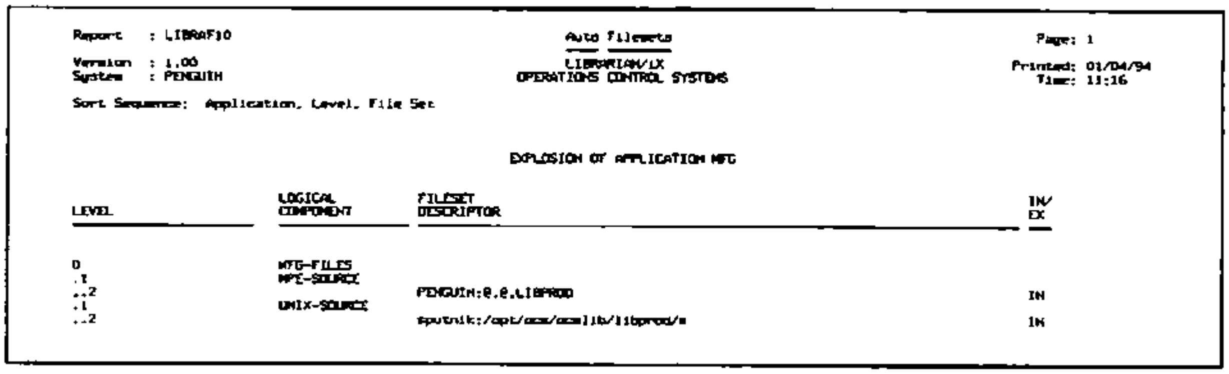
Description

Identifies the logical components of an application or fileset, and auto–fileset descriptors. Components are listed by level to let you review the application hierarchy.

When running this report, you are prompted for the name of the fileset or application you want to explode.

- If you enter a *fileset name,* only that fileset is exploded.
- If you enter an *application name,* the application fileset and all of its components are exploded.
- If you enter an asterisk (*), all of the application filesets in the database are exploded.

Sample



```
Report   : LIBRAF10                      Auto Filesets                         Page: 1

Version  : 1.00                          LIBRARIAN/LX                      Printed: 01/04/94
System   : PENGUIN                   OPERATIONS CONTROL SYSTEMS               Time: 11:16

Sort Sequence:  Application, Level, File Set


                                     EXPLOSION OF APPLICATION MFG

                        LOGICAL       FILESET                                          IN/
LEVEL                   COMPONENT     DESCRIPTOR                                       EX


0                       MFG-FILES
.1                      MPE-SOURCE
..2                                   PENGUIN:@.@.LIBPROD                              IN
.1                      UNIX-SOURCE
..2                                   sputnik:/opt/ocs/ocslib/libprod/*                IN
```

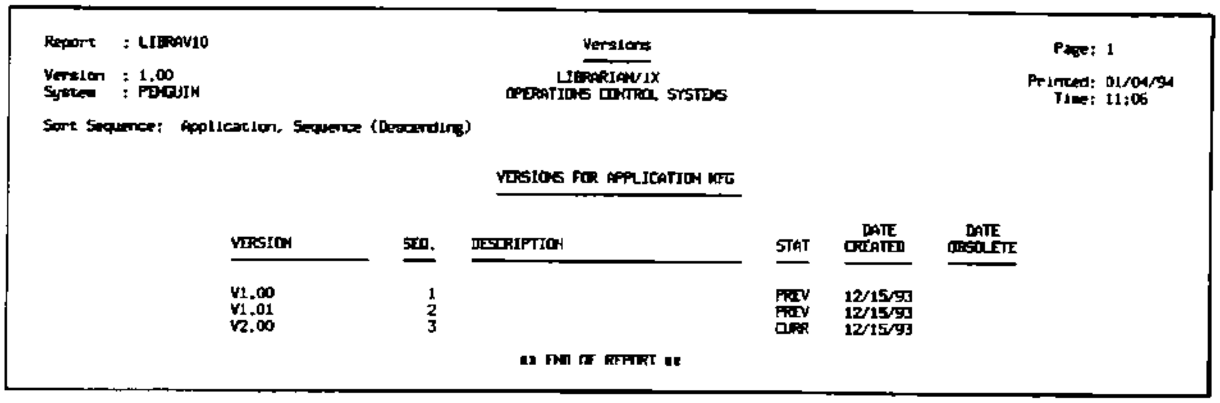
Field Descriptions

Level	The level of the fileset in the hierarchy.
Logical Component	The name of the fileset.
Fileset Descriptor	The location of files associated with the fileset.
Include/Exclude	Specifies whether files identified by the descriptor should be included or excluded from the fileset (IN=include, EX=exclude).

VERSIONS REPORT RAV10

Sort Sequence	Application, Sequence (descending)
Files Accessed	D-APPL-VERSION D-SYSTEM-PROFILE
When to Run	On demand
Menu Access	**Info...Versions...RAV10 Versions**
Description	Identifies all defined versions in the LIBRARIAN system. This report provides an overview of each version, including version name, current status of the version, and dates created and obsoleted.

Sample



```
Report   : LIBRAV10                          Versions                              Page: 1

Version  : 1.00                            LIBRARIAN/iX                     Printed: 01/04/94
System   : PENGUIN                  OPERATIONS CONTROL SYSTEMS                 Time: 11:06

Sort Sequence:  Application, Sequence (Descending)

                                   VERSIONS FOR APPLICATION MFG

                                                                   DATE       DATE
             VERSION       SEQ.   DESCRIPTION              STAT   CREATED    OBSOLETE
             ________      ___    ___________________      ____   ________   ________

             V1.00           1                             PREV   12/15/93
             V1.01           2                             PREV   12/15/93
             V2.00           3                             CURR   12/15/93
                                  ** END OF REPORT **
```

Field Descriptions

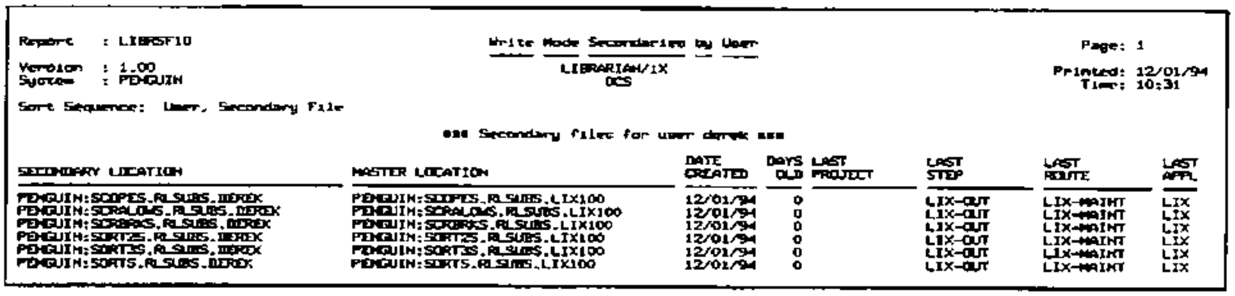
Version	Versions of the application, listed in ascending order.
Sequence	Chronological number of the version in the application.
Description	Description of the version.
Status	Status of the version, as follows:

CURR	Current version
PREV	Previous version
OBS	Obsolete
FL	Flushed

Date Created	Date the version was created.
Date Obsolete	Date the version was made obsolete.

WRITE MODE SECONDARIES BY USER RSF10

Sort Sequence	User, Secondary File
Files Accessed	M–APPLICATION D–SYSTEM–PROFILE D–FILE M–USER
When to Run	On demand
Menu Access	**Info...Files...RSF10 Write Mode Secondaries by User**
Description	Shows write–mode secondary files sorted by the user who currently owns these files. When running this report, you are prompted to enter the name of a user. To select all users, enter asterisk (*).



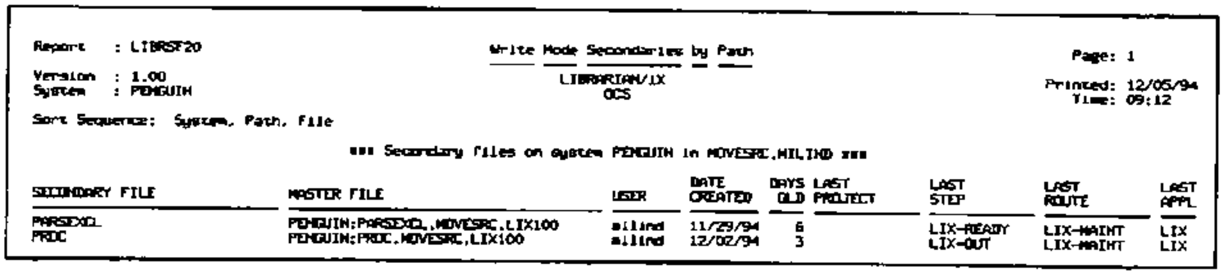
```
Report    : LIBRSF10                      Write Mode Secondaries by User                                   Page: 1
Version   : 1.00                                  LIBRARIAN/IX                                     Printed: 12/01/94
System    : PENGUIN                                   OCS                                             Time: 10:31
Sort Sequence:  User, Secondary File

                                         *** Secondary files for user derek ***

                                                                  DATE      DAYS LAST          LAST        LAST        LAST
SECONDARY LOCATION               MASTER LOCATION                  CREATED    OLD PROJECT       STEP        ROUTE       APPL
PENGUIN:SCOPES.RLSUBS.DEREK      PENGUIN:SCOPES.RLSUBS.LIX100     12/01/94     0               LIX-OUT     LIX-MAINT   LIX
PENGUIN:SCRALOWS.RLSUBS.DEREK    PENGUIN:SCRALOWS.RLSUBS.LIX100   12/01/94     0               LIX-OUT     LIX-MAINT   LIX
PENGUIN:SCRBRKS.RLSUBS.DEREK     PENGUIN:SCRBRKS.RLSUBS.LIX100    12/01/94     0               LIX-OUT     LIX-MAINT   LIX
PENGUIN:SORT2S.RLSUBS.DEREK      PENGUIN:SORT2S.RLSUBS.LIX100     12/01/94     0               LIX-OUT     LIX-MAINT   LIX
PENGUIN:SORT3S.RLSUBS.DEREK      PENGUIN:SORT3S.RLSUBS.LIX100     12/01/94     0               LIX-OUT     LIX-MAINT   LIX
PENGUIN:SORTS.RLSUBS.DEREK       PENGUIN:SORTS.RLSUBS.LIX100      12/01/94     0               LIX-OUT     LIX-MAINT   LIX
```

Field Descriptions

Secondary Location	Location of the write–mode secondary file.
Master Location	Location of the associated master file.
Date Created	Date on which the file was created.
Days Old	Number of days since the write–mode secondary file was created.
Last Step	Last step with which this file is associated.
Last Route	Last route with which this file is associated.
Last Appl	Last application with which this file is associated.

WRITE MODE SECONDARIES BY PATH **RSF20**

Sort Sequence	System, Path, File
Files Accessed	M–APPLICATION D–SYSTEM–PROFILE D–FILE M–USER
When to Run	On demand
Menu Access	**Info...Files...RSF20 Write Mode Secondaries by Path**
Description	Shows write–mode secondary files sorted by location.



```
Report   : LIBRSF20                         Write Mode Secondaries by Path                                    Page: 1

Version  : 1.00                                      LIBRARIAN/iX                                        Printed: 12/05/94
System   : PENGUIN                                       OCS                                                Time: 09:12

Sort Sequence:  System, Path, File

                              *** Secondary files on system PENGUIN in MOVESRC,MILIND ***

                                                                  DATE      DAYS LAST                LAST          LAST           LAST
SECONDARY FILE         MASTER FILE                        USER    CREATED    OLD PROJECT             STEP          ROUTE          APPL
____________________   _________________________________  ______  ________  ____ ____________        ____________  _____________  ____
PARSEXCL               PENGUIN:PARSEXCL,MOVESRC,LIX100    milind  11/29/94     6                     LIX-READY     LIX-MAINT      LIX
PROC                   PENGUIN:PROC,MOVESRC,LIX100        milind  12/02/94     3                     LIX-OUT       LIX-MAINT      LIX
```

Field Descriptions

Secondary File	Name of the write–mode secondary file.
Master File	Name of the associated master file.
User	Name of the user who owns the file.
Date Created	Date on which the file was created.
Days Old	Number of days since the write–mode secondary file was created.
Last Step	Last step with which this file is associated.
Last Route	Last route with which this file is associated.
Last Appl	Last application with which this file is associated.

FILESET STATUS REPORT RFD10

Sort Sequence	Application, Fileset, System, Path, Filename
Files Accessed	D-AHFSET-COMP, D-AHFSET–FILE, D-APPL-VERSION, D-FILE, D-FSET-COMP, D-FSET-FILE, D-SYSTEM-PROFILE, D-VFSET-COMP, D-VFSET-FILE, M-APPLICATION, M-FILE-SET
When to Run	On demand
Menu Access	**Info...Files...RFD10 Fileset Status**

Description

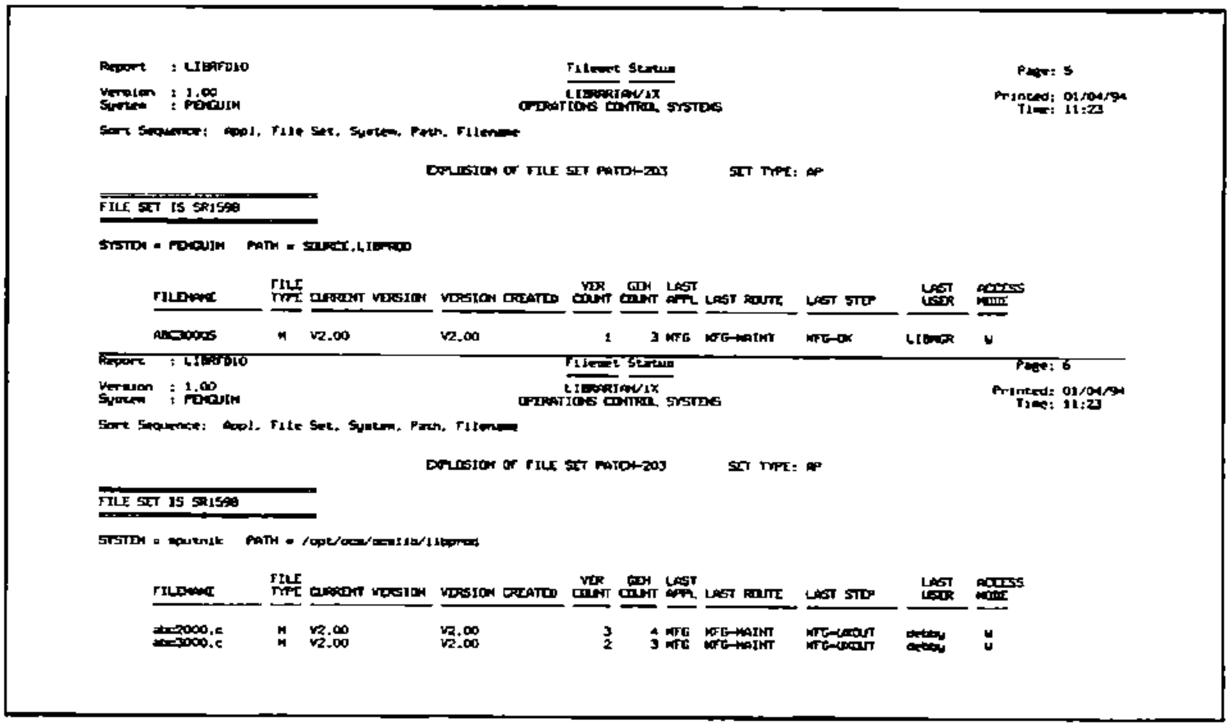
Identifies all files in a fileset or application. This report includes the version history, version and generation counts, file type, access mode, and most recent transaction.

When running this report, you are prompted for the name of the fileset or application you want to explode.

- If you enter a *fileset name*, only that fileset is exploded.
- If you enter an *application name*, the application fileset and all of its components are exploded. You are prompted for a version identifier. Enter a version name or press RETURN to use the current version.

  Each application begins on a new page.
- If you enter an asterisk (*), all of the application filesets in the database are exploded.

Sample



```
Report   : LIBRFD10                          Fileset Status                                 Page: 5
Version  : 1.00                               LIBRARIAN/iX                         Printed: 01/04/94
System   : PENGUIN                  OPERATIONS CONTROL SYSTEMS                        Time: 11:23
Sort Sequence:  Appl, File Set, System, Path, Filename

                         EXPLOSION OF FILE SET PATCH-203        SET TYPE: AP

FILE SET IS SR1598

SYSTEM = PENGUIN   PATH = SOURCE.LIBPROD

                  FILE                                      VER   GEN  LAST                                   LAST   ACCESS
    FILENAME      TYPE CURRENT VERSION  VERSION CREATED  COUNT COUNT APPL LAST ROUTE   LAST STEP      USER   MODE

    ABC3000S       M   V2.00            V2.00                1     3 MFG  MFG-MAINT    MFG-OK         LIBMGR   W
Report   : LIBRFD10                          Fileset Status                                 Page: 6
Version  : 1.00                               LIBRARIAN/iX                         Printed: 01/04/94
System   : PENGUIN                  OPERATIONS CONTROL SYSTEMS                        Time: 11:23
Sort Sequence:  Appl, File Set, System, Path, Filename

                         EXPLOSION OF FILE SET PATCH-203        SET TYPE: AP

FILE SET IS SR1598

SYSTEM = sputnik   PATH = /opt/ocs/ocslib/libprod

                  FILE                                      VER   GEN  LAST                                   LAST   ACCESS
    FILENAME      TYPE CURRENT VERSION  VERSION CREATED  COUNT COUNT APPL LAST ROUTE   LAST STEP      USER   MODE

    abc2000.c      M   V2.00            V2.00                3     4 MFG  MFG-MAINT    MFG-LAYOUT     debby    W
    abc3000.c      M   V2.00            V2.00                2     3 MFG  MFG-MAINT    MFG-LAYOUT     debby    U
```

Field Descriptions

Filename	Name of the file in the application.
File Type	Type of file:

M	Master
S	Secondary
GM	Generated master
GS	Generated secondary
CM	Copies of retained master
CS	Copies of retained secondary

Current Version	Current version of the file.
Version Created	Version when the file was created.
Ver Count	Version count (VCOUNT) for the file.
Gen Count	Generation count (GCOUNT) for the file.
Last Appl	Last application in which the file was processed.
Last Route	Last route that processed the file.
Last Step	Last step that processed the file.
Last User	Last user who performed a step on the file.
Access Mode	Access mode (W=write, R=read).

MASTER FILE STATUS REPORT RFD20

Sort Sequence Application, Fileset, System, Path, Filename

Files Accessed

D-AHFSET-COMP	D-FSET-COMP	D-VFSET-FILE
D-AHFSET-FILE	D-FSET-FILE	M-FILE-SET
D-APPL-VERSION	D-SYSTEM-PROFILE	M-APPLICATION
D-FILE	D-VFSET-COMP	

When to Run On demand

Menu Access **Info...Files...RFD20 Master File Status**

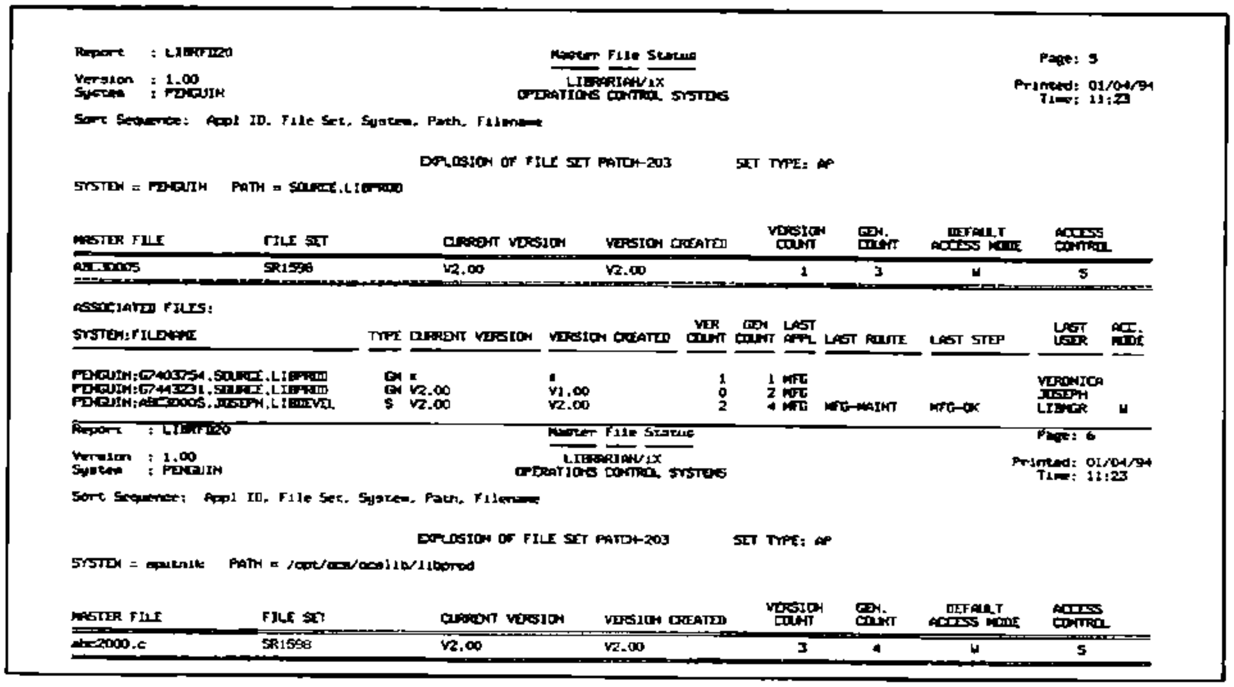
Description Identifies all master files in a fileset or application with their associated files. This report includes version data, version and generation counts, and access control/access mode. Information about the associated files includes file type, version data, version and generation counts, and the last step performed.

When running this report, you are prompted for the name of the fileset or application you want to explode.

- If you enter a *fileset name*, only that fileset is exploded.
- If you enter an *application name*, the application fileset and all of its components are exploded. You are prompted for a version identifier. Enter a version name or press RETURN to use the current version.

  Each application begins on a new page.
- If you enter an asterisk (*), all of the application filesets in the database are exploded.

Sample



```
Report   : LIBRFD20                         Master File Status                                   Page: 5

Version  : 1.00                               LIBRARIAN/iX                              Printed: 01/04/94
System   : PENGUIN                      OPERATIONS CONTROL SYSTEMS                                Time: 11:23

Sort Sequence:  Appl ID, File Set, System, Path, Filename


                              EXPLOSION OF FILE SET PATCH-203        SET TYPE: AP

SYSTEM = PENGUIN    PATH = SOURCE.LIBPROD

                                                                      VERSION    GEN.      DEFAULT      ACCESS
MASTER FILE        FILE SET          CURRENT VERSION   VERSION CREATED  COUNT    COUNT   ACCESS MODE    CONTROL
ABC3000S           SR1598            V2.00             V2.00              1        3          W            S

ASSOCIATED FILES:
                                                                  VER  GEN  LAST                                LAST    ACC.
SYSTEM:FILENAME                     TYPE CURRENT VERSION  VERSION CREATED  COUNT COUNT APPL LAST ROUTE  LAST STEP     USER    MODE

PENGUIN:G7403754.SOURCE.LIBPROD      GN  *                *                  1    1 MFG                             VERONICA
PENGUIN:G7443231.SOURCE.LIBPROD      GN  V2.00            V1.00              0    2 MFG                             JOSEPH
PENGUIN:ABC3000S.JOSEPH.LIBDEVEL     S   V2.00            V2.00              2    4 MFG  MFG-MAINT    MFG-OK        LIBMGR    W
Report   : LIBRFD20                         Master File Status                                   Page: 6

Version  : 1.00                               LIBRARIAN/iX                              Printed: 01/04/94
System   : PENGUIN                      OPERATIONS CONTROL SYSTEMS                                Time: 11:23

Sort Sequence:  Appl ID, File Set, System, Path, Filename


                              EXPLOSION OF FILE SET PATCH-203        SET TYPE: AP

SYSTEM = sputnik    PATH = /opt/ocs/ocslib/libprod

                                                                      VERSION    GEN.      DEFAULT      ACCESS
MASTER FILE        FILE SET          CURRENT VERSION   VERSION CREATED  COUNT    COUNT   ACCESS MODE    CONTROL
abc2000.c          SR1598            V2.00             V2.00              3        4          W            S
```

MASTER FILE STATUS REPORT (continued) RFD20

Field Descriptions

Master File	Name of the master file.
File Set	Name of the fileset to which the master file belongs.
Current Version	Current version number.
Version Created	Version when the master file was created.
Version Count	Version count (VCOUNT).
Gen Count	Generation count (GCOUNT).
Default Access Mode	Default access mode (W=write, R=read).
Access Control	The access control (X=exclusive, R=read, S=serial, M=multiwrite).

FILESET EXPLOSION REPORT RFE10

Sort Sequence Application, Level, Fileset, File ID

Files Accessed

D-AHFSET-COMP	D-FSET-FILE	M-FILE-SET
D-AHFSET-FILE	D-SYSTEM-PROFILE	
D-APPL-VERSION	D-VFSET-COMP	
D-FILE	D-VFSET-FILE	
D-FSET-COMP	M-APPLICATION	

When to Run On demand

Menu Access **Info...Files...RFE10 Fileset Explosion**

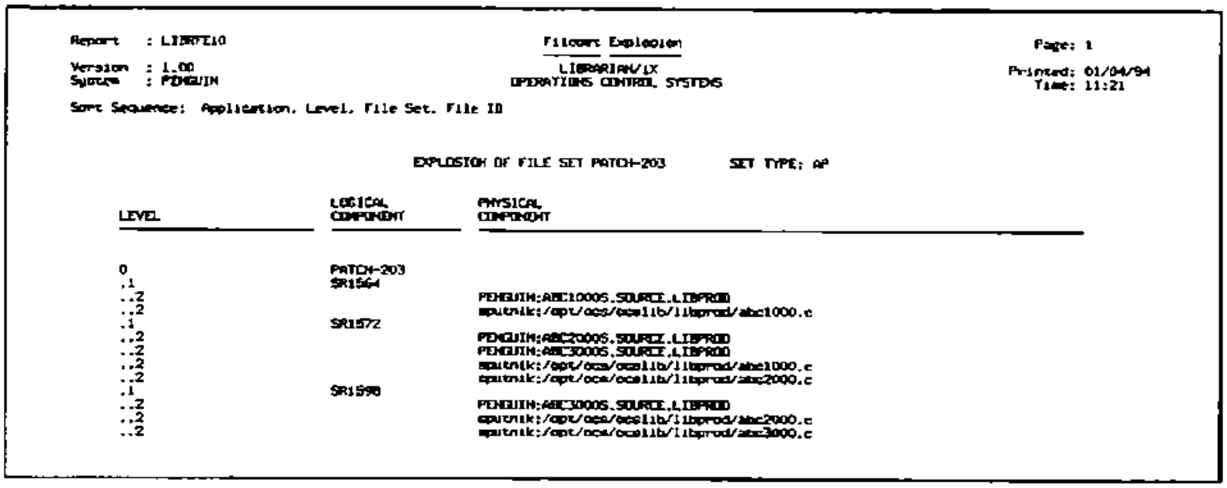
Description Identifies the logical and physical components of a fileset or application. This report lists the components by level to let you review the application hierarchy.

When running this report, you are prompted for the name of the fileset or application you want to explode.

- If you enter a *fileset name*, only that fileset is exploded.
- If you enter an *application name*, the application fileset and all of its components are exploded.

  You are prompted for a version identifier. Enter a version name or press RETURN to use the current version.
- If you enter an asterisk (*), all the application filesets in the database are exploded.

Sample



```
Report   : LIBRFE10                         Fileset Explosion                        Page: 1

Version  : 1.00                               LIBRARIAN/iX                  Printed: 01/04/94
System   : PENGUIN                   OPERATIONS CONTROL SYSTEMS                Time: 11:21

Sort Sequence:  Application, Level, File Set, File ID


                          EXPLOSION OF FILE SET PATCH-203        SET TYPE: AP

                              LOGICAL      PHYSICAL
    LEVEL                     COMPONENT    COMPONENT
    ______________________    _________    ____________________________________________


    0                         PATCH-203
    .1                        SR1564
    ..2                                    PENGUIN:ABC1000S.SOURCE.LIBPROD
    ..2                                    sputnik:/opt/ocs/ocslib/libprod/abc1000.c
    .1                        SR1572
    ..2                                    PENGUIN:ABC2000S.SOURCE.LIBPROD
    ..2                                    PENGUIN:ABC3000S.SOURCE.LIBPROD
    ..2                                    sputnik:/opt/ocs/ocslib/libprod/abc1000.c
    ..2                                    sputnik:/opt/ocs/ocslib/libprod/abc2000.c
    .1                        SR1598
    ..2                                    PENGUIN:ABC3000S.SOURCE.LIBPROD
    ..2                                    sputnik:/opt/ocs/ocslib/libprod/abc2000.c
    ..2                                    sputnik:/opt/ocs/ocslib/libprod/abc3000.c
```

FILESET EXPLOSION REPORT (continued) RFE10

Field Descriptions

Level	The level of the fileset in the hierarchy.
Logical Component	The fileset name.
Physical Component	The names of the files in the fileset.

FILESET EXPLOSION REPORT RFE20

Sort Sequence

Application, Level, Fileset, File ID

Files Accessed

D-AHFSET-COMP
D-AHFSET-FILE
D-APPL-VERSION
D-FILE
D-FSET-COMP
D-FSET-FILE
D-SYSTEM-PROFILE
D-VFSET-COMP
D-VFSET-FILE
M-APPLICATION
M-FILE-SET

When to Run

On demand

Menu Access

Info...Files...RFE20 Fileset Explosion

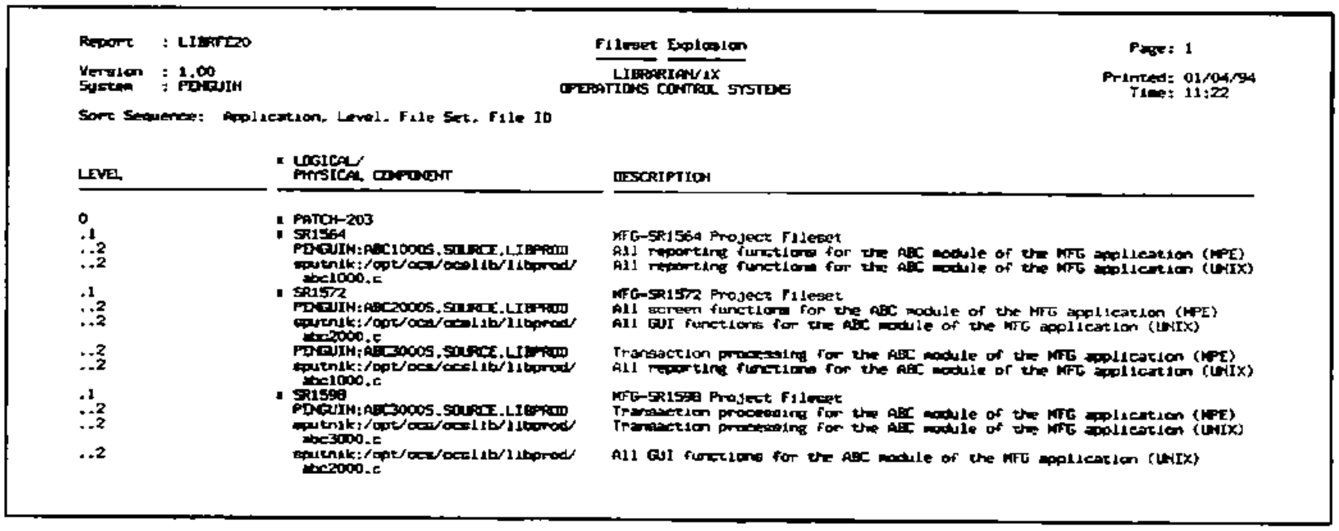
Description

Identifies the logical and physical components in a fileset or application. This report lists the components by level to let you review the application hierarchy.

When running this report, you are prompted for the name of the fileset or application you want to explode.

- If you enter a *fileset name*, only that fileset is exploded.
- If you enter an *application name*, the application fileset and all of its components are exploded. You are prompted for a version identifier. Enter a version name or press RETURN to use the current version.
- If you enter an asterisk (*), all the application filesets in the database are exploded.

Sample



```
Report   : LIBRFE20                        Fileset Explosion                                        Page: 1

Version  : 1.00                               LIBRARIAN/iX                                  Printed: 01/04/94
System   : PENGUIN                      OPERATIONS CONTROL SYSTEMS                              Time: 11:22

Sort Sequence:  Application, Level, File Set, File ID

                     * LOGICAL/
LEVEL                  PHYSICAL COMPONENT                  DESCRIPTION
___________________  _______________________________   _____________________________________________________________

0                    * PATCH-203
.1                   * SR1564                            MFG-SR1564 Project Fileset
..2                    PENGUIN:ABC1000S.SOURCE.LIBPROD   All reporting functions for the ABC module of the MFG application (MPE)
..2                    sputnik:/opt/ocs/ocslib/libprod/  All reporting functions for the ABC module of the MFG application (UNIX)
                        abc1000.c
.1                   * SR1572                            MFG-SR1572 Project Fileset
..2                    PENGUIN:ABC2000S.SOURCE.LIBPROD   All screen functions for the ABC module of the MFG application (MPE)
..2                    sputnik:/opt/ocs/ocslib/libprod/  All GUI functions for the ABC module of the MFG application (UNIX)
                        abc2000.c
..2                    PENGUIN:ABC3000S.SOURCE.LIBPROD   Transaction processing for the ABC module of the MFG application (MPE)
..2                    sputnik:/opt/ocs/ocslib/libprod/  All reporting functions for the ABC module of the MFG application (UNIX)
                        abc1000.c
.1                   * SR1598                            MFG-SR1598 Project Fileset
..2                    PENGUIN:ABC3000S.SOURCE.LIBPROD   Transaction processing for the ABC module of the MFG application (MPE)
..2                    sputnik:/opt/ocs/ocslib/libprod/  Transaction processing for the ABC module of the MFG application (UNIX)
                        abc3000.c
..2                    sputnik:/opt/ocs/ocslib/libprod/  All GUI functions for the ABC module of the MFG application (UNIX)
                        abc2000.c
```

Field Descriptions

Level	Level of the fileset in the hierarchy.
Logical Component	Fileset name preceded by an asterisk (*).
Physical Component	The names of the files in the fileset.
Description	Description of the fileset or physical file (if one exists).

PRE-FLUSH NOTIFICATION REPORT RFN10

Sort Sequence

Master File

Files Accessed

D-APPL-VERSION
D-FILE
D-SYSTEM-PROFILE

When to Run

On demand

Menu Access

Info...Files...RFN10 Pre–Flush Notification

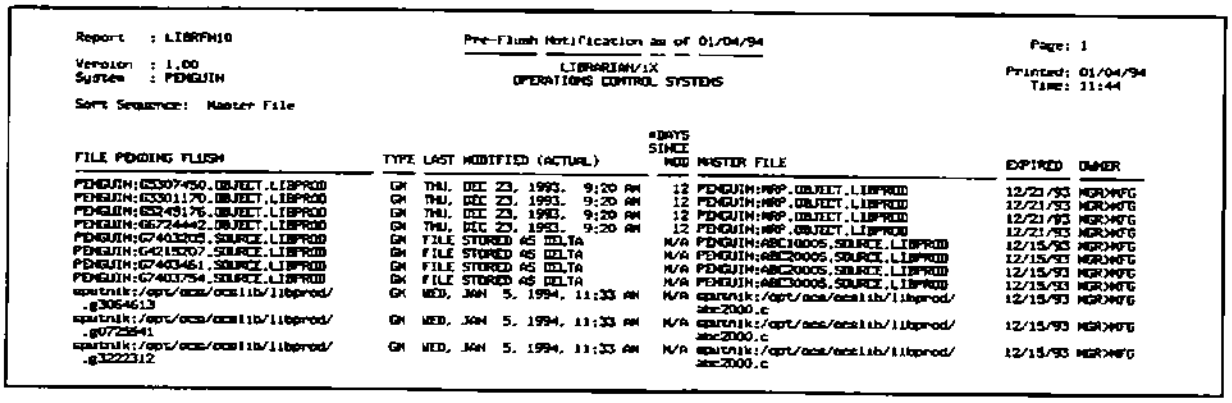
Description

Lists all files eligible to be flushed by the FLUSH utility. For each file, this report shows the file type, the date and time of last modification, the number of days since modification, the associated master file, the expiration date, and the file owner.

The "as of" date for this report can be specified in **MM/DD/YY** format in the **INFO** string of the **RUN** command. The following example lists all files to be flushed on 12/31/92 (if the **FLUSH** program were run on that date).

```
:RUN RFN10P.COMP.OCSLIB; INFO="12/31/92"
```

Sample



```
Report   : LIBRFN10                    Pre-Flush Notification as of 01/04/94                          Page: 1

Version  : 1.00                                 LIBRARIAN/iX                                       Printed: 01/04/94
System   : PENGUIN                        OPERATIONS CONTROL SYSTEMS                                  Time: 11:44

Sort Sequence:  Master File

                                                                   #DAYS
                                                                   SINCE
FILE PENDING FLUSH                     TYPE LAST MODIFIED (ACTUAL)   MOD MASTER FILE                          EXPIRED  OWNER
______________________________________ ____ ______________________ _____ _________________________________  ________ ________
PENGUIN:G5307450.OBJECT.LIBPROD         GM  THU, DEC 23, 1993,  9:20 AM   12 PENGUIN:MRP.OBJECT.LIBPROD         12/21/93 MGR>MFG
PENGUIN:G3301170.OBJECT.LIBPROD         GM  THU, DEC 23, 1993,  9:20 AM   12 PENGUIN:MRP.OBJECT.LIBPROD         12/21/93 MGR>MFG
PENGUIN:G5245176.OBJECT.LIBPROD         GM  THU, DEC 23, 1993,  9:20 AM   12 PENGUIN:MRP.OBJECT.LIBPROD         12/21/93 MGR>MFG
PENGUIN:G6724442.OBJECT.LIBPROD         GM  THU, DEC 23, 1993,  9:20 AM   12 PENGUIN:MRP.OBJECT.LIBPROD         12/21/93 MGR>MFG
PENGUIN:G7403205.SOURCE.LIBPROD         GM  FILE STORED AS DELTA         N/A PENGUIN:ABC10005.SOURCE.LIBPROD    12/15/93 MGR>MFG
PENGUIN:G4215207.SOURCE.LIBPROD         GM  FILE STORED AS DELTA         N/A PENGUIN:ABC20005.SOURCE.LIBPROD    12/15/93 MGR>MFG
PENGUIN:G7403461.SOURCE.LIBPROD         GM  FILE STORED AS DELTA         N/A PENGUIN:ABC20005.SOURCE.LIBPROD    12/15/93 MGR>MFG
PENGUIN:G7403754.SOURCE.LIBPROD         GK  FILE STORED AS DELTA         N/A PENGUIN:ABC30005.SOURCE.LIBPROD    12/15/93 MGR>MFG
sputnik:/opt/ocs/ocslib/libprod/        GK  WED, JAN  5, 1994, 11:33 AM  N/A sputnik:/opt/ocs/ocslib/libprod/   12/15/93 MGR>MFG
 .g3064613                                                                   abc2000.c
sputnik:/opt/ocs/ocslib/libprod/        GM  WED, JAN  5, 1994, 11:33 AM  N/A sputnik:/opt/ocs/ocslib/libprod/   12/15/93 MGR>MFG
 .g0725841                                                                   abc2000.c
sputnik:/opt/ocs/ocslib/libprod/        GM  WED, JAN  5, 1994, 11:33 AM  N/A sputnik:/opt/ocs/ocslib/libprod/   12/15/93 MGR>MFG
 .g3222312                                                                   abc2000.c
```

PRE-FLUSH NOTIFICATION REPORT (continued) RFN10

Field Descriptions

File Pending Flush	The name of the file to be flushed.
Type	The file type:

S	Secondary
GM	Generated Master
GS	Generated Secondary
CM	Copy of Generated Master
CS	Copy of Generated Secondary

Last Modified (Actual)	Date and time the file was last modified or date and time changes were added to the delta file.
#Days Since Mod	Number of days since the file was modified.
Master File	Name of the master file associated with the file to be flushed.
Expired	Date the file expired.
Owner	For Generated Masters, this field displays the application name; otherwise, it displays the name of the file owner.

PRE-FLUSH NOTIFICATION REPORT RFN20

Sort Sequence File Owner, File Type, Master File

Files Accessed A-APPL-VERSION
D-FILE
D-SYSTEM-PROFILE

When to Run On demand

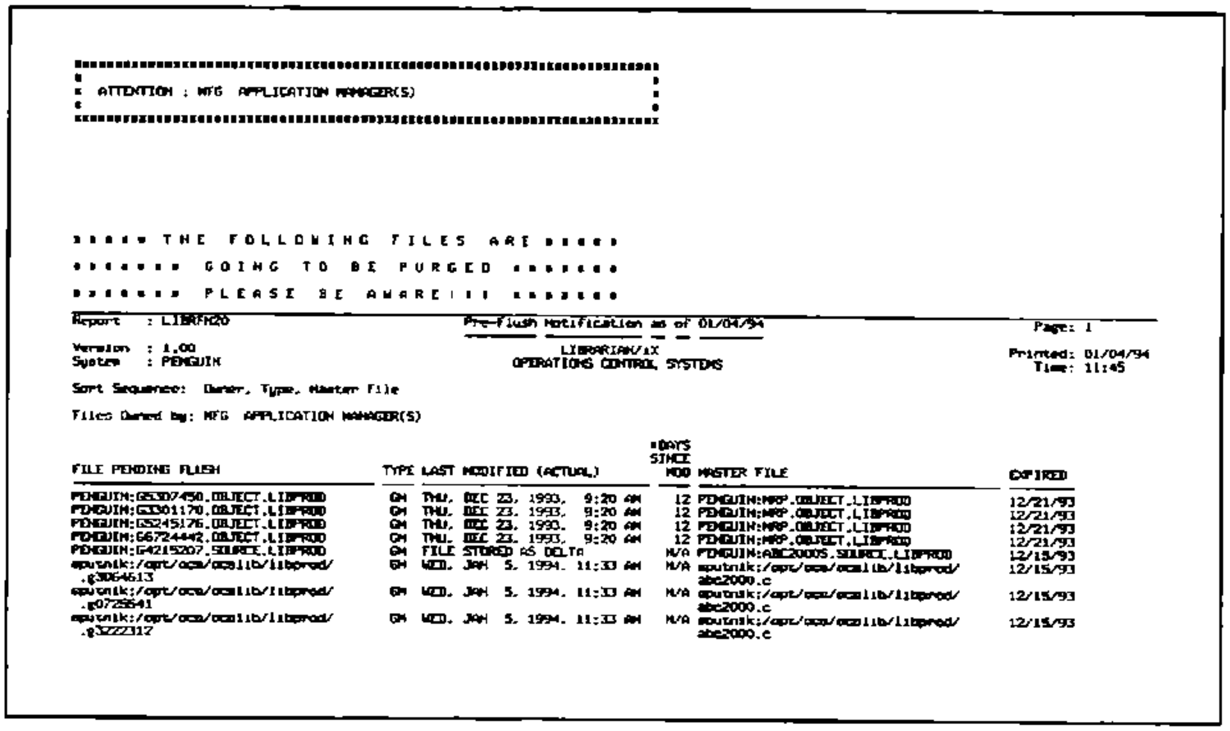
Menu Access **Info...Files...RFN20 Pre–Flush Notification**

Description For each user, this report lists the files that the FLUSH utility is ready to purge. The information for each file owner includes the file, the file type, the date and time of last modification, the number of days since the last modification, the associated master file, and the file expiration date.

The "as of" date for this report can be specified in MM/DD/YY format in the **INFO** string of the **RUN** command. The following example lists all files to be flushed on 12/31/92 (if the **FLUSH** program were run on that date).

```
:RUN RFN10P.COMP.OCSLIB; INFO="12/31/92"
```

Sample



```
****************************************************************************
*                                                                          *
*  ATTENTION : MFG  APPLICATION MANAGER(S)                                 *
*                                                                          *
****************************************************************************




* * * * *  T H E   F O L L O W I N G   F I L E S   A R E  * * * * *
* * * * * * *  G O I N G   T O   B E   P U R G E D  * * * * * * *
* * * * * * *  P L E A S E   B E   A W A R E ! ! !  * * * * * * *
Report   : LIBRFN20                 Pre-Flush Notification as of 01/04/94                 Page: 1

Version  : 1.00                            LIBRARIAN/iX                           Printed: 01/04/94
System   : PENGUIN                  OPERATIONS CONTROL SYSTEMS                          Time: 11:45

Sort Sequence:  Owner, Type, Master File

Files Owned by: MFG  APPLICATION MANAGER(S)

                                                                 #DAYS
                                                                 SINCE
FILE PENDING FLUSH                    TYPE LAST MODIFIED (ACTUAL)  MOD MASTER FILE                          EXPIRED
------------------------------------- ---- ----------------------- --- ------------------------------------ --------
PENGUIN:G5307450.OBJECT.LIBPROD        GM  THU, DEC 23, 1993,  9:20 AM  12 PENGUIN:MRP.OBJECT.LIBPROD       12/21/93
PENGUIN:G3301170.OBJECT.LIBPROD        GM  THU, DEC 23, 1993,  9:20 AM  12 PENGUIN:MRP.OBJECT.LIBPROD       12/21/93
PENGUIN:G5245176.OBJECT.LIBPROD        GM  THU, DEC 23, 1993,  9:20 AM  12 PENGUIN:MRP.OBJECT.LIBPROD       12/21/93
PENGUIN:G6724442.OBJECT.LIBPROD        GM  THU, DEC 23, 1993,  9:20 AM  12 PENGUIN:MRP.OBJECT.LIBPROD       12/21/93
PENGUIN:G4215207.SOURCE.LIBPROD        GM  FILE STORED AS DELTA        N/A PENGUIN:ABC2000S.SOURCE.LIBPROD  12/15/93
sputnik:/opt/ocs/ocslib/libprod/       GM  WED, JAN  5, 1994, 11:33 AM N/A sputnik:/opt/ocs/ocslib/libprod/ 12/15/93
 .g3064613                                                                 abc2000.c
sputnik:/opt/ocs/ocslib/libprod/       GM  WED, JAN  5, 1994, 11:33 AM N/A sputnik:/opt/ocs/ocslib/libprod/ 12/15/93
 .g0725641                                                                 abc2000.c
sputnik:/opt/ocs/ocslib/libprod/       GM  WED, JAN  5, 1994, 11:33 AM N/A sputnik:/opt/ocs/ocslib/libprod/ 12/15/93
 .g3222317                                                                 abc2000.c
```

PRE-FLUSH NOTIFICATION REPORT (continued) RFN20

Field Descriptions

File Pending Flush	The name of the file to be flushed.
Type	The file type:

S	Secondary
GM	Generated Master
GS	Generated Secondary
CM	Copy of Generated Master
CS	Copy of Generated Secondary

Last Modified (Actual)	Date and time the file was last modified or date and time changes were added to the delta file.
#Days Since Mod	The number of days since the file was modified.
Master File	The name of the master file with which the file to be flushed is associated.
Expired	The date the file expired.

FILE EXCEPTIONS REPORT — RFX10

Sort Sequence
File ID (System, Filename)

Files Accessed

D-AHFSET-COMP	D-FSET-COMP	D-VFSET-FILE
D-AHFSET-FILE	D-FSET-FILE	M-APPLICATION
D-APPL-VERSION	D-SYSTEM-PROFILE	M-FILE-SET
D-FILE	D-VFSET-COMP	

When to Run
On demand

Menu Access
Info...Files...RFX20 File Exceptions

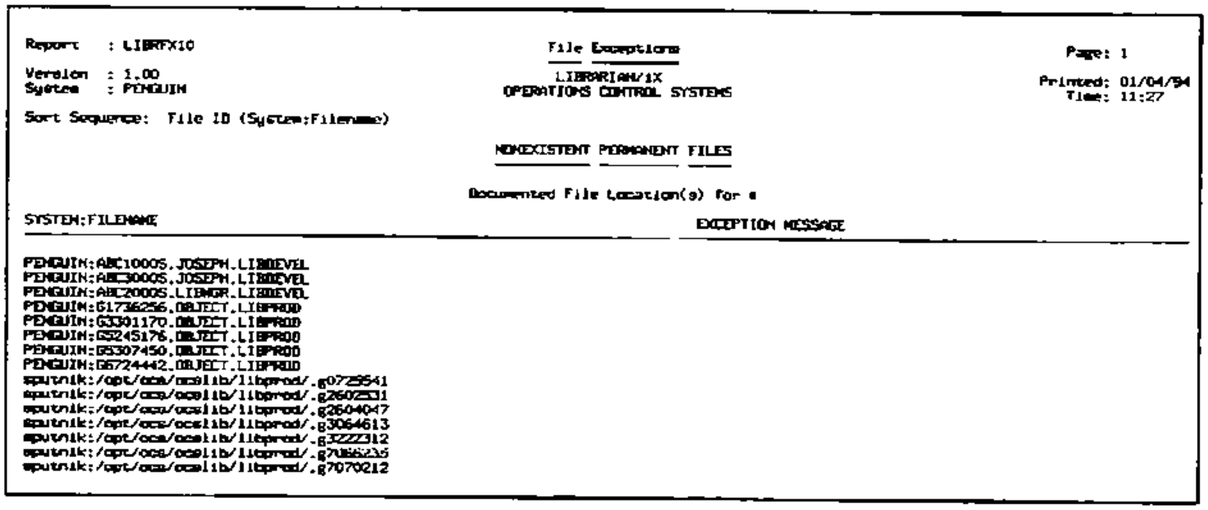
Description
Identifies files that do not currently exist on disk, or disk files that are unknown to LIBRARIAN.

When running this report, you are prompted for a group of physical files using wildcards, a logical fileset, or an application. A logical fileset name must be preceded by a percent sign (%), and an application name must be preceded by a dollar sign ($). Enter an asterisk (*) to explode all application filesets in the database.

You can create a listfile LISTFX10 to include each exception by responding Yes to the "Create list file of exceptions?" prompt. Then you can use the listfile as input to any LIBRARIAN operation (e.g., **VERIFY**, **CLEANDB**).

For wildcard specifications, "FILES UNKNOWN TO LIBRARIAN" are listed. For logical filesets and applications, "NONEXISTENT PERMANENT FILES" are listed.

Sample



```
Report   : LIBRFX10                       File Exceptions                          Page: 1

Version  : 1.00                            LIBRARIAN/iX                    Printed: 01/04/94
System   : PENGUIN                    OPERATIONS CONTROL SYSTEMS                  Time: 11:27

Sort Sequence:  File ID (System:Filename)

                                     NONEXISTENT PERMANENT FILES

                                  Documented File Location(s) for *

SYSTEM:FILENAME                                          EXCEPTION MESSAGE
---------------------------------------------------------------------------------------------

PENGUIN:ABC1000S.JOSEPH.LIBDEVEL
PENGUIN:ABC3000S.JOSEPH.LIBDEVEL
PENGUIN:ABC2000S.LIBMGR.LIBDEVEL
PENGUIN:G1736256.OBJECT.LIBPROD
PENGUIN:G3301170.OBJECT.LIBPROD
PENGUIN:G5245176.OBJECT.LIBPROD
PENGUIN:G5307450.OBJECT.LIBPROD
PENGUIN:G6724442.OBJECT.LIBPROD
sputnik:/opt/ocs/ocslib/libprod/.g0725541
sputnik:/opt/ocs/ocslib/libprod/.g2602531
sputnik:/opt/ocs/ocslib/libprod/.g2604047
sputnik:/opt/ocs/ocslib/libprod/.g3064613
sputnik:/opt/ocs/ocslib/libprod/.g3222312
sputnik:/opt/ocs/ocslib/libprod/.g7066235
sputnik:/opt/ocs/ocslib/libprod/.g7070212
```

Field Descriptions

System:Filename	The name of the system and file.
Exception Message	If there was a problem determining a file's existence, a message is shown here.

GENERATED FILES REPORT RGF10

Sort Sequence File

Files Accessed D-FILE

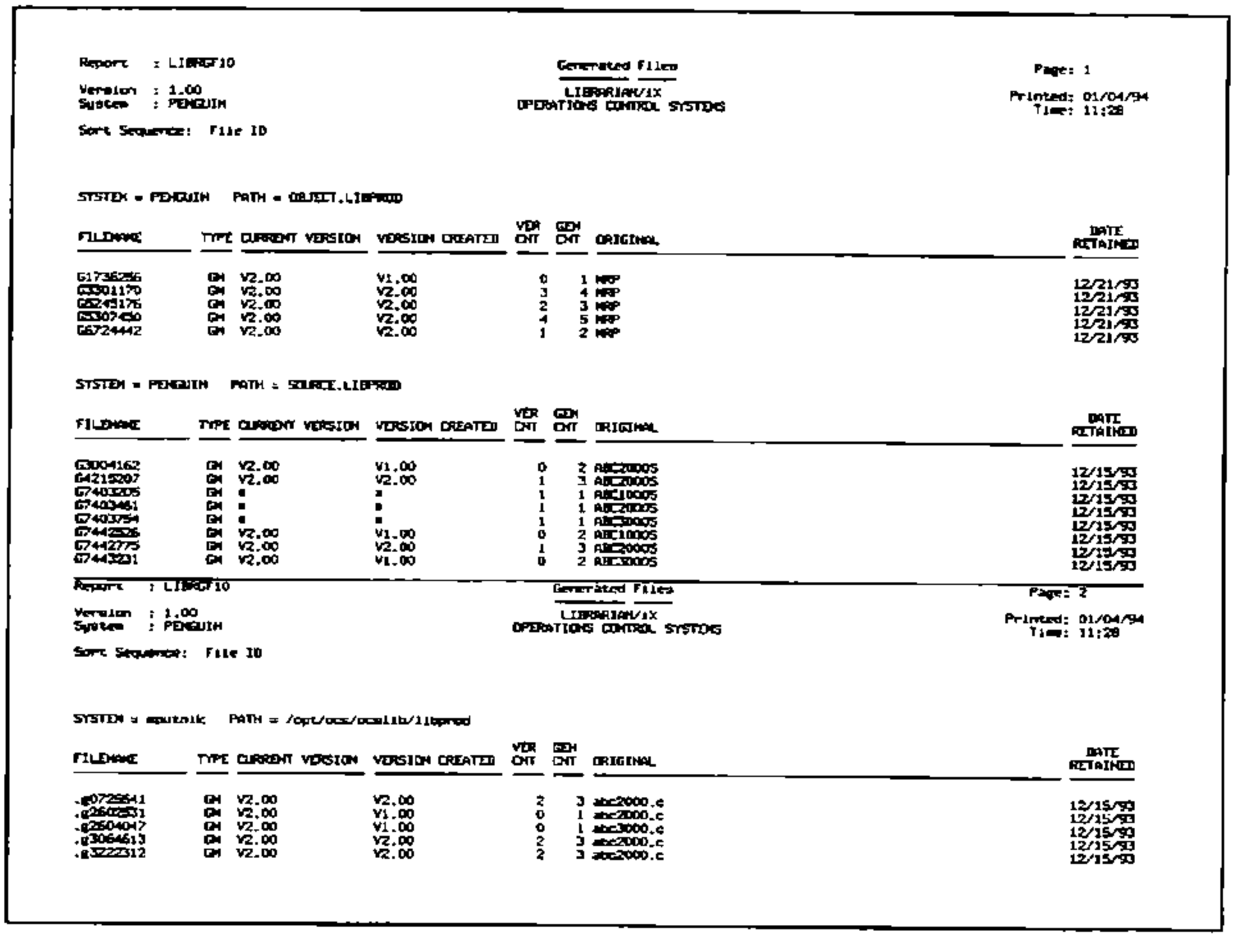
When to Run On demand

Menu Access **Info...Files...RGF10 Generated Files**

Description This report identifies retained files tracked by LIBRARIAN. These files are retained through version activity or safety retention.

The information listed on this report includes the type of retained file, version data, version and generation counts, and the original filename. Use this report to review the status of the retained files and to cross-reference retained filenames with their original filenames.

Sample



```
Report   : LIBRGF10                         Generated Files                            Page: 1

Version  : 1.00                               LIBRARIAN/iX                     Printed: 01/04/94
System   : PENGUIN                    OPERATIONS CONTROL SYSTEMS                  Time: 11:28

Sort Sequence:  File ID


SYSTEM = PENGUIN   PATH = OBJECT.LIBPROD

                                                     VER  GEN                                        DATE
FILENAME        TYPE CURRENT VERSION  VERSION CREATED CNT  CNT  ORIGINAL                             RETAINED

G1736256         GM  V2.00            V1.00             0    1 MRP                                  12/21/93
G3301170         GM  V2.00            V2.00             3    4 MRP                                  12/21/93
G5245176         GM  V2.00            V2.00             2    3 MRP                                  12/21/93
G5307450         GM  V2.00            V2.00             4    5 MRP                                  12/21/93
G6724442         GM  V2.00            V2.00             1    2 MRP                                  12/21/93

SYSTEM = PENGUIN   PATH = SOURCE.LIBPROD

                                                     VER  GEN                                        DATE
FILENAME        TYPE CURRENT VERSION  VERSION CREATED CNT  CNT  ORIGINAL                             RETAINED

G3004162         GM  V2.00            V1.00             0    2 ABC2000S                             12/15/93
G4215207         GM  V2.00            V2.00             1    3 ABC2000S                             12/15/93
G7403205         GM  *                *                 1    1 ABC1000S                             12/15/93
G7403461         GM  *                *                 1    1 ABC2000S                             12/15/93
G7403754         GM  *                *                 1    1 ABC3000S                             12/15/93
G7442526         GM  V2.00            V1.00             0    2 ABC1000S                             12/15/93
G7442775         GM  V2.00            V2.00             1    3 ABC2000S                             12/15/93
G7443231         GM  V2.00            V1.00             0    2 ABC3000S                             12/15/93
Report   : LIBRGF10                         Generated Files                            Page: 2

Version  : 1.00                               LIBRARIAN/iX                     Printed: 01/04/94
System   : PENGUIN                    OPERATIONS CONTROL SYSTEMS                  Time: 11:28

Sort Sequence:  File ID


SYSTEM = sputnik   PATH = /opt/ocs/ocslib/libprod

                                                     VER  GEN                                        DATE
FILENAME        TYPE CURRENT VERSION  VERSION CREATED CNT  CNT  ORIGINAL                             RETAINED

.g0729641        GM  V2.00            V2.00             2    3 abc2000.c                            12/15/93
.g2602531        GM  V2.00            V1.00             0    1 abc2000.c                            12/15/93
.g2604047        GM  V2.00            V1.00             0    1 abc3000.c                            12/15/93
.g3064613        GM  V2.00            V2.00             2    3 abc2000.c                            12/15/93
.g3222312        GM  V2.00            V2.00             2    3 abc2000.c                            12/15/93
```

GENERATED FILES REPORT (continued) RGF10

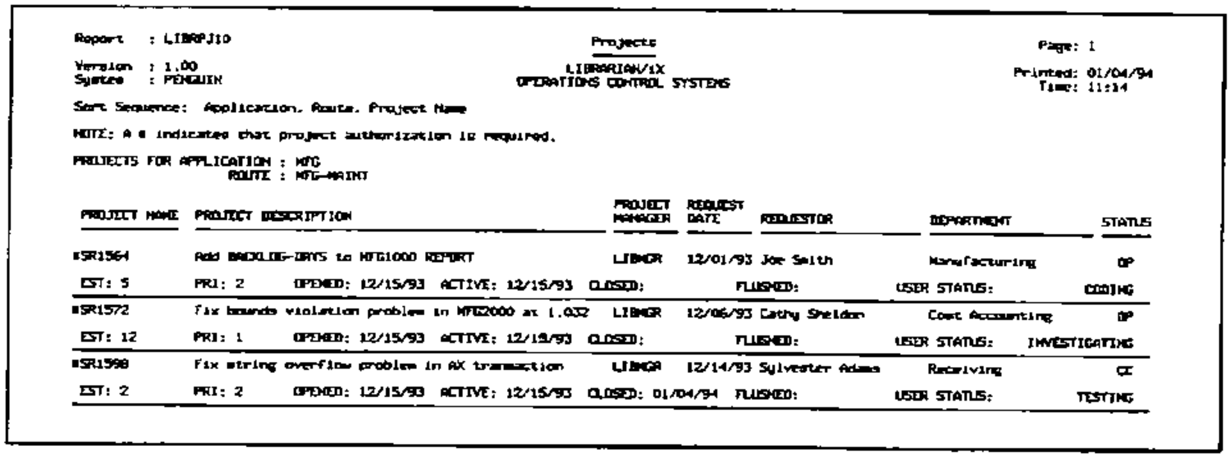
Field Descriptions

Filename	The name of the generated file.
Type	The type of file (GM=generated master, GS=generated secondary).
Current Version	The version of the file.
Version Created	The version when the file was first introduced.
Vers Cnt	The version count (VCOUNT) for the file.
Gen Cnt	The generation count (GCOUNT) for the file.
Original	The name of the related master or secondary.
Date Retained	The date the file was retained.

PROJECTS REPORT

Sort Sequence	Application, Route, Project Name
Files Accessed	D-PROJECTS D-SYSTEM-PROFILE M-APPLICATION
When to Run	On demand
Menu Access	**Info...Rules...RPJ10 Projects**
Description	Shows information on all projects, including the current project status, authorization requirements, the project route ID, project manager, and the dates the project was opened, activated, closed, and flushed.

Sample



```
Report  : LIBRPJ10                          Projects                                        Page: 1

Version : 1.00                          LIBRARIAN/iX                                Printed: 01/04/94
System  : PENGUIN                 OPERATIONS CONTROL SYSTEMS                           Time: 11:14

Sort Sequence:  Application, Route, Project Name

NOTE: A # indicates that project authorization is required.

PROJECTS FOR APPLICATION : MFG
                   ROUTE : MFG-MAINT

                                                          PROJECT  REQUEST
PROJECT NAME  PROJECT DESCRIPTION                         MANAGER  DATE     REQUESTOR            DEPARTMENT          STATUS

#SR1564       Add BACKLOG-DAYS to MFG1000 REPORT          LIBMGR   12/01/93 Joe Smith            Manufacturing           OP
 EST: 5       PRI: 2    OPENED: 12/15/93  ACTIVE: 12/15/93  CLOSED:           FLUSHED:      USER STATUS:          CODING
#SR1572       Fix bounds violation problem in MFG2000 at 1.032  LIBMGR  12/06/93 Cathy Sheldon  Cost Accounting     OP
 EST: 12      PRI: 1    OPENED: 12/15/93  ACTIVE: 12/19/93  CLOSED:           FLUSHED:      USER STATUS:   INVESTIGATING
#SR1598       Fix string overflow problem in AX transaction  LIBMGR  12/14/93 Sylvester Adams  Receiving           CE
 EST: 2       PRI: 2    OPENED: 12/15/93  ACTIVE: 12/15/93  CLOSED: 01/04/94  FLUSHED:      USER STATUS:         TESTING
```

Field Descriptions

Project Name	The name of the project.
Project Description	A short description of the project.
Project Manager	The name of the project manager.
Request Date	The date the project was requested (documentation only).
Requestor	The name of the person requesting the project (documentation only).
Department	The name of the department requesting the project (documentation only).

Field Descriptions, continued

Status	Status of the project.
	CC — Closed to Checkout
	CL — Closed
	DC — Documented (but not opened yet)
	FP — Flush Pending
	OP — Open
EST	Estimated length of the project.
PRI	Priority (documentation only).
Opened	Date the project was opened.
Active	Date the project was first used.
Closed	Date the project was closed.
Flushed	Date the project was flushed.
User Status	User-defined status description (documentation only).

PENDING MASTER FILES REPORT RPM10

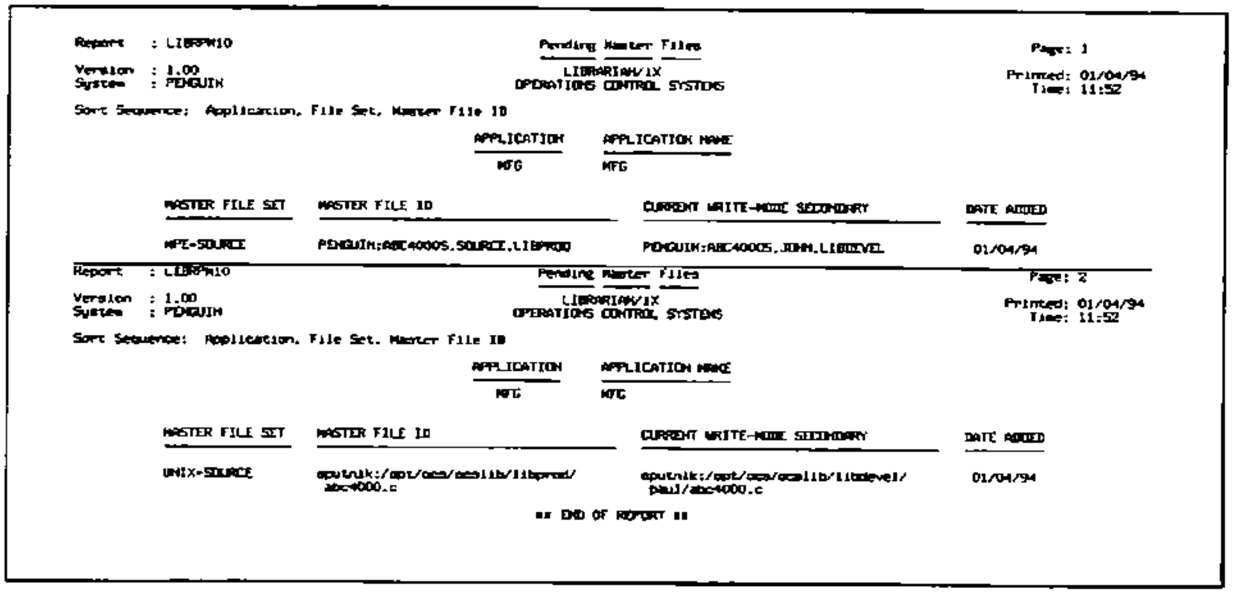
Sort Sequence	Application, Fileset, Master File ID
Files Accessed	D-FILE, D-FSET-COMP, D-FSET-FILE, D-SYSTEM-PROFILE, M-APPLICATION, M-FILE-SET
When to Run	On demand
Menu Access	**Info...Files...RPM10 Pending Master Files**

Description

Identifies all pending master files defined in the LIBRARIAN system. A *pending master file* is a new file that has not been checked in yet. A master record exists in the database, but the file has not yet been checked in to the library.

The fields on this report include the master fileset and pending master filename as well as the name of its current write mode secondary. Use this report to review the status of your pending master files.

Sample



```
Report   : LIBRPM10                        Pending Master Files                          Page: 1
Version  : 1.00                               LIBRARIAN/iX                        Printed: 01/04/94
System   : PENGUIN                      OPERATIONS CONTROL SYSTEMS                   Time: 11:52
Sort Sequence:  Application, File Set, Master File ID
                                         APPLICATION    APPLICATION NAME
                                             MFG        MFG

          MASTER FILE SET   MASTER FILE ID                       CURRENT WRITE-MODE SECONDARY        DATE ADDED

          MPE-SOURCE        PENGUIN:ABC4000S.SOURCE.LIBPROD      PENGUIN:ABC4000S.JOHN.LIBDEVEL       01/04/94
Report   : LIBRPM10                        Pending Master Files                          Page: 2
Version  : 1.00                               LIBRARIAN/iX                        Printed: 01/04/94
System   : PENGUIN                      OPERATIONS CONTROL SYSTEMS                   Time: 11:52
Sort Sequence:  Application, File Set, Master File ID
                                         APPLICATION    APPLICATION NAME
                                             MFG        MFG

          MASTER FILE SET   MASTER FILE ID                       CURRENT WRITE-MODE SECONDARY        DATE ADDED

          UNIX-SOURCE       sputnik:/opt/ocs/ocslib/libprod/     sputnik:/opt/ocs/ocslib/libdevel/    01/04/94
                             abc4000.c                            paul/abc4000.c
                                        ** END OF REPORT **
```

Field Descriptions

Application	The unique identifier for the application.
Application Name	The name of the application.
Master File Set	The name of the master fileset containing all filesets and files for the application.
Master File ID	The pending master filename.

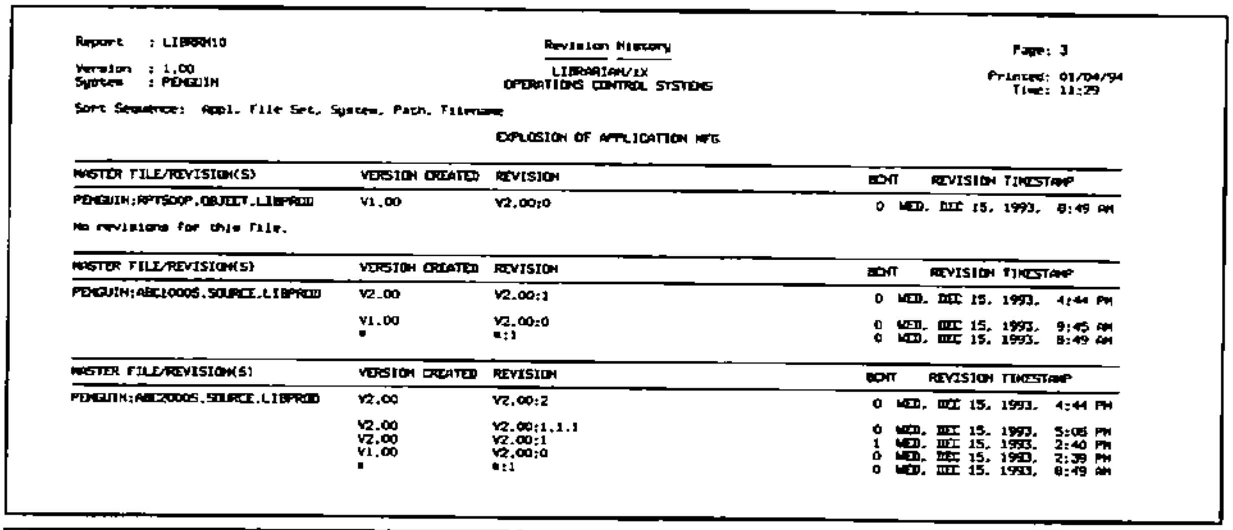
Field Descriptions, continued

Current Write-Mode Secondary	The name of the current write mode secondary.
Date Added	The date the pending master file record was added to the application.

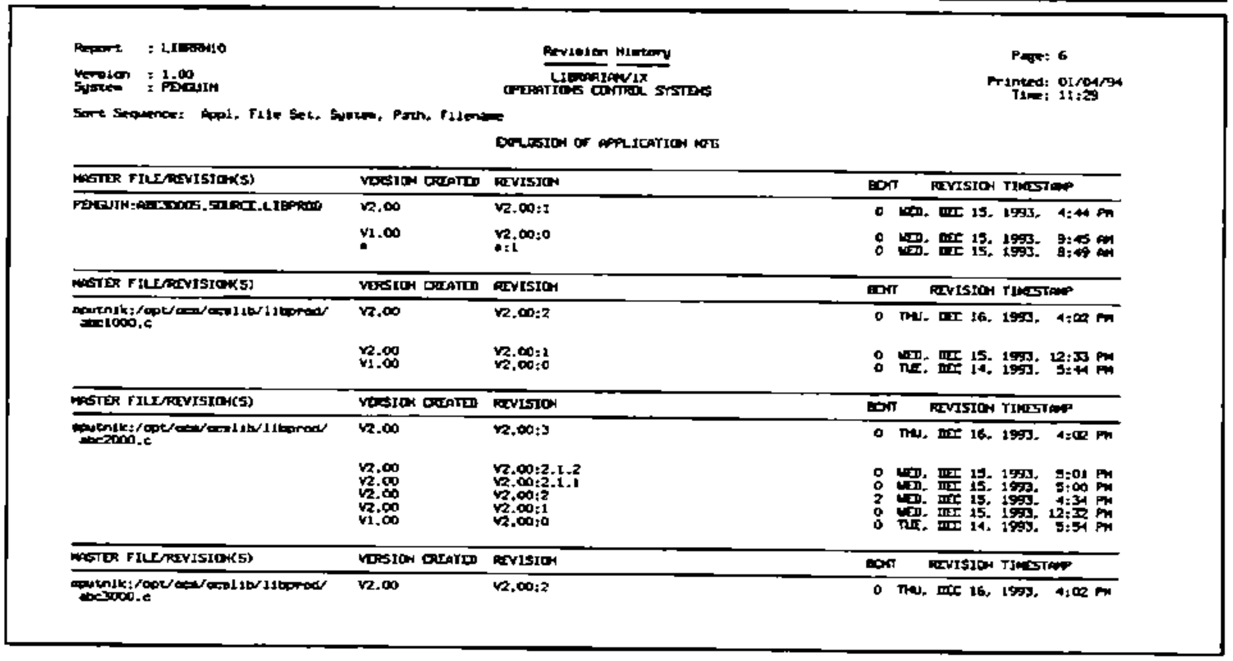
REVISION HISTORY REPORT RRH10

Sort Sequence	Application, Fileset, System, Path, Filename
Files Accessed	D-FILE
When to Run	On demand
Menu Access	**Info...Versions...RRH10 Revision History**
Description	Shows the revisions associated with each master file, including the version created, the revision number, the number of revisions, and the revision timestamp.

Sample



```
Report    : LIBRRH10                          Revision History                          Page: 3
Version   : 1.00                              LIBRARIAN/iX                     Printed: 01/04/94
System    : PENGUIN                     OPERATIONS CONTROL SYSTEMS                Time: 11:29
Sort Sequence:  Appl. File Set, System, Path, Filename

                                        EXPLOSION OF APPLICATION MFG

MASTER FILE/REVISION(S)          VERSION CREATED  REVISION                   RCNT   REVISION TIMESTAMP
PENGUIN;RPTSOOP.OBJECT.LIBPROD   V1.00            V2.00;0                     0  WED. DEC 15, 1993,  8:49 AM
No revisions for this file.

MASTER FILE/REVISION(S)          VERSION CREATED  REVISION                   RCNT   REVISION TIMESTAMP
PENGUIN;ABC1000S.SOURCE.LIBPROD  V2.00            V2.00:1                     0  WED. DEC 15, 1993,  4:44 PM

                                 V1.00            V2.00:0                     0  WED. DEC 15, 1993,  9:45 AM
                                 *                *:1                         0  WED. DEC 15, 1993,  8:49 AM

MASTER FILE/REVISION(S)          VERSION CREATED  REVISION                   RCNT   REVISION TIMESTAMP
PENGUIN;ABC2000S.SOURCE.LIBPROD  V2.00            V2.00:2                     0  WED. DEC 15, 1993,  4:44 PM

                                 V2.00            V2.00:1.1.1                 0  WED. DEC 15, 1993,  5:08 PM
                                 V2.00            V2.00:1                     1  WED. DEC 15, 1993,  2:40 PM
                                 V1.00            V2.00;0                     0  WED. DEC 15, 1993,  2:39 PM
                                 *                *:1                         0  WED. DEC 15, 1993,  8:49 AM
```



```
Report    : LIBRRH10                          Revision History                          Page: 6
Version   : 1.00                              LIBRARIAN/iX                     Printed: 01/04/94
System    : PENGUIN                     OPERATIONS CONTROL SYSTEMS                Time: 11:29
Sort Sequence:  Appl. File Set, System, Path, Filename

                                        EXPLOSION OF APPLICATION MFG

MASTER FILE/REVISION(S)          VERSION CREATED  REVISION                   RCNT   REVISION TIMESTAMP
PENGUIN:ABC3000S.SOURCE.LIBPROD  V2.00            V2.00:1                     0  WED. DEC 15, 1993,  4:44 PM

                                 V1.00            V2.00;0                     0  WED. DEC 15, 1993,  9:45 AM
                                 *                *:1                         0  WED. DEC 15, 1993,  8:49 AM

MASTER FILE/REVISION(S)          VERSION CREATED  REVISION                   RCNT   REVISION TIMESTAMP
sputnik:/opt/ocs/ocslib/libprod/ V2.00            V2.00:2                     0  THU. DEC 16, 1993,  4:02 PM
 abc1000.c

                                 V2.00            V2.00:1                     0  WED. DEC 15, 1993, 12:33 PM
                                 V1.00            V2.00;0                     0  TUE. DEC 14, 1993,  5:44 PM

MASTER FILE/REVISION(S)          VERSION CREATED  REVISION                   RCNT   REVISION TIMESTAMP
sputnik:/opt/ocs/ocslib/libprod/ V2.00            V2.00:3                     0  THU. DEC 16, 1993,  4:02 PM
 abc2000.c

                                 V2.00            V2.00:2.1.2                 0  WED. DEC 15, 1993,  5:01 PM
                                 V2.00            V2.00:2.1.1                 0  WED. DEC 15, 1993,  5:00 PM
                                 V2.00            V2.00:2                     2  WED. DEC 15, 1993,  4:34 PM
                                 V2.00            V2.00:1                     0  WED. DEC 15, 1993, 12:32 PM
                                 V1.00            V2.00;0                     0  TUE. DEC 14, 1993,  5:54 PM

MASTER FILE/REVISION(S)          VERSION CREATED  REVISION                   RCNT   REVISION TIMESTAMP
sputnik:/opt/ocs/ocslib/libprod/ V2.00            V2.00:2                     0  THU. DEC 16, 1993,  4:02 PM
 abc3000.c
```

Field Descriptions

Master File/Revision(s)	Name of the master file and associated revisions.
Version Created	Version when the master file was created.
Revision	Revision number.
BCNT	Number of branches associated with master file or revision.
Revision Timestamp	Timestamp associated with the revision.

TRANSACTION DETAIL REPORT RTD10

Sort Sequence Date, Time

Files Accessed D-SYSTEM-PROFILE D-TRANSDTL D-TRANSUM

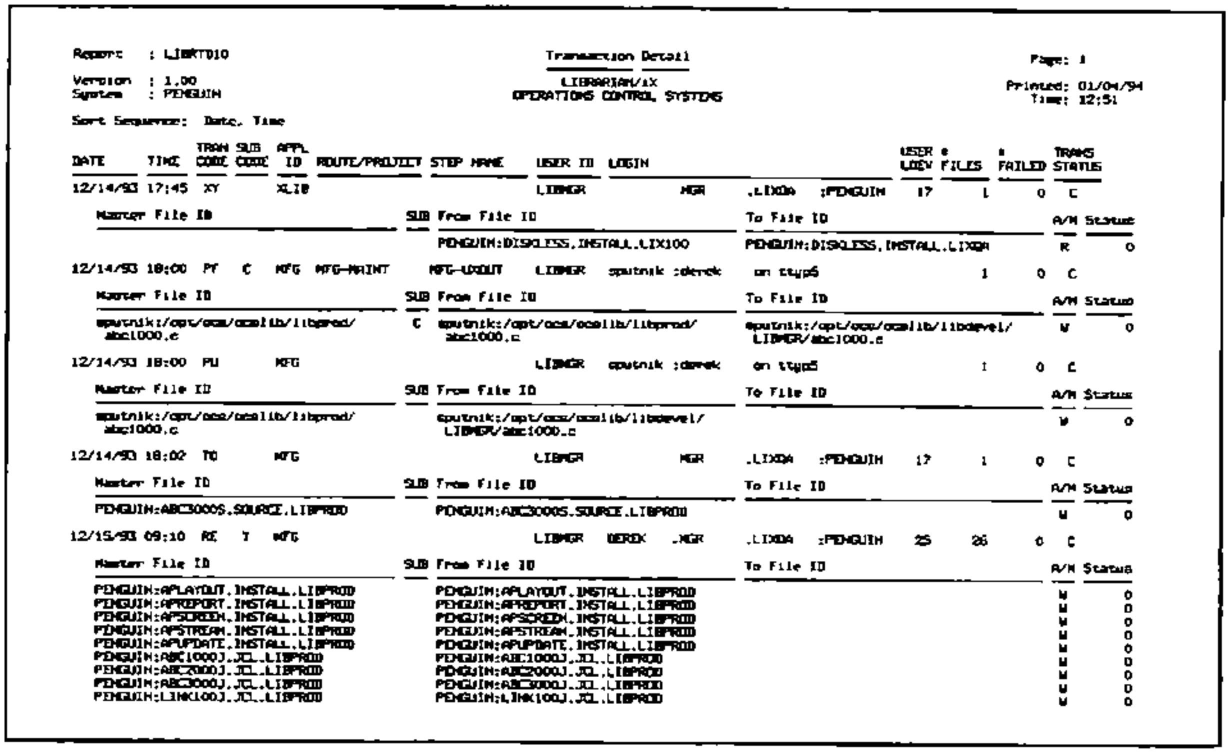
When to Run Prior to running the FLUSHLOG utility

Menu Access **Info...Log...RTD10 Transaction Detail**

Description Detailed information on transactions and corresponding file operations since the FLUSHLOG utility last purged audit trail records. FLUSHLOG uses the aging policy from the System Profile (SP) screen to determine which records to purge.

The information on this report includes type of transaction, project, step performed, user who issued the command, logical device, source and target locations, access mode/access control, and final status for each file processed. For each transaction, the report shows the number of files involved, the number of file movements that failed, and the overall transaction status.

Sample



```
Report    : LIBRTD10                                   Transaction Detail                                         Page: 1

Version   : 1.00                                          LIBRARIAN/iX                                         Printed: 01/04/94
System    : PENGUIN                                 OPERATIONS CONTROL SYSTEMS                                    Time: 12:51

Sort Sequence:  Date, Time

                TRAN SUB  APPL                                                                     USER #     #       TRANS
DATE      TIME  CODE CODE  ID  ROUTE/PROJECT STEP NAME   USER ID  LOGIN                            LDEV FILES FAILED STATUS
12/14/93 17:45  XY         XLIB                          LIBMGR             MGR   .LIXDA   :PENGUIN  17      1      0  C

   Master File ID                          SUB From File ID                        To File ID                      A/M Status

                                               PENGUIN:DISCLESS.INSTALL.LIX100     PENGUIN:DISCLESS.INSTALL.LIXDA   R       0
12/14/93 18:00  PT    C    MFG  MFG-MAINT     MFG-CKOUT   LIBMGR   sputnik :derek   on ttyp5                1      0  C

   Master File ID                          SUB From File ID                        To File ID                      A/M Status

   sputnik:/opt/ocs/ocslib/libprod/         C  sputnik:/opt/ocs/ocslib/libprod/    sputnik:/opt/ocs/ocslib/libdevel/ W       0
    abc1000.c                                   abc1000.c                           LIBMGR/abc1000.c
12/14/93 18:00  PU         MFG                            LIBMGR   sputnik :derek   on ttyp5                1      0  C

   Master File ID                          SUB From File ID                        To File ID                      A/M Status

   sputnik:/opt/ocs/ocslib/libprod/            sputnik:/opt/ocs/ocslib/libdevel/                                    W       0
    abc1000.c                                   LIBMGR/abc1000.c
12/14/93 18:02  TO         MFG                            LIBMGR             MGR   .LIXDA   :PENGUIN  17      1      0  C

   Master File ID                          SUB From File ID                        To File ID                      A/M Status

   PENGUIN:ABC3000S.SOURCE.LIBPROD             PENGUIN:ABC3000S.SOURCE.LIBPROD                                      W       0
12/15/93 09:10  RE    T    MFG                            LIBMGR   DEREK   .MGR   .LIXDA   :PENGUIN  25     26      0  C

   Master File ID                          SUB From File ID                        To File ID                      A/M Status

   PENGUIN:APLAYOUT.INSTALL.LIBPROD            PENGUIN:APLAYOUT.INSTALL.LIBPROD                                     W       0
   PENGUIN:APREPORT.INSTALL.LIBPROD            PENGUIN:APREPORT.INSTALL.LIBPROD                                     W       0
   PENGUIN:APSCREEN.INSTALL.LIBPROD            PENGUIN:APSCREEN.INSTALL.LIBPROD                                     W       0
   PENGUIN:APSTREAM.INSTALL.LIBPROD            PENGUIN:APSTREAM.INSTALL.LIBPROD                                     W       0
   PENGUIN:APUPDATE.INSTALL.LIBPROD            PENGUIN:APUPDATE.INSTALL.LIBPROD                                     W       0
   PENGUIN:ABC1000J.JCL.LIBPROD                PENGUIN:ABC1000J.JCL.LIBPROD                                         W       0
   PENGUIN:ABC2000J.JCL.LIBPROD                PENGUIN:ABC2000J.JCL.LIBPROD                                         W       0
   PENGUIN:ABC3000J.JCL.LIBPROD                PENGUIN:ABC3000J.JCL.LIBPROD                                         W       0
   PENGUIN:LINK100J.JCL.LIBPROD                PENGUIN:LINK100J.JCL.LIBPROD                                         W       0
```

TRANSACTION DETAIL REPORT (continued) RTD10

Field Descriptions

Date	The date the transaction was performed.
Time	The time the transaction was performed.
Tran Code	The code for the type of transaction. For a listing of transaction codes, refer to *Transaction Codes in Reports* at the beginning of this chapter.
Sub Code	The subcode for the transaction. For a listing of transaction subcodes, refer to *Transaction Codes in Reports* at the beginning of this chapter.
Appl ID	The application ID.
Route/Project	The name of the route/project.
Step Name	The name of the step performed.
User ID	The user who issued the command.
Login	The MPE/UNIX login.
User LDEV	The logical device.
# Files	The number of files involved in the transaction.
# Failed	The number of file movements that failed.
Trans Status	The overall status of the transaction (C=Complete, F=Failed).
Master File ID	Name of the master file associated with the transaction.
Sub	Subcode for the transaction. For a listing of transaction subcodes, refer to *Transaction Codes in Reports* at the beginning of this chapter.
From File ID	Source location for the transaction.
To File ID	Target location for the transaction.
A/M	Access mode (read or write).
Status	Status for the file.

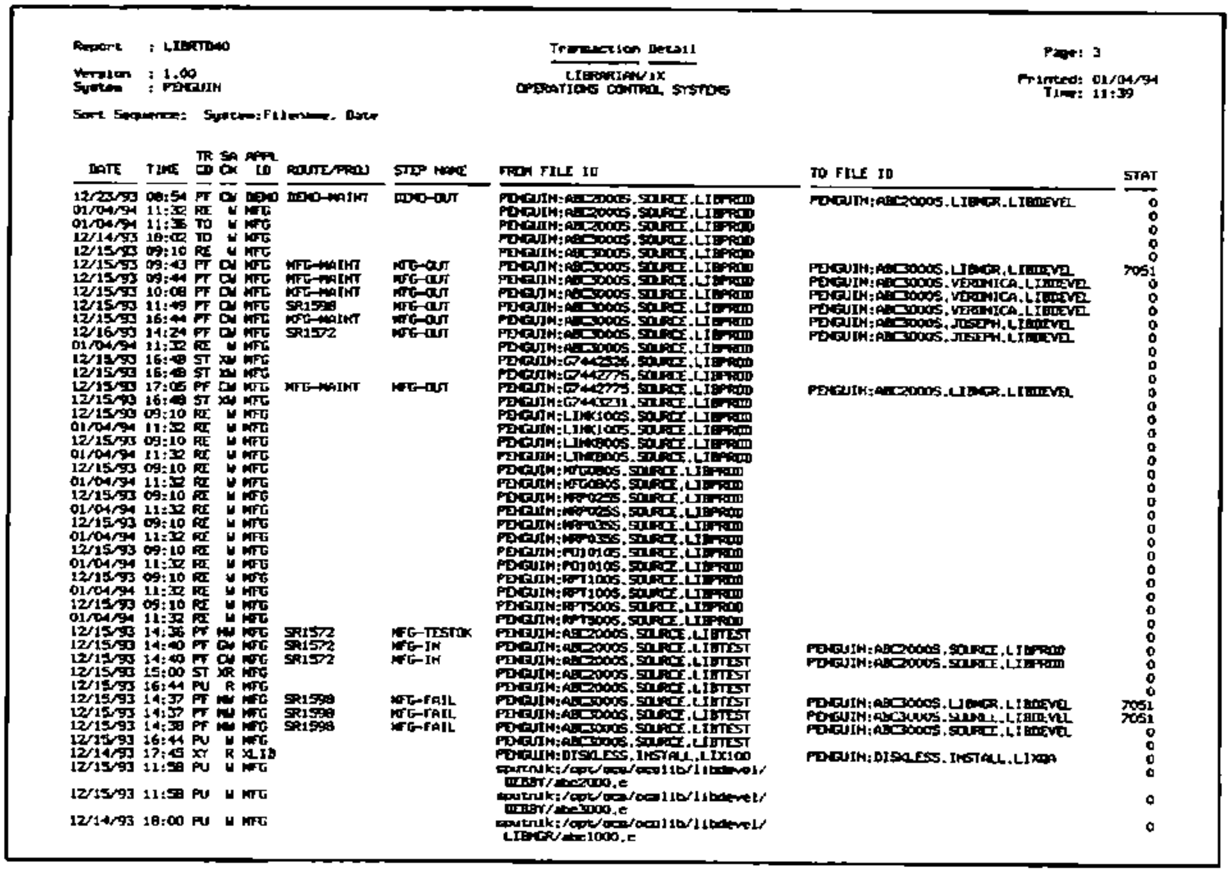
TRANSACTION DETAIL REPORT **RTD40**

Sort Sequence	System, Filename, Date
Files Accessed	D-SYSTEM-PROFILE D-TRANSDTL D-TRANSUM
When to Run	Prior to running the FLUSHLOG utility
Menu Access	**Info...Log...RTD40 Transaction Detail**
Description	Detailed information on transactions and corresponding file operations since the FLUSHLOG utility last purged audit trail records. FLUSHLOG uses the aging policy from the System Profile (SP) screen to determine which records to purge.

From SHOWLOG this report provides two different formats depending on the type of sort sequence selected. The default sort is by master filename, resulting in a column for Master Filename and no Time column. If Date is the sort sequence, the format contains a column for Time as well as a Date column. This sort sequence is shown in the sample report below.

The information on the report includes type of transaction, step performed, source and target locations, access mode of the file, and final status for each file in the transaction.

Sample



```
Report   : LIBRTD40                              Transaction Detail                                         Page: 3

Version  : 1.00                                     LIBRARIAN/iX                                   Printed: 01/04/94
System   : PENGUIN                            OPERATIONS CONTROL SYSTEMS                              Time: 11:39

Sort Sequence:  System:Filename, Date

                   TR SA APPL
  DATE     TIME    CD CK   ID  ROUTE/PROJ   STEP NAME    FROM FILE ID                         TO FILE ID                                STAT
________   _____   __ __ ____  __________   __________   __________________________________   _______________________________________   ____
12/23/93   08:54   PF CW DEMO  DEMO-MAINT   DEMO-OUT     PENGUIN:ABC2000S.SOURCE.LIBPROD      PENGUIN:ABC2000S.LIBMGR.LIBDEVEL             0
01/04/94   11:32   RE  W MFG                             PENGUIN:ABC2000S.SOURCE.LIBPROD                                                   0
01/04/94   11:36   TO  W MFG                             PENGUIN:ABC2000S.SOURCE.LIBPROD                                                   0
12/14/93   18:02   TO  W MFG                             PENGUIN:ABC3000S.SOURCE.LIBPROD                                                   0
12/15/93   09:10   RE  W MFG                             PENGUIN:ABC3000S.SOURCE.LIBPROD                                                   0
12/15/93   09:43   PF CW MFG   MFG-MAINT    MFG-OUT      PENGUIN:ABC3000S.SOURCE.LIBPROD      PENGUIN:ABC3000S.LIBMGR.LIBDEVEL          7051
12/15/93   09:44   PF CW MFG   MFG-MAINT    MFG-OUT      PENGUIN:ABC3000S.SOURCE.LIBPROD      PENGUIN:ABC3000S.VERONICA.LIBDEVEL           0
12/15/93   10:08   PF CW MFG   MFG-MAINT    MFG-OUT      PENGUIN:ABC3000S.SOURCE.LIBPROD      PENGUIN:ABC3000S.VERONICA.LIBDEVEL           0
12/15/93   11:49   PF CW MFG   SR1598       MFG-OUT      PENGUIN:ABC3000S.SOURCE.LIBPROD      PENGUIN:ABC3000S.VERONICA.LIBDEVEL           0
12/15/93   16:44   PF CW MFG   MFG-MAINT    MFG-OUT      PENGUIN:ABC3000S.SOURCE.LIBPROD      PENGUIN:ABC3000S.JOSEPH.LIBDEVEL             0
12/16/93   14:24   PF CW MFG   SR1572       MFG-OUT      PENGUIN:ABC3000S.SOURCE.LIBPROD      PENGUIN:ABC3000S.JOSEPH.LIBDEVEL             0
01/04/94   11:32   RE  W MFG                             PENGUIN:ABC3000S.SOURCE.LIBPROD                                                   0
12/15/93   16:48   ST XW MFG                             PENGUIN:G7442526.SOURCE.LIBPROD                                                   0
12/15/93   16:48   ST XW MFG                             PENGUIN:G7442775.SOURCE.LIBPROD                                                   0
12/15/93   17:05   PF CW MFG   MFG-MAINT    MFG-OUT      PENGUIN:G7442775.SOURCE.LIBPROD      PENGUIN:ABC2000S.LIBMGR.LIBDEVEL             0
12/15/93   16:48   ST XW MFG                             PENGUIN:G7443231.SOURCE.LIBPROD                                                   0
12/15/93   09:10   RE  W MFG                             PENGUIN:LINK100S.SOURCE.LIBPROD                                                   0
01/04/94   11:32   RE  W MFG                             PENGUIN:LINK100S.SOURCE.LIBPROD                                                   0
12/15/93   09:10   RE  W MFG                             PENGUIN:LINK800S.SOURCE.LIBPROD                                                   0
01/04/94   11:32   RE  W MFG                             PENGUIN:LINK800S.SOURCE.LIBPROD                                                   0
12/15/93   09:10   RE  W MFG                             PENGUIN:MFG080S.SOURCE.LIBPROD                                                    0
01/04/94   11:32   RE  W MFG                             PENGUIN:MFG080S.SOURCE.LIBPROD                                                    0
12/15/93   09:10   RE  W MFG                             PENGUIN:MRP025S.SOURCE.LIBPROD                                                    0
01/04/94   11:32   RE  W MFG                             PENGUIN:MRP025S.SOURCE.LIBPROD                                                    0
12/15/93   09:10   RE  W MFG                             PENGUIN:MRP035S.SOURCE.LIBPROD                                                    0
01/04/94   11:32   RE  W MFG                             PENGUIN:MRP035S.SOURCE.LIBPROD                                                    0
12/15/93   09:10   RE  W MFG                             PENGUIN:PO1010S.SOURCE.LIBPROD                                                    0
01/04/94   11:32   RE  W MFG                             PENGUIN:PO1010S.SOURCE.LIBPROD                                                    0
12/15/93   09:10   RE  W MFG                             PENGUIN:RPT100S.SOURCE.LIBPROD                                                    0
01/04/94   11:32   RE  W MFG                             PENGUIN:RPT100S.SOURCE.LIBPROD                                                    0
12/15/93   09:10   RE  W MFG                             PENGUIN:RPT500S.SOURCE.LIBPROD                                                    0
01/04/94   11:32   RE  W MFG                             PENGUIN:RPT500S.SOURCE.LIBPROD                                                    0
12/15/93   14:36   PF NW MFG   SR1572       MFG-TESTOK   PENGUIN:ABC2000S.SOURCE.LIBTEST                                                   0
12/15/93   14:40   PF CW MFG   SR1572       MFG-IN       PENGUIN:ABC2000S.SOURCE.LIBTEST      PENGUIN:ABC2000S.SOURCE.LIBPROD              0
12/15/93   14:40   PF CW MFG   SR1572       MFG-IN       PENGUIN:ABC2000S.SOURCE.LIBTEST      PENGUIN:ABC2000S.SOURCE.LIBPROD              0
12/15/93   15:00   ST XR MFG                             PENGUIN:ABC2000S.SOURCE.LIBTEST                                                   0
12/15/93   16:44   PU  R MFG                             PENGUIN:ABC2000S.SOURCE.LIBTEST                                                   0
12/15/93   14:37   PF NW MFG   SR1598       MFG-FAIL     PENGUIN:ABC3000S.SOURCE.LIBTEST      PENGUIN:ABC3000S.LIBMGR.LIBDEVEL          7051
12/15/93   14:37   PF NW MFG   SR1598       MFG-FAIL     PENGUIN:ABC3000S.SOURCE.LIBTEST      PENGUIN:ABC3000S.SOURCE.LIBDEVEL          7051
12/15/93   14:38   PF NW MFG   SR1598       MFG-FAIL     PENGUIN:ABC3000S.SOURCE.LIBTEST      PENGUIN:ABC3000S.SOURCE.LIBDEVEL             0
12/15/93   16:44   PU  W MFG                             PENGUIN:ABC3000S.SOURCE.LIBTEST                                                   0
12/14/93   17:45   XY  R XLIB                            PENGUIN:DISKLESS.INSTALL.LIX100      PENGUIN:DISKLESS.INSTALL.LIXGA               0
12/15/93   11:58   PU  W MFG                             sputnik:/opt/oca/ocalib/libdevel/                                                 0
                                                          DEBBY/abc2000.c
12/15/93   11:58   PU  W MFG                             sputnik:/opt/oca/ocalib/libdevel/                                                 0
                                                          DEBBY/abc3000.c
12/14/93   18:00   PU  W MFG                             sputnik:/opt/oca/ocalib/libdevel/                                                 0
                                                          LIBMGR/abc1000.c
```

Field Descriptions

Date	Date the transaction was performed.
Time	Time the transaction was performed.
TR/CD	Transaction code. For a listing of transaction codes, refer to *Transaction Codes in Reports* at the beginning of this chapter.
A/M	Access mode (W = write, R = read). For a listing of transaction subcodes, refer to *Transaction Codes in Reports* at the beginning of this chapter.
Appl ID	Application ID.
Route/Proj	Route/project name.
Step Name	Name of the step performed.
From File ID	Source location.
To File ID	Destination location, if applicable.
Status	The status for the file.

0	Completed successfully
6000–6500	CIERR. Error is STATUS–6000
7000–7500	FSERR. Code is STATUS–7000
7501–8500	IMAGE Error. Error is STATUS–8000
8501–9998	Internal OCS/LIBRARIAN error
9999	System failure

TRANSACTION SUMMARY REPORT RTS10

Sort Sequence Date, Time

Files Accessed D-MEMO
D-SYSTEM-PROFILE
D-TRANSUM

When to Run Prior to running the FLUSHLOG utility

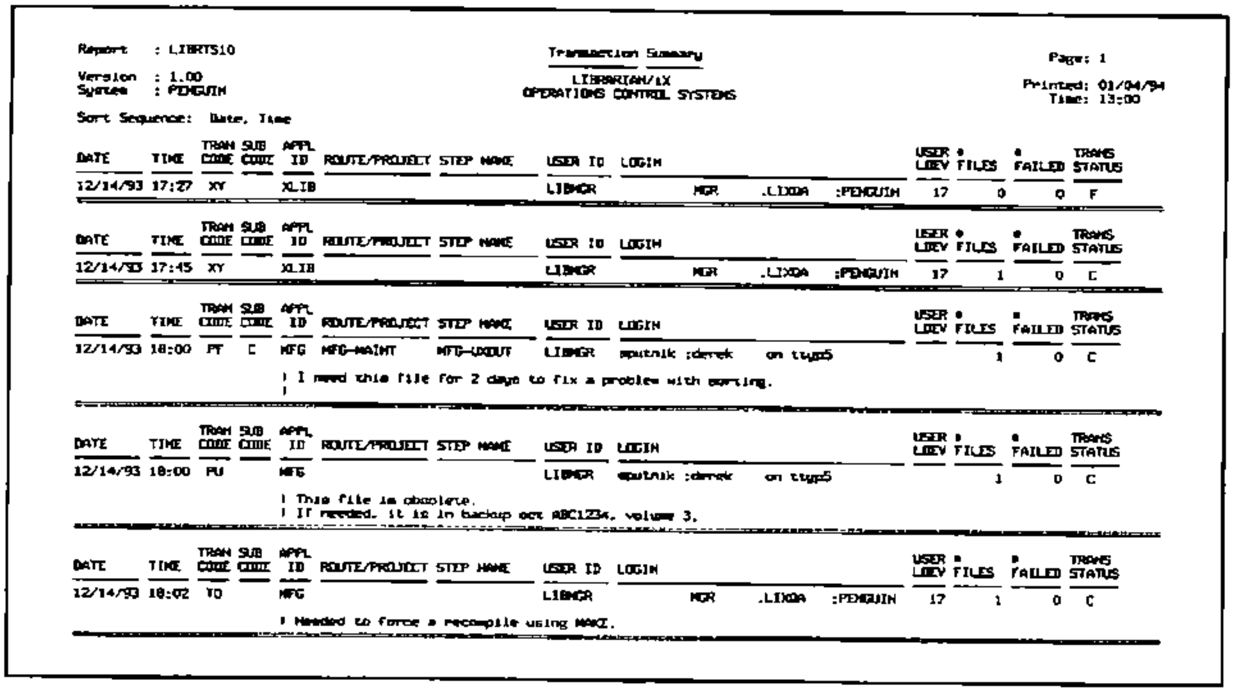
Menu Access Info...Log...RTS10 Transaction Summary

Description This report produces summaries of transactions since the FLUSHLOG utility last purged audit trail records. The FLUSHLOG utility uses the aging policy from the System Profile (SP) screen to determine which records to purge.

This report provides an overview of the transactions that occurred. However, it does not report the status of individual files within each transaction.

This report has the option of displaying the memo text entered for each transaction with the optional **MEMO** command parameter. If you want the text listed, respond with the letter Y to the prompt.

Sample



```
Report   : LIBRTS10                                Transaction Summary                                          Page: 1

Version  : 1.00                                       LIBRARIAN/iX                                        Printed: 01/04/94
System   : PENGUIN                              OPERATIONS CONTROL SYSTEMS                                   Time: 13:00

Sort Sequence:  Date, Time

               TRAN SUB  APPL                                                                      USER #          #      TRANS
DATE     TIME  CODE CODE  ID  ROUTE/PROJECT STEP NAME  USER ID  LOGIN                               LDEV FILES  FAILED  STATUS
12/14/93 17:27  XY       XLIB                          LIBMGR             MGR     .LIXDA   :PENGUIN  17      0       0   F

               TRAN SUB  APPL                                                                      USER #          #      TRANS
DATE     TIME  CODE CODE  ID  ROUTE/PROJECT STEP NAME  USER ID  LOGIN                               LDEV FILES  FAILED  STATUS
12/14/93 17:45  XY       XLIB                          LIBMGR             MGR     .LIXDA   :PENGUIN  17      1       0   C

               TRAN SUB  APPL                                                                      USER #          #      TRANS
DATE     TIME  CODE CODE  ID  ROUTE/PROJECT STEP NAME  USER ID  LOGIN                               LDEV FILES  FAILED  STATUS
12/14/93 18:00  PT   C   MFG  MFG-MAINT     MFG-UXOUT  LIBMGR   sputnik :derek   on ttyp5                    1       0   C
                         | I need this file for 2 days to fix a problem with sorting.
                         |

               TRAN SUB  APPL                                                                      USER #          #      TRANS
DATE     TIME  CODE CODE  ID  ROUTE/PROJECT STEP NAME  USER ID  LOGIN                               LDEV FILES  FAILED  STATUS
12/14/93 18:00  PU       MFG                           LIBMGR   sputnik :derek   on ttyp5                    1       0   C
                         | This file is obsolete.
                         | If needed, it is in backup set ABC1234, volume 3.

               TRAN SUB  APPL                                                                      USER #          #      TRANS
DATE     TIME  CODE CODE  ID  ROUTE/PROJECT STEP NAME  USER ID  LOGIN                               LDEV FILES  FAILED  STATUS
12/14/93 18:02  TO       MFG                           LIBMGR             MGR     .LIXDA   :PENGUIN  17      1       0   C
                         | Needed to force a recompile using MAKE.
```

TRANSACTION SUMMARY REPORT (continued) RTS10

Field Descriptions

Date	Date the transaction was performed.
Time	Time the transaction was performed.
Tran Code	Transaction code. For a list of transaction codes that appear on this report, refer to *Transaction Codes in Reports* at the beginning of this chapter.
Sub Code	Subcode for the transaction. For a listing of transaction subcodes, refer to *Transaction Codes in Reports* at the beginning of this chapter.
Appl ID	Application ID.
Route/Project	Name of the route/project.
Step Name	Name of the step performed.
User ID	The user who issued the command.
Login	The MPE/UNIX login.
User LDEV	The logical device.
# Files	Number of files involved in the transaction.
# Failed	Number of file movements that failed.
Trans Status	Status of the transaction: C Completed F Failed to complete

USERS REPORT RUD10

Sort Sequence

User ID

Files Accessed

D-SYSTEM-PROFILE
D-USER-CAPS
M-USER

When to Run

On demand

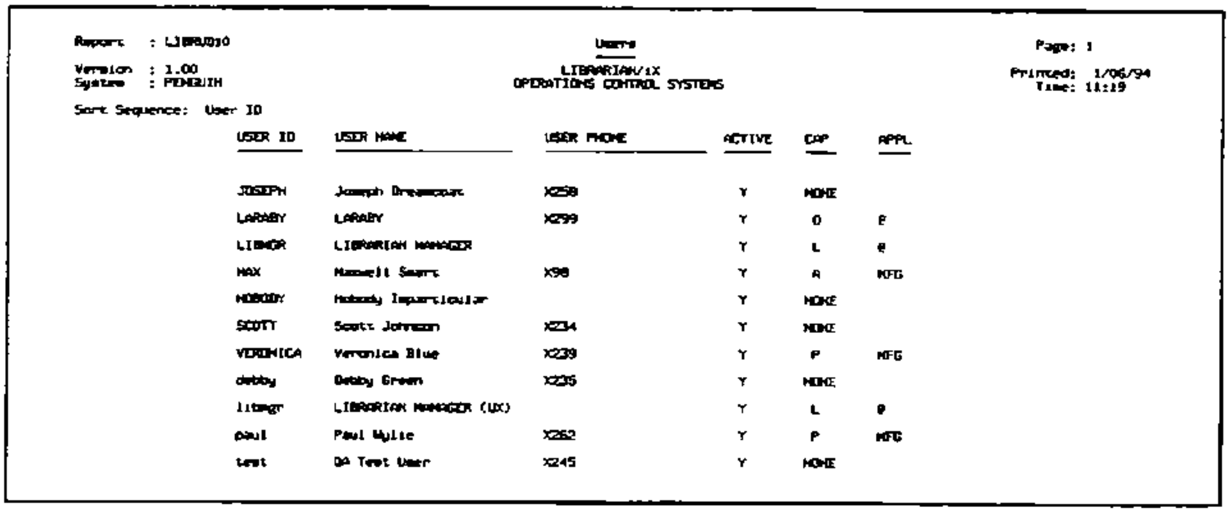
Menu Access

Info...Rules...RUD10 Users

Description

This report lists LIBRARIAN users. For each user, their capabilities and current status are shown. For users with Application or Project Manager capability, the associated applications are listed in addition to user information.

Sample



```
Report   : LIBRUD10                          Users                                    Page: 1

Version  : 1.00                          LIBRARIAN/iX                          Printed:  1/06/94
System   : PENGUIN                 OPERATIONS CONTROL SYSTEMS                     Time: 11:19

Sort Sequence:  User ID
                        USER ID    USER NAME                 USER PHONE        ACTIVE   CAP    APPL.

                        JOSEPH     Joseph Dreamcoat          X258                Y      NONE
                        LARABY     LARABY                    X299                Y       O     *
                        LIBMGR     LIBRARIAN MANAGER                             Y       L     *
                        MAX        Maxwell Smart             X98                 Y       A     MFG
                        NOBODY     Nobody Imparticular                           Y      NONE
                        SCOTT      Scott Johnson             X234                Y      NONE
                        VERONICA   Veronica Blue             X239                Y       P     MFG
                        debby      Debby Green               X235                Y      NONE
                        libmgr     LIBRARIAN MANAGER (UX)                        Y       L     *
                        paul       Paul Wylie                X262                Y       P     MFG
                        test       DA Test User              X245                Y      NONE
```

Field Descriptions

User ID	Unique identifier of the LIBRARIAN user.
User Name	Full name of the user.
User Phone	Phone number of the user.
Active	Indicates whether the user is currently an active user (Y=active user, N=inactive user).

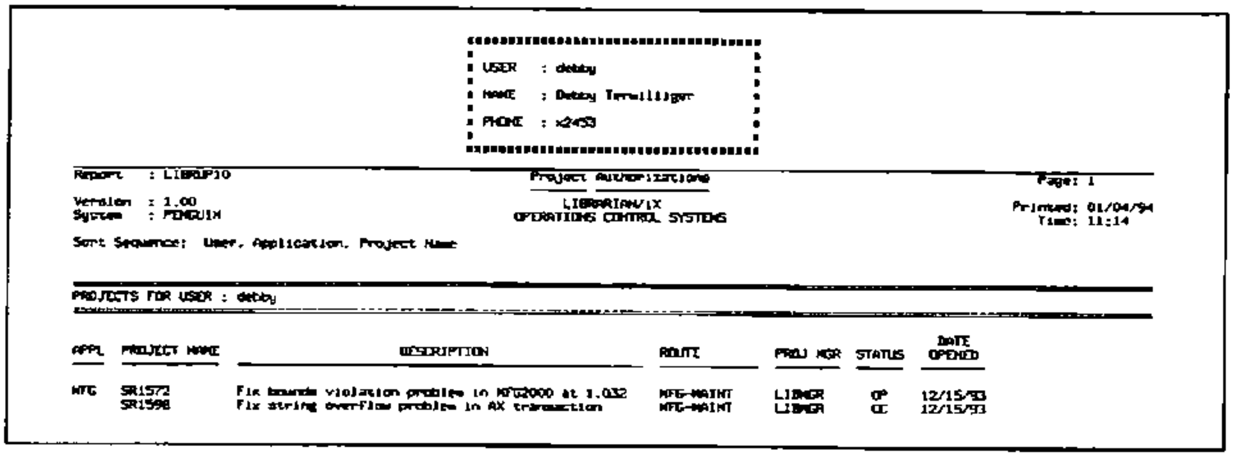
Field Descriptions, continued

Cap	The user's capabilities:
	L — LIBRARIAN manager
	A — Application manager
	R — Rule administrator
	P — Project manager
	O — Operator
	X — Unrestricted X command access
	NONE — No capabilities
Appl	The application for which the user has the special capability.

PROJECT AUTHORIZATIONS REPORT RUP10

Sort Sequence	User, Application, Project Name
Files Accessed	D-PROJECTS D-SYSTEM-PROFILE M-USER
When to Run	On demand
Menu Access	**Info...Rules...RUP10 Project Authorizations**
Description	Lists project information for each user. The information includes the project, the project manager, the status, and the date the project was opened.

Sample



```
                                   ********************************
                                   *                              *
                                   * USER   : debby               *
                                   *                              *
                                   * NAME   : Debby Terwilliger   *
                                   *                              *
                                   * PHONE  : x2453               *
                                   *                              *
                                   ********************************
Report   : LIBRUP10                     Project Authorizations                         Page: 1

Version  : 1.00                           LIBRARIAN/iX                       Printed: 01/04/94
System   : PENGUIN                  OPERATIONS CONTROL SYSTEMS                  Time: 11:14

Sort Sequence:  User, Application, Project Name

PROJECTS FOR USER : debby

                                                                                       DATE
APPL  PROJECT NAME              DESCRIPTION                  ROUTE      PROJ MGR STATUS OPENED

MFG   SR1572       Fix bounds violation problem in MFG2000 at 1.032  MFG-MAINT  LIBMGR  OP  12/15/93
      SR1598       Fix string overflow problem in AX transaction     MFG-MAINT  LIBMGR  DC  12/15/93
```

Field Descriptions

Appl	Application for which the user is authorized.
Project Name	Name of the projects for which the user is authorized.
Description	A short description of the project.
Route	Name of the route corresponding to the project.
Proj Mgr	Name of the project manager.
Status	Initial status of the project (OP=open, DC=documented).
Date Opened	Date the project was opened.

STEP AUTHORIZATIONS REPORT RUS10

Sort Sequence Application, Route, Sort Number, Step, User

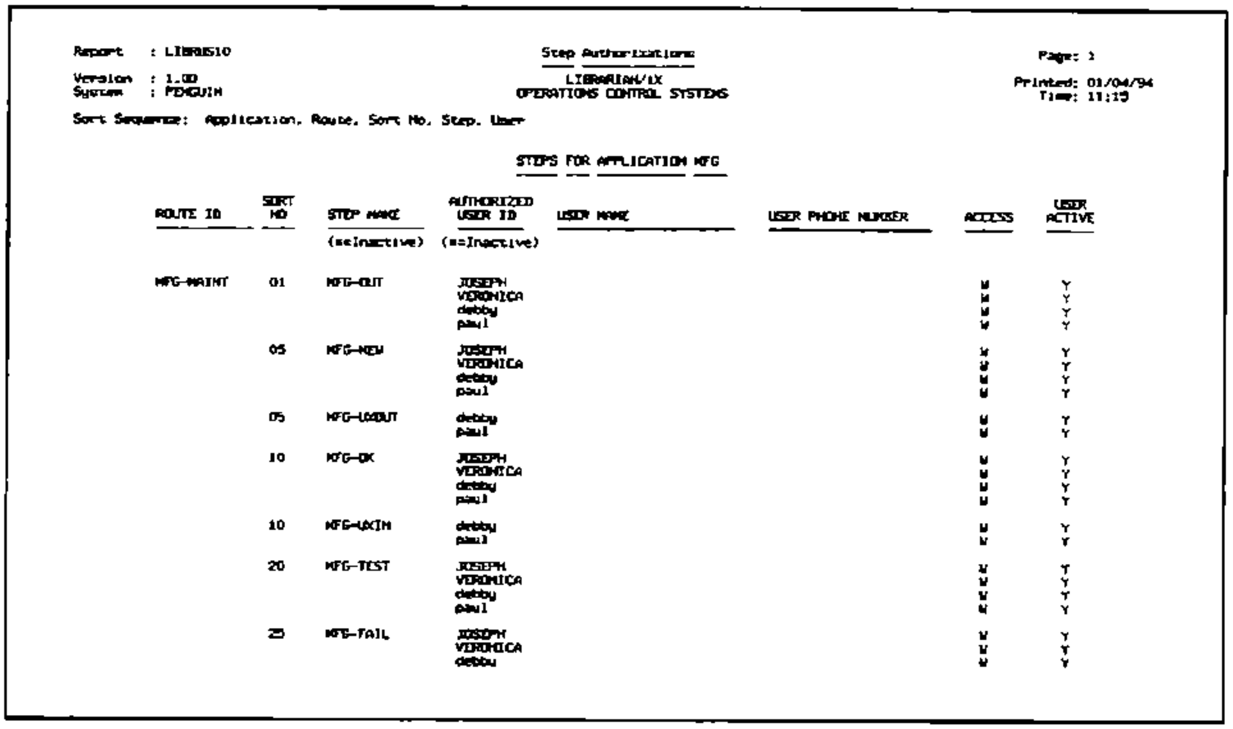
Files Accessed D-STEP
M-USER
D-USER-STEP
D-SYSTEM-PROFILE

When to Run On demand

Menu Access **Info...Rules...RUS10 Step Authorizations**

Description Lists the authorization information for all defined steps. This report provides detailed information about the step including the authorized users and the access mode available for each user. In addition, steps without authorized users are listed. Inactive steps and users are identified.

Sample



```
Report   : LIBRUS10                          Step Authorizations                         Page: 1
Version  : 1.00                                LIBRARIAN/iX                     Printed: 01/04/94
System   : PENGUIN                       OPERATIONS CONTROL SYSTEMS                 Time: 11:15
Sort Sequence:  Application, Route, Sort No, Step, User

                                          STEPS FOR APPLICATION MFG

                  SORT                  AUTHORIZED
      ROUTE ID     NO    STEP NAME       USER ID    USER NAME          USER PHONE NUMBER    ACCESS   USER
                                                                                                     ACTIVE
                         (*=Inactive)  (*=Inactive)

      MFG-MAINT    01    MFG-OUT         JOSEPH                                               W        Y
                                         VERONICA                                             W        Y
                                         debby                                                W        Y
                                         paul                                                 W        Y

                   05    MFG-NEW         JOSEPH                                               W        Y
                                         VERONICA                                             W        Y
                                         debby                                                W        Y
                                         paul                                                 W        Y

                   05    MFG-LOGOUT      debby                                                W        Y
                                         paul                                                 W        Y

                   10    MFG-OK          JOSEPH                                               W        Y
                                         VERONICA                                             W        Y
                                         debby                                                W        Y
                                         paul                                                 W        Y

                   10    MFG-LOGIN       debby                                                W        Y
                                         paul                                                 W        Y

                   20    MFG-TEST        JOSEPH                                               W        Y
                                         VERONICA                                             W        Y
                                         debby                                                W        Y
                                         paul                                                 W        Y

                   25    MFG-FAIL        JOSEPH                                               W        Y
                                         VERONICA                                             W        Y
                                         debby                                                W        Y
```

Field Descriptions

Route ID	Unique identifier of the route.
Sort No	Number of the step within the route.
Step Name	Name of the step.

STEP AUTHORIZATIONS REPORT (continued) RUS10

Field Descriptions (continued)

Authorized User ID	User ID of persons authorized to use the route.
User Name	Full name of authorized persons.
User Phone Number	Phone number of the user.
Access	Type of access (W=write, R=read).
User Active	Indicates whether the user is active.

FILE VERSIONS REPORT RVD10

Sort Sequence Version ID, File ID

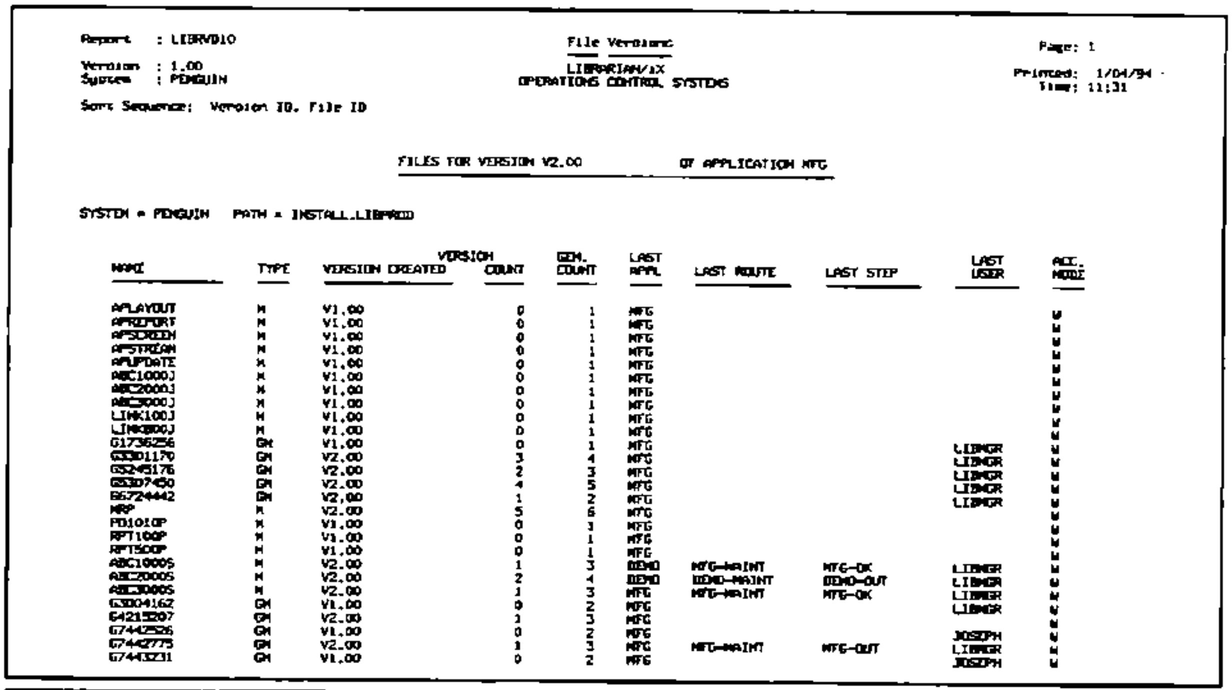
Files Accessed D-FILE

When to Run On demand

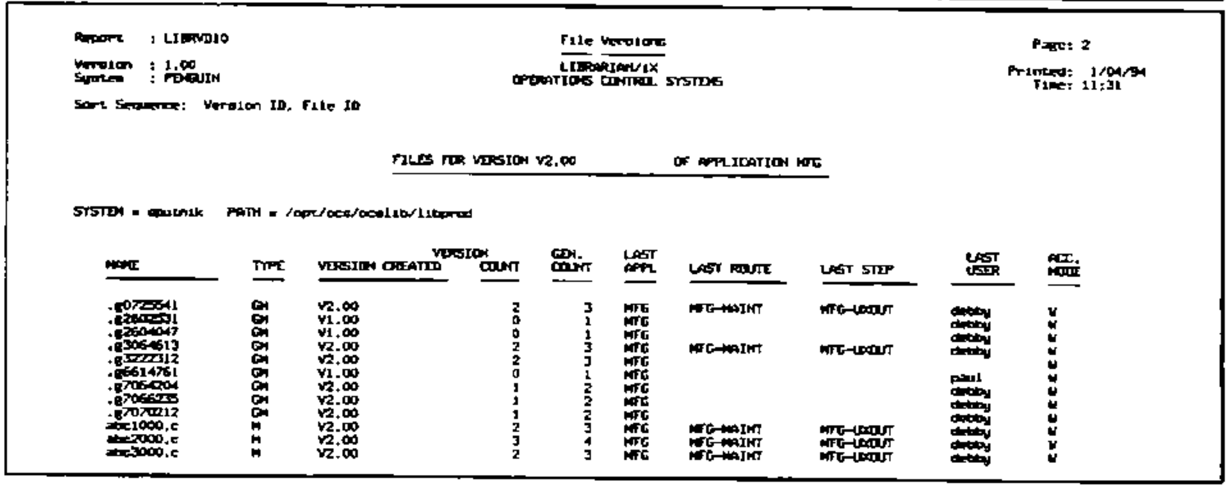
Menu Access **Info...Versions...RVD10 File Versions**

Description Identifies all files in a version. The information for each file includes the version ID, version count, generation count, the date and time of last modification, the user who performed the last step, and the access mode of the file.

Sample



```
Report   : LIBRVD10                      File Versions                              Page: 1

Version  : 1.00                          LIBRARIAN/iX                          Printed:  1/04/94
System   : PENGUIN                 OPERATIONS CONTROL SYSTEMS                     Time: 11:31

Sort Sequence:  Version ID, File ID


                       FILES FOR VERSION V2.00          OF APPLICATION MFG


SYSTEM = PENGUIN   PATH = INSTALL.LIBPROD

                                         VERSION   GEN.   LAST                                    LAST    ACC.
 NAME            TYPE   VERSION CREATED    COUNT   COUNT  APPL   LAST ROUTE    LAST STEP          USER    MODE

 APLAYOUT        M      V1.00                 0       1   MFG                                             W
 APREPORT        M      V1.00                 0       1   MFG                                             W
 APSCREEN        M      V1.00                 0       1   MFG                                             W
 APSTREAM        M      V1.00                 0       1   MFG                                             W
 APUPDATE        M      V1.00                 0       1   MFG                                             W
 ABC1000J        M      V1.00                 0       1   MFG                                             W
 ABC2000J        M      V1.00                 0       1   MFG                                             W
 ABC3000J        M      V1.00                 0       1   MFG                                             W
 LINK100J        M      V1.00                 0       1   MFG                                             W
 LINK800J        M      V1.00                 0       1   MFG                                             W
 G1736256        GM     V1.00                 0       1   MFG                                    LIBMGR   W
 G3301170        GM     V2.00                 3       4   MFG                                    LIBMGR   W
 G5245176        GM     V2.00                 2       3   MFG                                    LIBMGR   W
 G5307450        GM     V2.00                 4       5   MFG                                    LIBMGR   W
 B6724442        GM     V2.00                 1       2   MFG                                    LIBMGR   W
 MRP             M      V2.00                 5       6   MFG                                             W
 PD1010P         M      V1.00                 0       1   MFG                                             W
 RPT100P         M      V1.00                 0       1   MFG                                             W
 RPT500P         M      V1.00                 0       1   MFG                                             W
 ABC1000S        M      V2.00                 1       3   DEMO   MFG-MAINT     MFG-OK            LIBMGR   W
 ABC2000S        M      V2.00                 2       4   DEMO   DEMO-MAINT    DEMO-OUT          LIBMGR   W
 ABC3000S        M      V2.00                 1       3   MFG    MFG-MAINT     MFG-OK            LIBMGR   W
 G3004162        GM     V1.00                 0       2   MFG                                    LIBMGR   W
 G4215207        GM     V2.00                 1       3   MFG                                             W
 G7442526        GM     V1.00                 0       2   MFG                                    JOSEPH   W
 G7442775        GM     V2.00                 1       3   MFG    MFG-MAINT     MFG-OUT           LIBMGR   W
 G7443231        GM     V1.00                 0       2   MFG                                    JOSEPH   W
```



```
Report   : LIBRVD10                      File Versions                              Page: 2

Version  : 1.00                          LIBRARIAN/iX                          Printed:  1/04/94
System   : PENGUIN                 OPERATIONS CONTROL SYSTEMS                     Time: 11:31

Sort Sequence:  Version ID, File ID


                       FILES FOR VERSION V2.00          OF APPLICATION MFG


SYSTEM = sputnik   PATH = /opt/ocs/ocslib/libprod

                                         VERSION   GEN.   LAST                                    LAST    ACC.
 NAME            TYPE   VERSION CREATED    COUNT   COUNT  APPL   LAST ROUTE    LAST STEP          USER    MODE

 .g0725541       GM     V2.00                 2       3   MFG    MFG-MAINT     MFG-UXOUT         debby    W
 .g2842531       GM     V1.00                 0       1   MFG                                    debby    W
 .g2604047       GM     V1.00                 0       1   MFG                                    debby    W
 .g3064613       GM     V2.00                 2       3   MFG    MFG-MAINT     MFG-UXOUT         debby    W
 .g3222312       GM     V2.00                 2       3   MFG                                             W
 .g6614761       GM     V1.00                 0       1   MFG                                    paul     W
 .g7064204       GM     V2.00                 1       2   MFG                                    debby    W
 .g7066235       GM     V2.00                 1       2   MFG                                    debby    W
 .g7070212       GM     V2.00                 1       2   MFG                                    debby    W
 abc1000.c       M      V2.00                 2       3   MFG    MFG-MAINT     MFG-UXOUT         debby    W
 abc2000.c       M      V2.00                 3       4   MFG    MFG-MAINT     MFG-UXOUT         debby    W
 abc3000.c       M      V2.00                 2       3   MFG    MFG-MAINT     MFG-UXOUT         debby    W
```

Field Descriptions

Name	Name of file.
Type	Type of file: M Master S Secondary GM Generated master GS Generated secondary CM Copy of generated master CS Copy of generated secondary
Version Created	Version when the master file was first introduced.
Version Count	Version count (VCOUNT).
Gen Count	The generation count (GCOUNT).
Last Appl	The last application that processed the file.
Last Route	The last route that processed the file.
Last Step	The last step that processed the file.
Last User	The last user who processed the file.
Acc Mode	The access mode (W=write, R=read).

FILE VERSIONS AND TIMESTAMPS REPORT RVT10

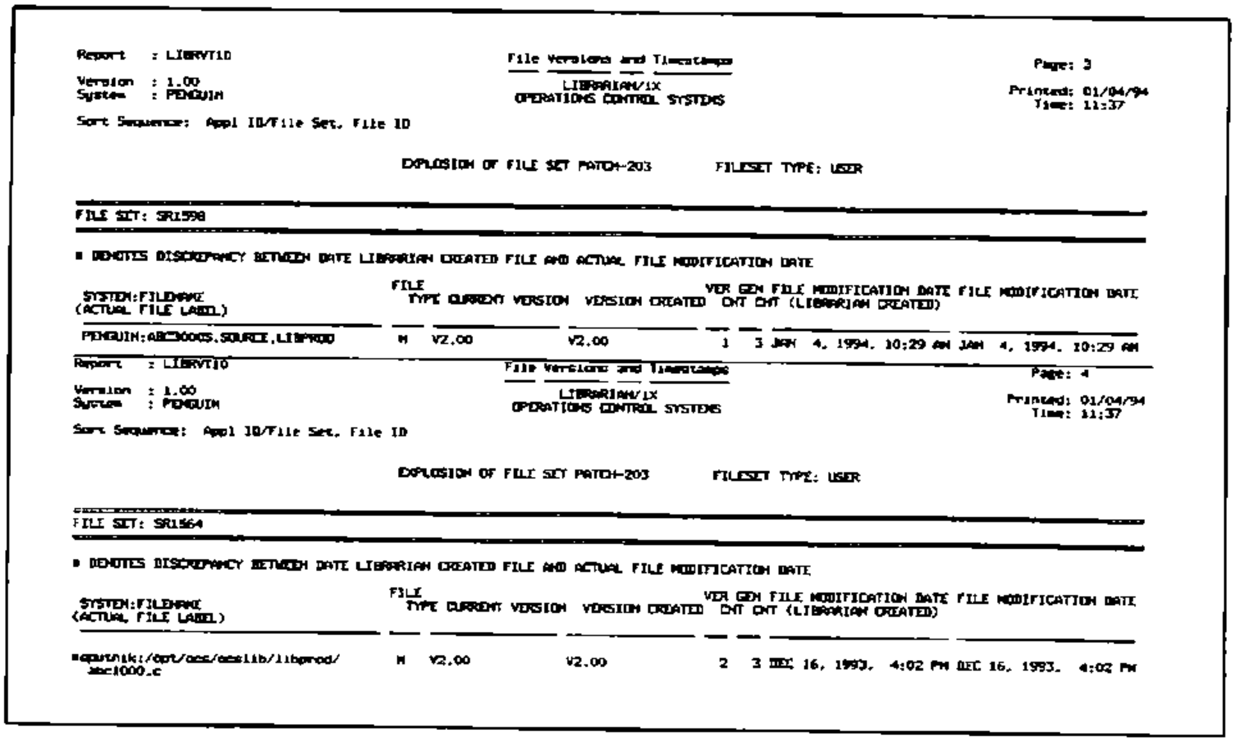
Sort Sequence	Application/Fileset, File ID		
Files Accessed	D-AHFSET-COMP D-AHFSET-FILE D-APPL-VERSION D-FILE D-FSET-COMP	D-FSET-FILE D-SYSTEM-PROFILE D-VFSET-COMP D-VFSET-FILE M-APPLICATION	M-FILE-SET
When to Run	On demand		
Menu Access	**Info...Versions...RVT10 File Versions and Timestamps**		

Description

Identifies all files in an application or fileset and highlights any changes made outside the LIBRARIAN program. Shows version and timestamp information for each file, including version ID, version count, generation count, modification timestamp when the file was created by LIBRARIAN, and the actual modification timestamp for the file.

If you select the "AT" option while generating this report, all secondary files at the location specified will be selected, rather than the master files

Sample



```
Report   : LIBRVT10                         File Versions and Timestamps                        Page: 3
Version  : 1.00                                    LIBRARIAN/iX                           Printed: 01/04/94
System   : PENGUIN                         OPERATIONS CONTROL SYSTEMS                            Time: 11:37
Sort Sequence:  Appl ID/File Set, File ID

                           EXPLOSION OF FILE SET PATCH-203        FILESET TYPE: USER

FILE SET: SR1598

* DENOTES DISCREPANCY BETWEEN DATE LIBRARIAN CREATED FILE AND ACTUAL FILE MODIFICATION DATE

                               FILE                                    VER GEN FILE MODIFICATION DATE FILE MODIFICATION DATE
 SYSTEM:FILENAME                 TYPE CURRENT VERSION  VERSION CREATED  CNT CNT (LIBRARIAN CREATED)
(ACTUAL FILE LABEL)
 PENGUIN:ABC3000S.SOURCE.LIBPROD  M   V2.00            V2.00             1   3 JAN  4, 1994, 10:29 AM JAN  4, 1994, 10:29 AM
Report   : LIBRVT10                         File Versions and Timestamps                        Page: 4
Version  : 1.00                                    LIBRARIAN/iX                           Printed: 01/04/94
System   : PENGUIN                         OPERATIONS CONTROL SYSTEMS                            Time: 11:37
Sort Sequence:  Appl ID/File Set, File ID

                           EXPLOSION OF FILE SET PATCH-203        FILESET TYPE: USER

FILE SET: SR1564

* DENOTES DISCREPANCY BETWEEN DATE LIBRARIAN CREATED FILE AND ACTUAL FILE MODIFICATION DATE

                               FILE                                    VER GEN FILE MODIFICATION DATE FILE MODIFICATION DATE
 SYSTEM:FILENAME                 TYPE CURRENT VERSION  VERSION CREATED  CNT CNT (LIBRARIAN CREATED)
(ACTUAL FILE LABEL)

*sputnik:/opt/ocs/ocslib/libprod/ M   V2.00            V2.00             2   3 DEC 16, 1993,  4:02 PM DEC 16, 1993,  4:02 PM
  abc1000.c
```

FILE VERSIONS AND TIMESTAMPS REPORT (continued) RVT10

Field Descriptions

Filename	Name of the file in the application fileset.
File Type	Type of file:
	M — Master
	S — Secondary
	GM — Generated master
	GS — Generated secondary
	CM — Copies of retained master
	CS — Copies of retained secondary
Current Version	Current version number.
Version Created	Version when the file was first introduced.
Vcnt	Version count (VCOUNT) for the file.
Gcnt	Generation count (GCOUNT) for the file.
File Modification Date (LIBRARIAN Created)	Last date and time the file was modified by LIBRARIAN.
File Modification Date (Actual File Label)	Last date and time the file was modified (outside LIBRARIAN).

FILE VERSIONS AND TIMESTAMPS EXCEPTIONS REPORT

RVT20

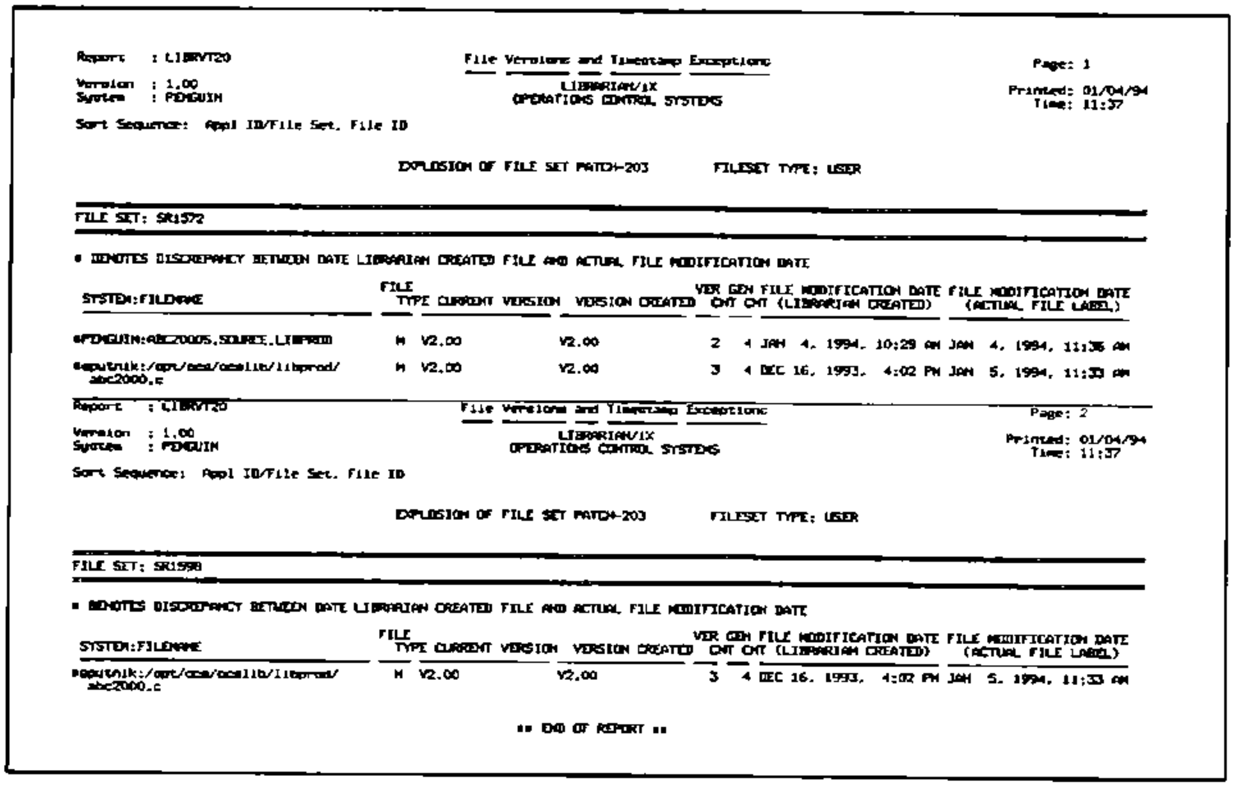
Sort Sequence	Application/Fileset, File ID
Files Accessed	D-AHFSET-COMP, D-AHFSET-FILE, D-APPL-VERSION, D-FILE, D-FSET-COMP, D-FSET-FILE, D-SYSTEM-PROFILE, D-VFSET-COMP, D-VFSET-FILE, M-APPLICATION, M-FILE-SET
When to Run	On demand
Menu Access	**Info...Versions...RVT20 File Versions/Timestamp Exceptions**

Description

Identifies all files in an application or fileset that have been changed outside LIBRARIAN. Shows version and timestamp information for each file including version ID, version count, generation count, modification timestamp when the file was created by LIBRARIAN, and current modification timestamp in the file.

If you select the "AT" option while generating this report, all secondary files at the location specified will be selected, rather than the master files

Sample



```
Report   : LIBRVT20                       File Versions and Timestamp Exceptions                 Page: 1

Version  : 1.00                                      LIBRARIAN/iX                        Printed: 01/04/94
System   : PENGUIN                          OPERATIONS CONTROL SYSTEMS                      Time: 11:37

Sort Sequence:  Appl ID/File Set, File ID

                                 EXPLOSION OF FILE SET PATCH-203        FILESET TYPE: USER

FILE SET: SR1572

* DENOTES DISCREPANCY BETWEEN DATE LIBRARIAN CREATED FILE AND ACTUAL FILE MODIFICATION DATE

                                      FILE                                 VER GEN FILE MODIFICATION DATE FILE MODIFICATION DATE
 SYSTEM:FILENAME                       TYPE CURRENT VERSION  VERSION CREATED CNT CNT (LIBRARIAN CREATED)    (ACTUAL FILE LABEL)

*PENGUIN:ABC2000S.SOURCE.LIBPROD        M  V2.00            V2.00            2   4 JAN  4, 1994, 10:29 AM JAN  4, 1994, 11:36 AM

*sputnik:/opt/ocs/ocslib/libprod/       M  V2.00            V2.00            3   4 DEC 16, 1993,  4:02 PM JAN  5, 1994, 11:33 AM
  abc2000.c
Report   : LIBRVT20                       File Versions and Timestamp Exceptions                 Page: 2

Version  : 1.00                                      LIBRARIAN/iX                        Printed: 01/04/94
System   : PENGUIN                          OPERATIONS CONTROL SYSTEMS                      Time: 11:37

Sort Sequence:  Appl ID/File Set, File ID

                                 EXPLOSION OF FILE SET PATCH-203        FILESET TYPE: USER

FILE SET: SR1598

* DENOTES DISCREPANCY BETWEEN DATE LIBRARIAN CREATED FILE AND ACTUAL FILE MODIFICATION DATE

                                      FILE                                 VER GEN FILE MODIFICATION DATE FILE MODIFICATION DATE
 SYSTEM:FILENAME                       TYPE CURRENT VERSION  VERSION CREATED CNT CNT (LIBRARIAN CREATED)    (ACTUAL FILE LABEL)
*sputnik:/opt/ocs/ocslib/libprod/       M  V2.00            V2.00            3   4 DEC 16, 1993,  4:02 PM JAN  5, 1994, 11:33 AM
  abc2000.c

                                          ** END OF REPORT **
```

FILE VERSIONS AND TIMESTAMP EXCEPTIONS REPORT (continued)

Field Descriptions

Filename	Name of the file in the application fileset.
File Type	Type of file:
	M — Master
	S — Secondary
	GM — Generated master
	GS — Generated secondary
	CM — Copies of retained master
	CS — Copies of retained secondary
Current Version	Current version number.
Version Created	Version when the file was first introduced.
Ver Cnt	Version count (VCOUNT) for the file.
Gen Cnt	Generation count (GCOUNT) for the file.
File Modification Date (LIBRARIAN Created)	Last date and time the file was modified by LIBRARIAN.
File Modification Date (Actual File Label)	Last date and time the file was modified outside of LIBRARIAN.

Macro Control Language 7

Macros are files containing LIBRARIAN commands. They can operate either on a single file or on a group of files, and can take parameters. Macros can also contain looping and conditional logic by using the LIBRARIAN macro control language.

This chapter describes the control language you can use in macros. The following topics are included:

- Executing Macros
- Editing Macros
- Macro Filelists
- Macro Parameters
- Comments in Macros
- Using **STREAM** and **RUN** Commands in Macros
- Macro Control Language Summary
- Macro Control Command Descriptions and Syntax

For more information on using macros, refer to Chapter 9, "Macros" in the *LIBRARIAN/iX User's Guide*.

Executing Macros

To execute commands in a macro file, type the name of the macro file at the LIBRARIAN prompt. For example:

```
>PAYFILE
```

If the macro filename is the same name as a LIBRARIAN command or step, precede the name with **XEQ**. For example:

```
>XEQ CHECKIN
```

For details on using the **XEQ** command to execute macros, refer to Chapter 1, "Commands".

Editing Macros

You can create and edit macros with any editor. Macros must be saved as ASCII files. A macro file can contain system commands, LIBRARIAN commands, and special macro control language commands, as described later in this chapter.

System commands, if ambiguous, must be preceded with a colon (:).

Note Use the ampersand (&) for MPE, or the backslash (\) for UNIX as a line continuation character in macros.

Macro Filelists

Macros can process lists of files. In addition, procedure files can contain macros, as discussed in Chapter 9, "Macros" of the *LIBRARIAN/iX User's Guide*. To do this, include the following statement as the first nonblank line in the macro file, or procedure:

```
OPTION FILES
```

The macro processor then expects you to specify files as the first argument when you execute the macro (see the beginning of Chapter 1). The following example executes a macro called DISTRIB with a set of files:

MPE
UNIX

```
>DISTRIB @.OBJECT.PROD
>DISTRIB ./prod/object/*
```

The macro processor explodes these files in the same way as the **VERIFY** command. You must fully qualify files that are outside your current login directory.

Macros also let you authorize files based on a specified step. For example, the files will be authorized as though that step was about to be performed, but the step action will not be taken. This feature allows you to authorize the macro for specific users (by authorizing the step), to default locations, and to define file scope through the step fileset, source location, and step refinement exclusions. Macro authorization by step is done by specifying the following statement as the first statement in the macro file. For example:

OPTION FILES=*stepname* [*.route* [*.application*]]

When using **OPTION FILES**, the macro processor builds a listfile called XEQLIST, containing the authorized files. You can refer to this listfile within the macro to operate on these files. The following example authorizes files using the CHECKIN step, performs the step, and then distributes the files to a remote system.

```
OPTION FILES=CHECKIN
CHECKIN !XEQLIST
DISTRIBUTE *;NOVIOLATIONS
```

The files that comprise XEQLIST can be referenced one at a time, using the %%[] parameter. For more information, refer to the %%[] parameter in the following section.

User-Defined Help

Help for macros will display user-defined help included in the macro file or procedure. The help text must be entered first, immediately following the OPTION statement, if one exists. Precede the help text by a vertical bar (|) in the first column. For example:

```
PROCEDURE xyz
| This is help text.
| See how it works
ECHO OFF
TEST-CMD
END
```

or

```
OPTION FILES
| This is help for a macro file.
| See how it immediately follows
| the OPTION file statement.
```

If no help text is provided, the macro itself is listed unless **OPTION NOHELP** is specified.

Macro Parameters

This section discusses the parameters that you can use in your macros. You can use parameters to substitute values at execution time.

Parameters can take values such as filenames (specified when you execute the macro), user-supplied parameter values (specified during execution or with the **PARM** command), or system defined parameters.

Prefix all parameters with two percent signs (%%). The first % designates the position where the substitution occurs.

Valid parameters include:

%%[ ]	Substitutes filenames between the square brackets (applying an optional edit mask enclosed in the brackets). This parameter is valid only when you use **OPTION FILES.** If you do not use this parameter within a **LOOP/NEXT** structure, the command will be iterated for each file specified. You can transform filenames by placing edit masks inside the brackets of the parameter %%[]. For more information about edit masks, refer to *Edit Masks* at the beginning of Chapter 1, "Commands."

%%*	If you specify a filename in the **LOOP/ NEXT** structure for each iteration, the next record in that file is substituted wherever %%* is encountered within the loop.
%%0-99	Substitutes the value of the user-supplied parameter with the specified number from 0 to 99.
%%A	Substitutes the current login account.
%%E	Sends an escape character (for video enhancements).
%%G	Substitutes the current login group.
%%N	Turns on line drawing (CONTROL-N).
%%0	Turns off line drawing (CONTROL-O).
%%P	Substitutes the project name supplied by the user during **OPTION FILES** authorization.
%%S	Substitutes the current system name.
%%U	Substitutes the current login user.
%% (!variable name)	Substitutes the value for the MPE/iX or UNIX user or system defined variable enclosed in parentheses. The exclamation point (!) is optional.

Like variables, you can reuse the parameters %%0 through %%99. In the following example, parameter zero is reused:

```
PARM 0;PROMPT="Enter your user ID"
IF "%%0"="DEBBY" THEN
...

PARM 0;PROMPT="Enter your security access code"
IF "%%0"="DUMMY" THEN
...
```

A parameter retains its assigned value until you change it.

You can use additional macro parameters whose values are determined by the current file being processed (i.e., in a **LOOP...NEXT** block or on a line that refers to %% [ ]):

%% (!OWNER)	The owner of the file.
%% (!MSUSER)	The ID of the user who checked out the file.
%% (!TAG)	The file's tag.
%% (!PROJECT)	The name of the project this file is associated with.

For example, you can automatically send mail messages to the owner or checkout user of the file being processed. The following example illustrates this.

```
OPTION FILES=CHECKIN
LOOP
   COMMENT Store off owner prior to checkin
   PARM 1="%%(!OWNER)"
   CONTINUE
   CHECKIN %%[]
   IF LIBERRORS>0 OR LIBFAIL>0 THEN
      MAIL %%1;CHECKIN FAILED for %%[]
   ELSE
      MAIL %%1;CHECKIN SUCCEEDED for %%[]
   ENDIF
NEXT
```

Comments in Macros

Comment lines can appear anywhere in a macro. Comment lines start with an asterisk (*) or vertical bar (|).

Using STREAM and RUN Commands in Macros

When using the **STREAM** and **RUN** commands in a macro, you can include input to them in the macro file. Input to **STREAM** would be JCL lines, and input to **RUN** would include program input such as responses to prompts. Do not precede this input with the LIBRARIAN prompt (>) or the MPE prompt (:). Instead, use an exclamation point (!) for JCL.

For the **RUN** command, use the STDIN parameter, setting it to XEQ (i.e., **STDIN=XEQ**). Do not precede the **RUN** command name with a colon.

For **STREAM**, do not specify a filename. Lines from the macro file are passed to your program (**RUN**) or to a jobstream (**STREAM**) with parameter substitution until ">" is encountered on a separate line.

Macro Control Language Summary

The macro control language commands allow you to

- turn command echoing on/off, or display a message,
- control command flow through conditional logic and looping structures,
- refer to files,
- set default values for parameters,
- define procedures,
- prompt users for parameter values.

Table 7–1 lists the macro control language commands and functions. Page references locate the detailed command descriptions.

Table 7-1. Macro Control Language Commands Summary

Command Name	Function	Page
ADJUST	Adjusts a JCW value up or down.	7–7
CONTINUE	Indicates that macro processing should continue if an error occurs.	7–8
ECHO	Turns echoing on and off, or displays a message.	7–9
END	Signifies the end of a procedure or signals premature termination of a macro	7–10
GOTO	Resumes execution at a specified location.	7–11
IF/ELSE	Executes the intervening block of commands if a conditional expression is true.	7–12
LOOP/NEXT	Executes a block of commands for each file in a filelist, or each record in a specified file	7–13
MENU	Controls default menu mode operation.	7–14
OPTION	Specifies macro options.	7–15
PARM	Sets default values for parameters in macros, optionally prompting the user, or providing a menu.	7–16
PROCEDURE	Specifies the name of a procedure.	7–19
REPEAT/UNTIL	Executes the intervening block of commands until a conditional expression is true.	7–20
SETVAR	Assigns a value to a macro variable.	7–21
WAIT	Allows you to pause macro processing with a message.	7–22
WHILE/ENDWHILE	Executes the intervening block of commands while a conditional expression is true.	7–23

Macro Control Command Descriptions and Syntax

This section contains descriptions and syntax for all macro control language commands.

The macro control language command descriptions include the following:

Syntax	How to enter command.
Parameters	Detailed description of each command parameter.
Operation	The basic function and detailed descriptions of the command.
Examples	Examples of the command.

ADJUST

Adjusts a JCW value up or down.

Syntax

ADJUST *JCWname*, *increment*

Parameters

JCWname	The name of the JCW.
increment	A positive or negative number (often 1).

Operation

ADJUST lets you adjust the value of a JCW up or down. This is useful when creating loops from 1 to *n*. If the JCW does not exist, it is created with a value of zero.

Example

Adjust the JCW MYJCW by typing:

```
ADJUST MYJCW,1
```

CONTINUE

Indicates that macro processing should continue if an error occurs.

Syntax

```
CONTINUE
```

Parameters

None

Operation

CONTINUE lets you indicate that if an error occurs, processing of the macro file should continue. The **CONTINUE** command must precede each command for which this behavior is desired. If you do not use **CONTINUE**, the remainder of the macro file is not processed.

Example

In the following example, if an error occurs during checkout, processing should continue:

```
    CONTINUE
    CHECKOUT %%[]
```

ECHO

Turns echoing on and off, or displays a message.

Syntax

```
ECHO  { ON                  }
      { OFF                 }
      { NULL                }
      { STDLIST             }
      { [ PROMPT ] message  }
```

Parameters

ON	Turns ON echoing of commands in the macro.
OFF	Turns OFF echoing of commands in the macro.
NULL	Redirects LIBRARIAN output to $NULL.
STDLIST	Restores LIBRARIAN output to $STDLIST.
PROMPT	Echoes the specified message without a carriage return.
message	The message displayed to the user. Omitting *message* produces a blank line.

Operation

ECHO lets you toggle the echoing of commands to ON and OFF, display a message to STDLIST, and/or redirect LIBRARIAN output.

Example

Turn the echoing of commands off by typing:

```
ECHO OFF
```

END

Signifies the end of a procedure in a procedure file, or premature termination of a macro being processed.

Syntax

```
END
```

Parameters

None

Operation

END marks the end of a procedure in a procedure file. Additionally, you can use the **END** command to terminate a macro permanently.

Example

The following example shows a procedure called SJJ, terminated by **END**:

```
PROCEDURE SJJ
ECHO OFF
SHOWJOB EXEC;JOB=@J
END
```

GOTO

Resumes execution at a specified location.

Syntax

GOTO *label*

Parameters

label	The location where the macro should resume execution.

Operation

GOTO allows you to resume executing a macro at a specific point. The label must appear after the **GOTO** statement in the format:

```
label:
```

Example

Start execution at label ERROR: by typing:

```
GOTO ERROR
    .
    .
    .
 ERROR:
    .
    .
    .
```

IF/ELSE

Executes the intervening block of commands if a conditional expression is true.

Syntax

```
IF ⎧ EXIST filename                                ⎫
   ⎪ EXISTTEMP filename                            ⎪
   ⎨ "string1" = "string2"                         ⎬
   ⎪ "string1" IN "string2"                        ⎪
   ⎩ JCWname/value <relop> JCWname/value           ⎭  [ THEN ]
ELSE
ENDIF
```

Parameters

EXIST *filename*	True, if the file exists in the permanent domain.
EXISTTEMP *filename*	True, if the file exists in the temporary domain.
"string1" = *"string2"*	True, if the two strings match.
"string1" **IN** *"string2"*	True, if the first string is a substring of the second substring.
JCWname/value	The name of a JCW and/or a value.
relop	The relational operator. Valid operators include:

=	equal to
<>	not equal to
<=	less than or equal to
>=	greater than or equal to

Operation

The **IF/ELSE** structure executes the intervening block of commands if the conditional expression is true. **IF** statements can be nested, and can exist within loops:

Example

In the following example, when the condition is met, "Your JCW is set to zero" is printed. If the condition is not met, "Your JCW is not set to zero" is printed:

```
IF LIBJCW = 0 THEN
    ECHO Your JCW is set to zero.
ELSE
    ECHO Your JCW is not set to zero.
ENDIF
```

LOOP/NEXT

Executes a block of commands for each file in XEQLIST (the listfile created when an **OPTION FILES** macro is invoked), or for each record in a specified file.

Syntax

```
LOOP  [ filename ]
...
NEXT
```

Parameters

filename By default, **LOOP** executes the block of commands for each file in XEQLIST, substituting the current filename wherever %%[] is encountered. If you specify a filename, however, the **LOOP** is for each record in that file and the data in that record is substituted wherever %%* is encountered within the loop.

Operation

The **LOOP/NEXT** structure executes the intervening block of commands for each authorized file. If you want to execute several commands on a group of files or records in a specific file, use the **LOOP/NEXT** structure.

The current filename is substituted wherever the %%[] parameter is located. When using **LOOP/NEXT** in a file other than the default XEQLIST, use the %%* parameter wherever you want the current record to be substituted. For more information on the %%[] and %%* parameters, refer to *Macro Parameters*, earlier in this chapter.

Note Loops cannot be nested.

Example

The following example shows a **LOOP/NEXT** structure:

```
LOOP
    <command 1>
    <command 2>
NEXT
```

MENU

Controls default menu mode operation.

Syntax

```
MENU  ⎧ FILE  ⎫
      ⎪ LAST  ⎪
      ⎨ MACRO ⎬
      ⎪ MAIN  ⎪
      ⎪ OFF   ⎪
      ⎩ STEPS ⎭
```

Parameters

FILE	Opens file menu initially and any time user returns from command mode.
LAST	Always returns to the last menu that was open when returning from command mode.
MACRO	Opens macro menu initially and any time user returns from command mode.
MAIN	Displays main menu initially and any time user returns from command mode (default).
OFF	Bypasses menu mode and brings user directly to the command line prompt.
STEPS	Opens the steps menu initially and any time user returns from command mode.

Operation

The **MENU** command is designed for use in an AUTOXEQ file to control which menu appears, by default, if any. If you want to bypass menus altogether when accessing LIBRARIAN, then use the **MENU OFF** command in your AUTOXEQ file.

Example

Turn menus off so that only the command mode prompt appears:

```
MENU OFF
```

OPTION

Specifies options to control the execution of macros.

Syntax

```
OPTION [ FILES [ = step [ .route [ .appl ] ] ] ]
    [, FILES ]
    [, LOCALPARMS ]
    [, NOBREAK ]
    [, NOHELP ]
    [, NOWARN ]
    [, ONEPARM ]
    [, PARMS ]
    [, QUIET ]
```

Parameters

FILES	Requires a user to specify files when executing the macro or procedure. You can nest multiple macros with OPTION FILES. LIBRARIAN keeps track of the nesting level, and opens a new XEQLIST1,2,3,...n as each nested macro is invoked. Nested looping is also supported through nested OPTION FILES macros. See the second example that follows.
step	The name of the step to authorize files.
route	The name of the route to which the step belongs.
appl	The name of the application to which the route and step belong.
LOCALPARMS	Causes the XEQ parameters to be local to this macro when it is executed within another macro (parameters are global by default). With the **LOCALPARMS** option set, parameters 0 –99 are initialized to NULL values upon entry to the macro and reset to their original values upon exit. If this option is not used, parameter values are inherited from the calling macro, and if their values are modified within this macro, the changed values are "passed back" to the calling macro on exit.
NOBREAK	Disables **Break** and **Control-Y** during macro processing.
NOHELP	Disables help for this macro.
NOWARN	If a parameter does not have a value in an XEQ macro, **NOWARN** will initialize the value to blank and suppress the warning message to the user.
ONEPARM	Allows the user's input (data supplied on the command–line) to be accepted into PARM 0 (%%0) as a single string, rather than parsed into separate parameters. This is useful when the intent of the macro is to extend the operation of a defined step, but preserve step syntax including multiple file specifications, AT and TO clauses, and run–time parameters. See the last example on the following page.
PARMS	In menu mode, causes a dialog box to appear, allowing you to specify parameters. If **OPTION FILES** has been defined, a dialog for filenames and optional edit masks appears. If neither **OPTION FILES** nor **OPTION PARMS** is defined, no dialog appears and the macro is executed immediately.
QUIET	Suppresses the standard messages and prompts, and uses the default answers for commands within the macro.

OPTION (continued)

Operation

OPTION must appear as the first nonblank, noncomment line in a macro. **OPTION FILES** causes LIBRARIAN to authorize the file(s) specified when executing the macro. **OPTION FILES**=*step* authorizes files according to the authorization rules for the specified step.

The listfile XEQLIST is built containing the names of authorized files when **OPTION FILES** is used. For more information, refer to *Macro Filelists,* earlier in this chapter.

Examples

The following example submits source code for testing, and compiles each related program in the production account using **MAKE**:

```
OPTION FILES=ABC-SUBMIT.ABC-MAINT.ABC,NOBREAK,NOHELP
ABC-SUBMIT !XEQLIST
MAKE ABCMAKE.PUB.ABCQA,%%[@-P.OBJECT.ABCQA]
```

The next example checks in files specified by the user, and notifies the owner of every copy of each file checked in:

```
PROCEDURE ABC-IN
OPTION FILES=ABC-IN
LOOP
    SETJCW LIBOK=O
    CONTINUE
    ABC-IN %%[ ]
    IF LIBOK>0 THEN
            CONTINUE
            ABC-NOTIFY * AT @.@.@.@
    ENDIF
NEXT
END

PROCEDURE ABC-NOTIFY
OPTION FILES
LOOP
    MAIL %%(!OWNER), A new version of %% [ ] &
                    has been checked in
NEXT
END
```

The following example changes the owner after sending files to the test area, using the same syntax as the ABC–TEST step:

```
PROCEDURE ABC-TEST
OPTION ONEPARM,NOHELP,NOBREAK
PARM 1;PROMPT="Owner for test files:";required
SETJCW LIBOK=0
CONTINUE
ABC-TEST %%0
IF LIBOK>0 THEN
    ALLOW LIBMGR:pass
    CONTINUE
    SET * OWNER=%%1
    ALLOWENDIF
END
```

PARM

Sets parameter values used in macros, optionally presenting a menu or prompting the user.

Syntax

```
>PARM n [= "default_value"]

        [ ;MENU=menu-file, [entry] | INLINE
        [ ;NUMBER ]
        [ ;PICK [=delimiter] | PICKFILE [ ;REQUIRED ] ]
        [ ;SORT ]
        [ ;TITLE=title ]]  |
        [ ;PROMPT = "prompt_text" [ ;NOECHO]]
        [ ;REQUIRED]
        [ ;UPSHIFT]
```

Parameters

n	The parameter number (0 to 99).
default_value	The default value of the parameter.
MENU	Displays a menu to get a parameter value from the user.
menu-file	A text file containing a menu specification with one record for each option, in the following format:

```
menu-item[=translation]
```

where *menu-item* is what appears on the user menu, and *translation* is the value returned for the parameter (if different).

The following menu files are automatically created by LIBRARIAN and are available for your use:

APPLMENU	menu of all defined applications.
RTMENU	menu of routes for all authorized steps.
STEPMENU	menu of all authorized steps.
PROJMENU	menu of all authorized projects.

Menus can be combined in a single menu file by using entry points.

entry	The name of an entry point in a menu specification file.

PARM (continued)

Parameters, continued

Entry points are identified by a keyword that is followed by a colon (:) in a menu file, and a blank line between menus. For example:

```
MENU1:
Option1
Option2
Option3

MENU2:
OptionA
OptionB
OptionC
```

INLINE	If *menu–file* is **INLINE**, *menu–items* are listed in the macro file on lines immediately following the **PARM** command. Use a > in column 1 on a line by itself after the last menu item to indicate the end of the menu.
NOECHO	Turns off echoing of the user's response to a prompt. This option is useful when prompting for passwords within a macro.
NUMBER	Returns the menu item number (physical record number) rather than the menu item text/translation.
PICK=*delimiter*	Specifies that multiple selections are allowed. Selections are returned in a string separated by *delimiter.* If delimiter is not specified, there will be no separation between choices.
PICKFILE	Indicates that multiple selections are allowed. Each choice is written as a record to a temporary file PICK*x*, where *x* is the parameter number. This file can be used with LOOP..NEXT. The value of the parameter will be the name of the file where selections are stored.
SORT	Presents the menu in sorted order.
TITLE	Specifies a title for the menu.
PROMPT	Allows you to prompt for a value.
prompt_text	The prompt to display. Use "%%%" in the text prompt to substitute the current or default value for the parameter.
REQUIRED	Specifies that the user must choose at least one item from a menu or must provide a response to a prompt.

Operation

PARM specifies values for parameters used in a macro. Macros can contain a maximum of 100 parameters substituted at runtime. If you refer to macros within a macro, they will all share the same parameter values.

Menus that you create with the **PARM** command operate in the same way as standard LIBRARIAN menus. Refer to Chapter 2, "Getting Started" in the *LIBRARIAN/iX User's Guide* for more information.

Examples

Example 1 uses a parameter to request a project name from the user.

Parameters, continued

```
HELP PROJECTS
PARM 1;PROMPT "Enter Project Name :";REQUIRED
ABC-CHECKIN.%%1;NOVIOLATIONS
```

Example 2 presents a menu of patches for the user to select from.

```
PARM 2 ;MENU=INLINE; TITLE=Patches ;REQUIRED
NS Transport=NSTCDV1
Network=NMSCDU1
AIFSYSWIDESET= MPEXF46
>
```

Example 3 uses LMAINT to create an indirect list of files, presented to the user as a menu. Selections are then displayed, one per line.

```
ECHO NULL
LMAINT
OUTPUT %SOURCE-FILES TO SRCFILES ;ALL
EXIT
ECHO STDLIST
PARM 3 ;MENU=SRCFILES ;TITLE=Source Menu ;PICKFILE
LOOP %%3
    ECHO %%*
NEXT
```

PROCEDURE

Specifies the name of a procedure in a procedure file, and marks the beginning of that procedure.

Syntax

PROCEDURE *procname*

Parameters

procname	The name of the procedure.

Operation

The **PROCEDURE** statement indicates the beginning of a new procedure, and specifies its name containing a maximum of twelve characters. To make procedures available in a LIBRARIAN session, use the **SET PROCEDURE** command.

Note If a loaded procedure name is the same as a step name, the procedure takes precedence over the step.

Example

The following example produces a procedure called SJJ:

```
PROCEDURE SJJ
ECHO OFF
SHOWJOB EXEC;JOB=@J
END
```

REPEAT/UNTIL

Executes the intervening block of commands until a conditional expression is true.

Syntax

```
REPEAT
...
UNTIL condition
```

Parameters

condition	A valid condition, including text strings, parameters, JCWs, or literal numbers. (See **IF/ELSE**)

Operation

The **REPEAT/UNTIL** structure executes the intervening block of commands until a conditional expression holds a true value.

Note Loops cannot be nested.

Example

The following example repeats the commands until the JCW **LIBJCW** is zero:

```
REPEAT
    <command 1>
    <command 2>
UNTIL LIBJCW = 0
```

SETVAR

Assigns a value to a macro variable.

Syntax

```
SETVAR varname  ⎧ <space> ⎫  expression
                ⎨ ,       ⎬
                ⎩ ;       ⎭
```

Parameters

varname	The variable that is to be set to a value.
expression	The expression that is evaluated and assigned to *varname*.

Operation

This command assigns values to macro variables. Variable names may be any combination of letters and numbers plus the underbar character, up to a total of 255 characters. Variables must start with a letter or the underbar character.

The *expression* parameter may be a macro expression, a Boolean, integer, or string value, or the name of another variable. If *expression* consists of elements and operators ('abc' + 'cd' or 2*5+1), SETVAR will evaluate it. The operators defined in Table 7–2 may be used in *expression*.

Table 7–2. Logical Operators – The SETVAR Command

Logical operators:	AND, OR, XOR, NOT
Boolean functions and values:	BOUND, TRUE, FALSE, ALPHA, ALPHANUM, NUMERIC, ODD
Comparison operators:	==, <>, <, >, <=, >=
Bit manipulation operators:	LSL, LSR, CSR, CSL, BAND, BOR, BXOR, BNOT
Arithmetic operators:	MOD, ABS, *, /, +, –, ^ (exponentiation)
Functions returning strings:	CHR, DWNS, UPS, HEX, OCTAL, INPUT, LFT, RHT, RPT, LTRIM, RTRIM, STR
Functions returning integers:	ABS, LEN, MAX. MIN, ORD, POS, TYPOF

Operands can be any variable, integer constant (hexadecimal ($), octal (%), or decimal) quoted string constant, the Boolean constants TRUE and FALSE.

Compound logical expressions can be formed using the AND, NOT, XOR, and OR logical operators, and nested within parentheses.

WAIT

Creates a pause during macro processing until the user presses a key.

Syntax

WAIT [*message* | *pausetime*]

Parameters

message	The message to display.
pausetime	The number of seconds to wait.

Operation

The **WAIT** command creates a pause during macro processing either until the user presses a key or until a time interval has passed. Additionally, you can provide a message. If you do not specify a message, users are, by default, prompted with the line "Press any key to continue...".

Examples

Pause with a message by typing:

```
WAIT Processing complete. Please press a key.
```

Pause for five seconds by typing:

```
WAIT 5
```

WHILE/ENDWHILE

Executes the intervening block of commands while a conditional expression is true.

Syntax

```
WHILE condition
. . .
ENDWHILE
```

Parameters

condition	The condition that should be true, including text strings, parameters, JCWs, or literal numbers. (See **IF/ELSE**)

Operation

The **WHILE/ENDWHILE** structure executes the intervening block of commands while the conditional expression holds a true value.

Note Loops cannot be nested.

Example

The following example executes the commands while the JCW LIBJCW is zero:

```
WHILE LIBJCW = 0
     <command 1>
     <command 2>
ENDWHILE
```

MAKE 8

The MAKE facility helps you keep applications up–to–date. It accomplishes this by building only those components whose dependencies have changed.

This chapter explains how to create makefiles with information about Make rule syntax. The following topics are covered:

- Makefiles
- Edit Masks
- Variables

Refer to Chapter 8, "Rebuilding Applications with Make" in the *LIBRARIAN/iX User's Guide* for more information.

Makefiles

A makefile is simply a text file that contains your MAKE rules, which includes dependency information and instructions for how to rebuild each component of your application. You can create and maintain this file in the editor of your choice. Although "makefile" is the default name for this file, you can use any name you wish. A makefile can contain comments, rules, and user-defined variables.

A makefile consists of one or more rules depending on the complexity of your application. Each rule defines a specific or generic target/dependency relationship and the commands required to rebuild the target from the dependencies.

Conventions

When creating your makefile, adhere to the following conventions:

- Insert a blank line between rules.
- Use the backslash (\) as a line continuation character.
- When listing targets and dependencies, use a minimum of one space between filenames.

Comments

Comments should be on separate lines and can appear anywhere in a makefile. A comment line in a makefile must start with a pound sign (#).

For example:

```
#This is a comment.
```

Comments that begin with #NOTE are treated in a special way at run time. If you use the **ECHO** option when you run MAKE, you will see comments displaying on the screen as the makefile is processed. For example, suppose you have the following comment:

```
#NOTE Processing report rules...
```

When MAKE processes this file with the **ECHO** option, the following line would appear:

```
Processing report rules...
```

Comments that begin with #OPTION followed by an option list invoke those options automatically at runtime. For example:

```
#OPTION SHOW ECHO
```

The option list can include any MAKE parameters (**SHOW**, **ECHO**, **NOMAKE**, **ALL**, etc.) as described in Chapter 1.

Rules

Rules are statements that inform MAKE about dependencies between files. They also identify the command(s) required to rebuild an object when its dependencies change. Rules can be standard, generic, or implicit:

- Use *standard rules* when the targets and their dependencies are specific file names. Standard rules are required for the highest level target in a dependency tree.
- Use *generic rules* to derive a dependency name from a target name with an edit mask.
- Use *implicit rules* when the target and dependency have the same name but belong to different groups.

Table 8–1 summarizes the standard rule elements that apply when the targets do not include wildcards. Table 8–2 summarizes the generic rule elements that apply when you use wildcards in the target name.

Table 8-1.   Standard Rule Elements

Standard Rules	
Standard Rule Syntax	**Independent Rule Syntax**
target–list : dependency–list *commands*	*target–list :: dependency–list* *commands*
Parameter	**Description**
target-list	Specifies names of target(s) that must be rebuilt if any item in the dependency list has changed (i.e., timestamp of dependency is later than the target). Targets can be file names, variable expressions, or dummy names. If a dummy name is used that does not correspond to an existing file, *commands* are always performed.
dependency-list	Specifies names of dependencies for the target(s). If any item in the dependency has changed (i.e., timestamp of dependency is later than the target), the *command(s)* given for the rule are performed.
commands	Specifies operating system command(s) to execute if target is older than any of its dependencies. Any number of commands can be issued. Because they are output to a JCL file, precede commands with a job character, if necessary (i.e., !).
Operators	
:= (basic generic rule)	Specifies that target and dependency names differ only in suffix.
:– (extended generic rule)	Specifies edit mask(s) in the dependency list, so that dependencies are determined by applying the edit mask to the target being evaluated.
: (standard rule)	Causes all rules that have targets with the same name to be combined into one rule (i.e., all dependencies and commands are combined).
:: (independent rule)	Causes each rule to be treated independently (i.e., only the commands of the rule whose dependency has changed are executed).

Table 8-2. **Generic Rule Elements**

Generic Rules	
Extended Generic Rule Syntax	**Basic Generic Rule Syntax**
target-list :– dependency-edit-mask(s) *commands*	*@suffix := @suffix* *commands*
Parameter	**Description**
target-list	Specifies names of target(s) that must be rebuilt if any item in the dependency list has changed (i.e., the timestamp of the dependency is later than the target). Targets are file names with wildcards. MAKE always applies the first generic rule found for a potential target being analyzed.
dependency-edit-mask	Specifies the edit mask describing how to create the dependency name from the target name.
suffix	Specifies suffix of filename for a dependency or target. With the colon equals (:=) format, filenames of dependency and target can differ only in suffix (i.e., ACOB10S and ACOB10U).
commands	Specifies operating system command(s) to execute if target is older than any of its dependencies. Any number of commands can be issued. Because commands are output to a JCL file, precede commands with a job character, if necessary (i.e., !). Since the rules are generic, commands usually refer to variables to fill in the appropriate filenames.
Operators	
:= (basic generic rule)	Specifies that target and dependency names differ only in suffix.
:– (extended generic rule)	Specifies edit mask(s) in the dependency list, so that dependencies are determined by applying the edit mask to the target being evaluated.

Table 8–3 summarizes the implicit rule where the target and dependency names differ by group only.

Table 8–3. Implicit Rule Elements

Implicit Rules	
Implicit Rule Syntax	
target–group. dependency–group : *commands*	
Parameter	**Description**
dependency group	Specifies the group in which the dependencies reside.
target group	Specifies the group in which the targets reside.
commands	Specifies the operating system command(s) to execute if the target is older than any of its dependencies. Any number of commands can be issued. Because commands are output to a JCL file, precede commands with a job character, if necessary (e.g., !).
Operators	
group1.group2: *(implicit rule)*	Specifies that targets and dependencies differ in group name only.

Job Card Placement

Since you can have MAKE evaluate any standard rule in your makefile, you must define job cards at the likely entry points into the makefile. Any subsequent job cards encountered after the entry point rule are ignored by MAKE.

Because the first rule of a makefile is the default entry point, a job card should be defined in the first rule. Consider where other typical entry points are and place job cards in those rules.

Job card information is written on a separate line. The first three nonblank characters in a job card line must be a colon (:), followed by a greater-than sign (>), and a job character (usually an exclamation point (!)). For example:

```
:> !JOB MYMAKE, MGR.MYACCT/FOOBAR
```

Source Scan for Dependencies

Use MAKE to scan COBOL, Pascal, and C source for dependent modules identified in INCLUDE and COPY statements. When such dependencies are found, they are treated in the same way as explicit dependencies. To invoke source scan for dependencies, follow the dependency file name with a plus sign (+), as in the following examples:

```
ABC1000P.PROG : ABC1000S.SOURCE+
ABC@P.PROG :- =S.SOURCE+
```

Edit Masks

Edit masks are used in generic rule dependency lists and variable filename substitutions.

For editing filename variables, edit masks are enclosed in double quotes ("") and placed immediately after the filename variable to be altered. For example:

```
$<"@S"
```

The edit mask above appends an S onto the current dependency name, where the $< variable substitutes the current dependency name. Edit masks can also be used to alter filenames derived from wildcard variable substitution. For example:

```
$[@S.RSOURCE]"=-.PROG"
```

The LISTF variable creates a list of files that match the wildcard (@S.RSOURCE). The edit mask then alters the filenames obtained by removing the last character and changing the group. It is important to note that this expression generates filenames for the PROG group regardless of whether the files actually exist in the PROG group.

If an edit mask is used in the dependency list of an extended generic rule, the dependency name is derived from the target name. For example:

```
@.PROG :- @S.RSOURCE
```

The generic rule above specifies that the dependency name is derived from the target name by appending an S to the target name and looking in the RSOURCE group. For example, a target name of RPT1020.PROG would be transformed into RPT1020S.RSOURCE as the dependency filename. Table 8–4 describes the characters that can be used in edit masks:

Table 8–4. Edit Mask Characters

Character	Description
@	Copies original value into the edited version. Typically preceded and/or followed by other characters.
=	Copies all remaining characters once the minus sign (-), question mark (?), and literal characters have been evaluated.
?	Copies the character at this position into the resulting string. It can also be combined with the minus sign.
-	Indicates that the original character in that position should not be included in the edited result. It can also be combined with the equal sign.

For more information about edit masks, see *Edit Masks,* at the beginning of Chapter 1, "Commands."

Variables

Variables provide a shorthand for checking makefiles. Variable references are substituted with either user-defined or system-defined values. User-defined variables have the following format:

variable_name = substitution_text

The substitution text replaces all references to the variable at run time. User-defined variables can be assigned the value of other user- or system-defined variables.

Referencing Variables

By default, references to MAKE variables are prefixed with a dollar sign ($). To use MPE–defined names, such as $OLDPASS or $NULL, you must globally change the variable prefix to another character, such as %. To do this, add the following line at the beginning of your makefile.

```
$ = %
```

This changes the variable prefix from $ to %.

Predefined Variables

Following is a list of system-defined MAKE variables:

$[ ]	(LISTF variable)	Executes LISTF and returns all files that match the wildcard specification between the brackets. Can be edited using an edit mask by placing the edit mask in double quotes immediately after the closing bracket. LISTF exclusions, e.g., $[a@–a1–a2–a3] excludes a1,a2,a3 from a@.
${ }	(Prompt variable)	Prompts the user with the string specified between the braces. The value entered by the user is then substituted into the makefile. You can assign this value to a named variable for repeated reference.
$( )	(User variable)	Substitutes the value of the variable specified within the parenthesis. If the variable name is a single character, the parentheses can be omitted.
$(!*system_variable*)	(System variable variable)	Substitutes the MPE system variable between the parentheses.
$<	(Dependency variable)	Substitutes the fully qualified name of the current dependency when performing an action. Can be edited using edit mask.

$@	(Full-target variable)	Substitutes the fully qualified name (including group and account) of the current target. Can be edited using edit mask.
$*	(Target variable)	Substitutes the name of the current target as entered in the rule (i.e., no group or account is given unless it is specified in the rule).

Special Variables

In addition to the standard variables, seven special variables are available. These variables include:

- STREAM
- SCHEDULE
- STREAM and SCHEDULE
- ACCOUNT
- GROUP
- EXCLUDE
- COPYMEM
- ALTPATH

STREAM

You can optionally specify MPE **:STREAM** parameters when MAKE streams a job. When the STREAM variable is used, its value is passed as a parameter list to **STREAM**. For example, if the following variable is used anywhere in a makefile, then MPE launches the job at 5:00 p.m.

```
STREAM = AT=17:00
```

For more information on **STREAM**, Refer to the MPE *Commands Reference Manual*.

SCHEDULE

You may have access to scheduling or streamer programs and do not wish your jobs to be streamed directly to MPE. In this case, MAKE recognizes the SCHEDULE variable. If you define a variable named SCHEDULE anywhere in the makefile, then MAKE expects its value to be the name of the scheduler program. MAKE runs this program (and passes the name of the MAKE jobstream via the **INFO** string), instead of streaming the file. The program name may optionally have a slash (/) at the end, followed by S, P or G corresponding to the lib=*x* parameter that the scheduler program requires.

For example, if the following variable is used in the makefile, then MAKE would run **STREAMER.COMP.EXPRESS** with a **LIB=G** parameter, and pass the name of the MAKE jobstream in the **INFO** string rather than streaming the MAKE command file directly to MPE:

```
SCHEDULE = STREAMER.COMP.EXPRESS/G
```

STREAM and SCHEDULE

If you use both the STREAM and SCHEDULE variables, the scheduler program is invoked and the stream options are appended to the command output filename passed via the INFO string.

The EXPRESS **STREAMER** command, for example, implements the same options as the MPE **:STREAM** command. This also provides a means of specifying additional scheduling parameters.

ACCOUNT

If you run MAKE outside of the account where the files to be evaluated reside, you can use the special ACCOUNT variable to set the account globally. With this variable, you only need to qualify your target and dependency filenames up to the group level in the makefile. For example:

```
ACCOUNT = QAACCT
```

If you specify the ACCOUNT variable in the makefile, you should specify only filenames up to the group level, since the ACCOUNT variable appends the account name to all filenames in the makefile.

EXCLUDE

The special EXCLUDE variable can be used to globally exclude delta files and generation files from LISTF variable processing. For example:

```
EXCLUDE = D#######.@.@  G#######.@.@
```

COPYMEM

The COPYMEM variable is used in conjunction with the MAKE automatic dependency scan feature to indicate that COPYLIB members are stored as individual files in [GROUP[.ACCOUNT]]. Without the COPYMEM variable, the COPYLIB file name is used as the dependency, rather than member names. For example:

```
COPYMEM=MYGROUP
COPYMEM=MYGROUP.MYACCT
```

GROUP

If you run MAKE outside of the group where the files to be evaluated reside, you can use the special GROUP variable to set the group globally.

For example:

```
GROUP = MAKEGRP
```

If you specify the GROUP variable in a makefile, you can only specify filenames, since the GROUP variable appends the group name to all filenames in the makefile.

ALTPATH

The ALTPATH variable provides automatic search of an alternate account location when a designated dependency is not found in the default account defined by the ACCOUNT variable or logon account.

The ALTPATH account is defined by entering a variable definition, usually at the beginning of the makefile. For example,

```
ACCOUNT=ABCDEV
ALTPATH=ABCLIB

ABC : $[@.PROG]
      :>!JOB.........

ABC1000P.PROG : ABC1000S.SOURCE
      !rebuild statements...

@P.PROG :- =S.SOURCE
      !rebuild statements...
```

If the dependency (source file) of the target being evaluated does not exist in the same account as the target (object file), then MAKE searches for the same *file.group* in the ALTPATH account. For instance, if ABC3000P.PROG is found in the account ABCDEV, but ABC3000S.SOURCE.ABCDEV does not exist, MAKE searches for ABC3000S.SOURCE.ABCLIB. If MAKE finds the dependent file in the ALTPATH account, it uses that file as the dependency. All other MAKE logic remains the same.

Menu Hierarchy and Dialogs 9

LIBRARIAN menu mode is an alternative to command mode, allowing you to select options from a set of menus. This chapter summarizes the LIBRARIAN menu hierarchy and describes menu options and dialogs. The following topics are included in this chapter:

- Main Menu
- File Menu
- User Menu
- Macros Menu
- Tools Menu
- Info Menu
- Admin Menu
- Help Menu
- Dialogs

Main Menu

The main menu consists of a horizontal menu bar that appears at the top of the screen under the OCS/LIBRARIAN title bar. It also shows whether you are running LIBRARIAN under MPE/iX or UNIX. Special functions, such as switching between menu mode and command mode, are available with the function keys as displayed in Figure 9–1.

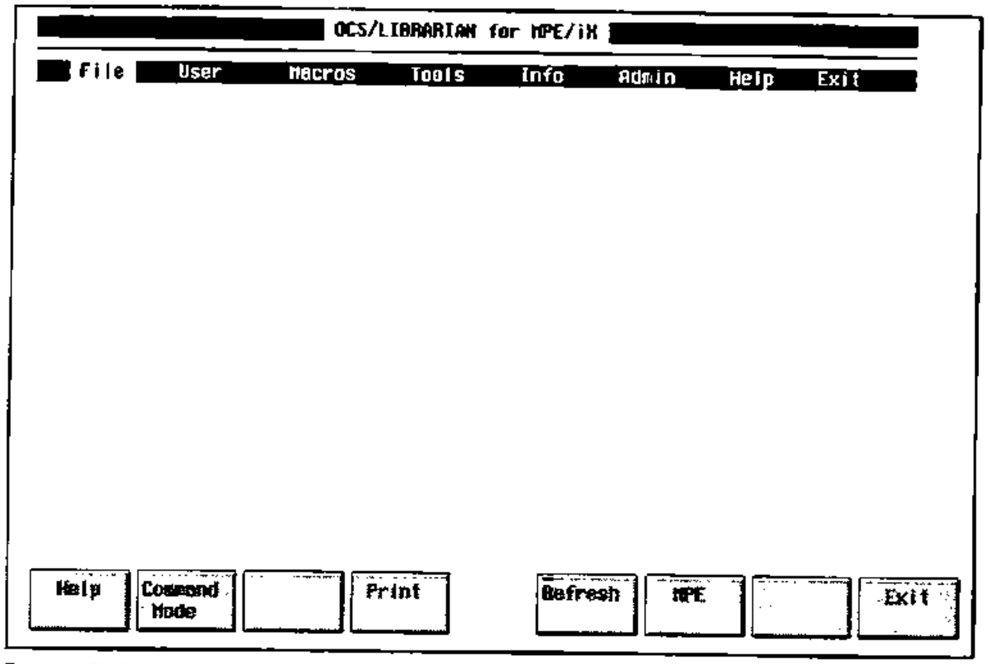
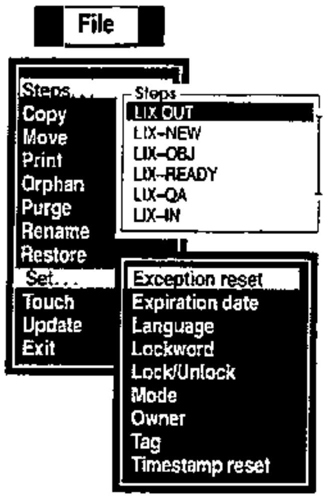


Figure 9–1. LIBRARIAN Main Menu

File Menu

The File menu displays a list of operations that apply to files.

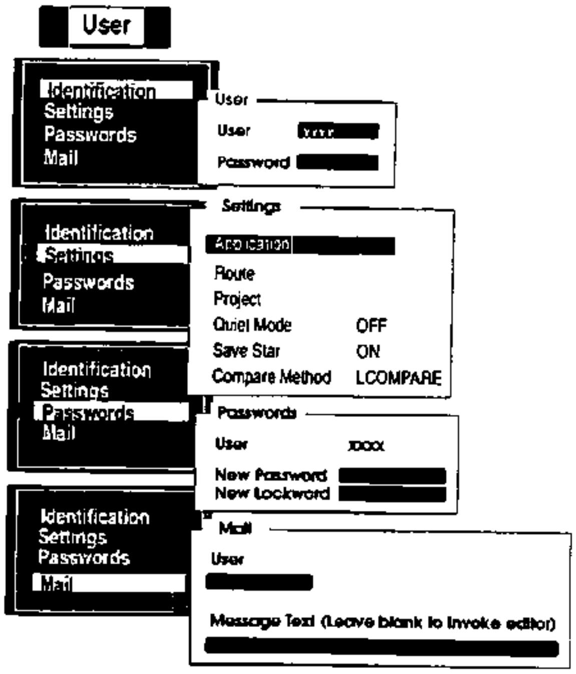
Note Menu options followed by an ellipsis (e.g., **Steps...**) have submenus.



Steps	Open a menu of steps that you are authorized to perform.
Copy	Copy files to a new location.
Move	Move files from one location to another.
Print	Print a file on the screen or a printer.
Orphan	Disable tracking of read–mode secondary files.
Purge	Purge files from the system.
Rename	Rename files or files in a fileset.
Restore	Reconstruct a previous revision of a file making it the most current. Automatically retain the current file before replacing it with the older version.
Set	Displays a menu of file attributes that you can change. These attributes include: **Exception reset** **Mode** **Expiration date** **Owner** **Language** **Tag** **Lockword** **Timestamp reset** **Lock/Unlock**
Update	Update a read–mode secondary with the latest associated master.
Exit	Terminate an active LIBRARIAN session.

User Menu

The **User** menu gives you access to dialogs for viewing and changing the current environment, changing your password, and sending receiving mail.



User
User xxxx
Password

Settings
Application
Route
Project
Quiet Mode OFF
Save Star ON
Compare Method LCOMPARE

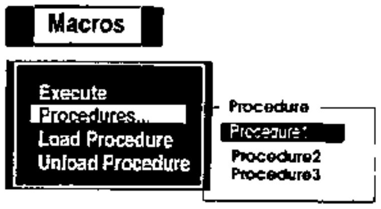
Passwords
User xxxx
New Password
New Lockword

Mail
User
Message Text (Leave blank to invoke editor)

Identification	Enter new user name and password, or show current user.
Settings	Set up the environment for the current LIBRARIAN session. Creates a temporary file called **LIBSET**. If you save this file, LIBRARIAN will automatically load these settings each time you run LIBRARIAN.
Passwords	Change user password/lockword.
Mail	Read mail or send a message to a user or to the audit trail.

Macros Menu

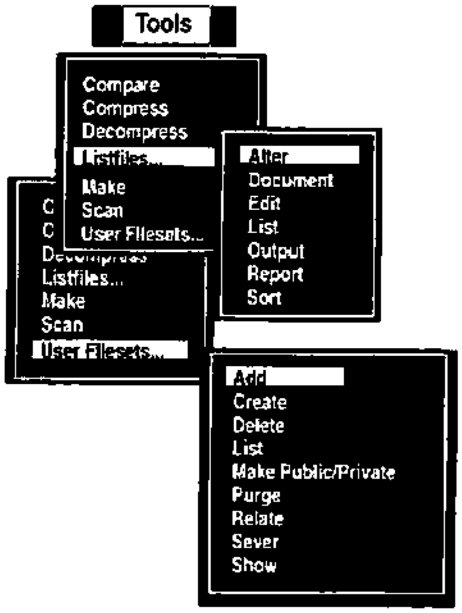
The **Macros** menu displays the options for loading and using LIBRARI-AN macros.



Execute	Execute a specific macro file or procedure.
Procedures	Open a menu of loaded macros and select macro to execute.
Load procedure	Load macros from a procedure file.
Unload procedure	Unload all currently loaded macros.

Tools Menu

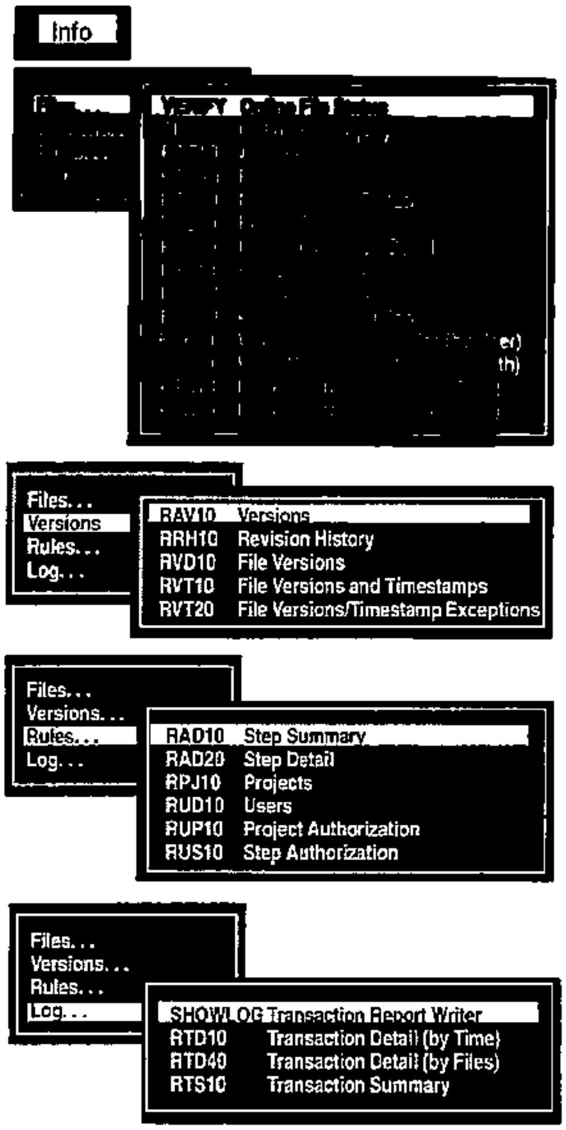
The **Tools** menu displays a list of special LIBRARIAN file utilities.



Compare	Show differences between files or revisions.
Compress	Compress files.
Decompress	Decompress files.
Listfiles	Display a menu of LMAINT operations for managing (indirect files).
Make	Run the MAKE utility to rebuild applications.
Scan	Scans files for strings of text, and optionally replace text.
User Filesets	Display a menu of FMAINT operations that lets you group files according to your own needs.

Info Menu

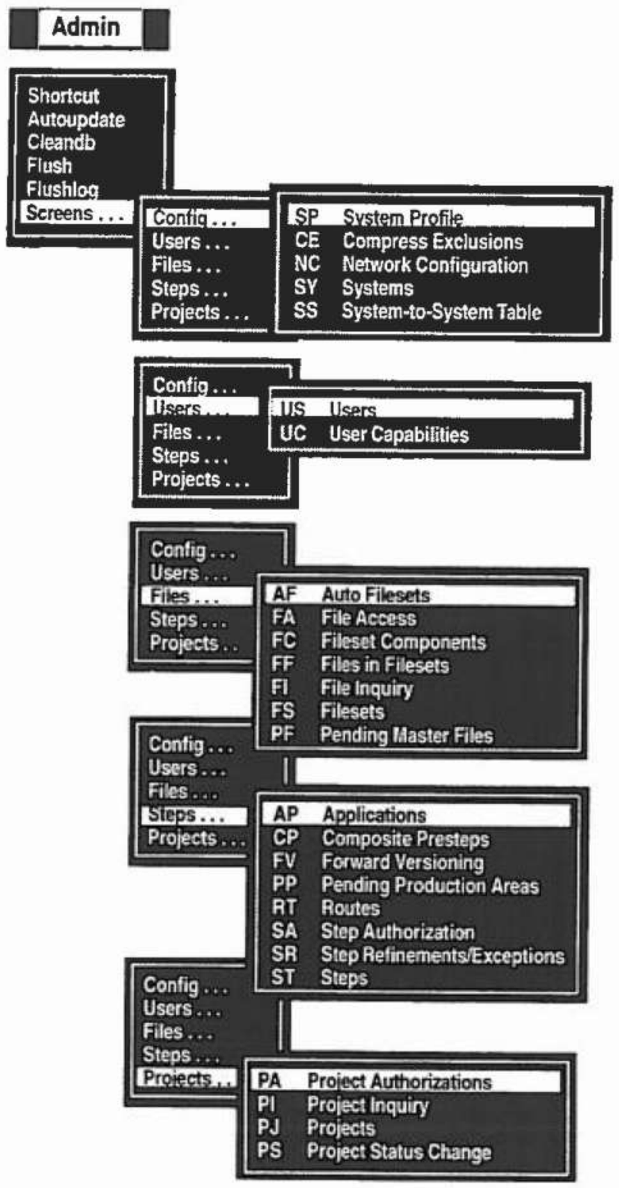
The **Info** menu includes four menus for generating offline reports and online inquiries.



Files	Open the menu of reports and online inquiry screens related to files.
Versions	Open the menu of reports related to versions.
Rules	Open the menu of reports related to steps, projects and user authorizations.
Log	Open the menu of all reports related to the transaction log.

Admin Menu

The **Admin** menu displays utilities for maintaining and updating the LIBRARIAN databases. You can also access data entry screens from this menu to load and fine–tune rules.

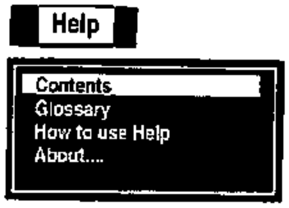


Shortcut	Set up basic LIBRARIAN rules.
Autoupdate	Run the Auto Fileset update utility.
Cleandb	Purge records from the LIBDB tracking database for files that no longer exist on a disk.
Flush	Run the FLUSH utility to purge old files.
Flushlog	Run the FLUSHLOG utility to purge old transaction records.
Screens	Open a menu of screen submenus.

Config	Open a menu of screens for examining, adding, changing, or deleting system and network data in the data base.
Users	Open a menu of screens for defining users and their capabilities.
Files	Open a menu of screens for defining filesets and file attributes.
Steps	Open a menu of screens for adding routes and steps.
Projects	Open a menu of screens for defining and maintaining projects.

Help Menu

The **Help** menu includes a list of options for getting information about LIBRARIAN.



Contents	Open a menu of help topics.
Glossary	Access the glossary of LIBRARIAN terms.
How to use Help	Get information on how to use the help facility.
About	Display copyright, version, and other information about the release of LIBRARIAN you are running.

Dialogs

Many menu options have associated dialog boxes for you to provide further information about an operation. The most common dialog requests a list of files and options for the current transaction. For example, when you select a step from the **Steps** menu, the dialog box shown in Figure 9-2.

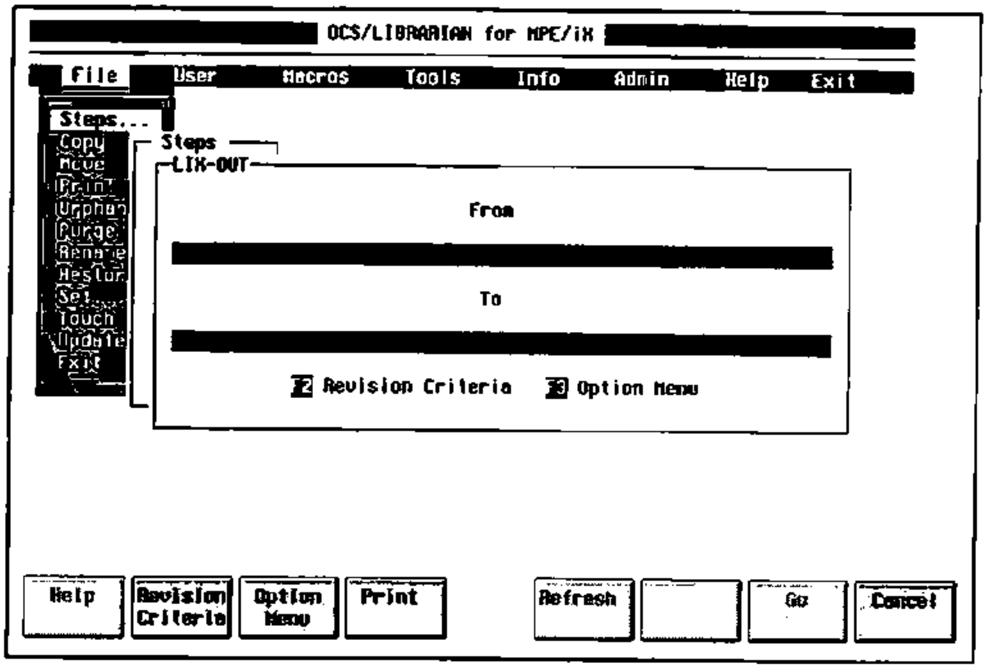


Figure 9–2. Sample Dialog

In this dialog, you can enter a source and destination file. Note that these fields scroll to the left if you type past the end of the field.

You can apply revision criteria to the files listed by pressing **F2**. The Revision Criteria menu appears as shown in Figure 9–3:

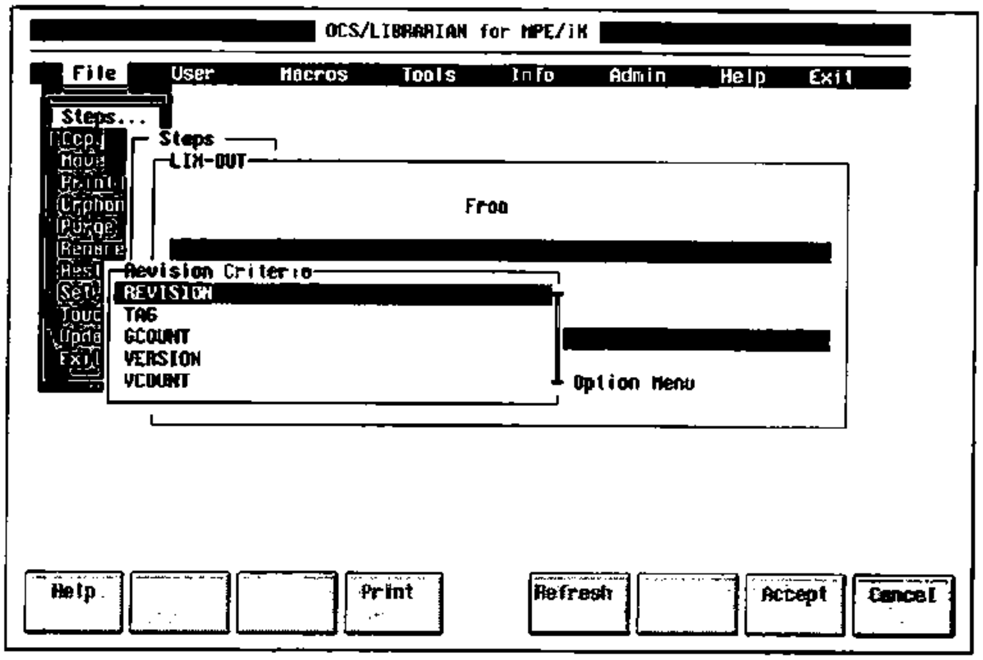


Figure 9-3. Revision Criteria Menu

When you select an option from this menu, a field appears allowing you to specify a value. Press **F8** (Cancel) to leave this menu without accepting the options you selected, or press **F7** (Apply) to leave this menu, applying the options you selected.

You can select step options and override default parameters by pressing **F3** (Option Menu) in the dialog box. A menu of the most common options appears as shown in Figure 9–4.

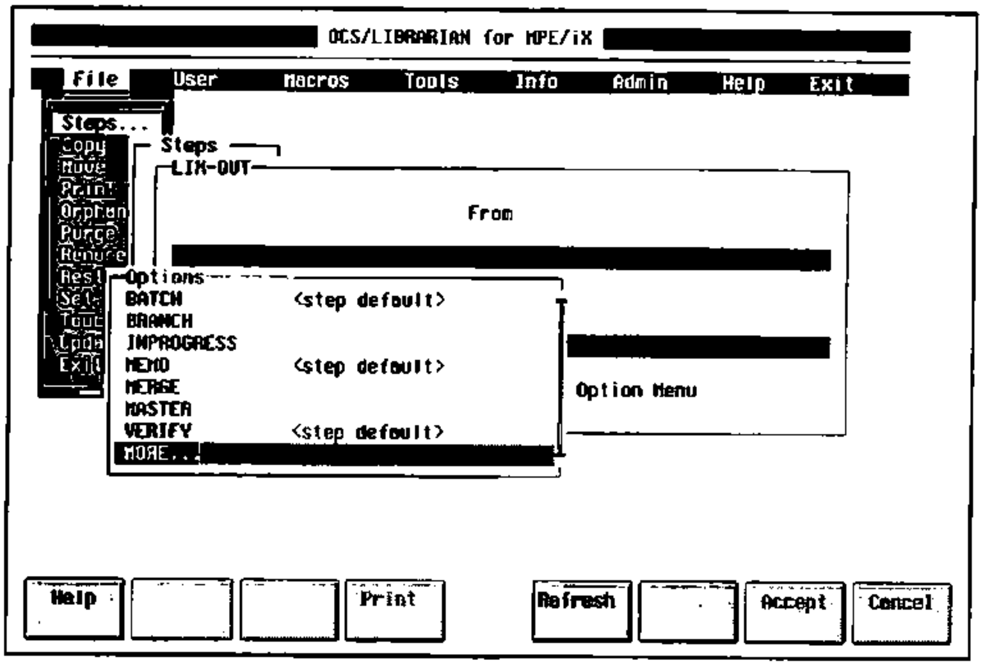


Figure 9-4. Sample Options Menu

Other options are available by selecting **More...** from this menu. When you are finished selecting options and/or setting parameter values, press **F7** to accept the values or **F8** to cancel. You will return to the main dialog box.

After specifying files, revision criteria and/or options, press **F7** (Go) to proceed with the transaction or **F8** (Cancel) to return to the previous menu.

Utility Program 10

LIBRARIAN provides you with the LIBUTILP utility program to perform miscellaneous functions, including globally changing system references in the database. This utility facilitates moving applications or an entire LIBRARIAN implementation to a new system.

The LIBUTILP utility program enables you to do the following:

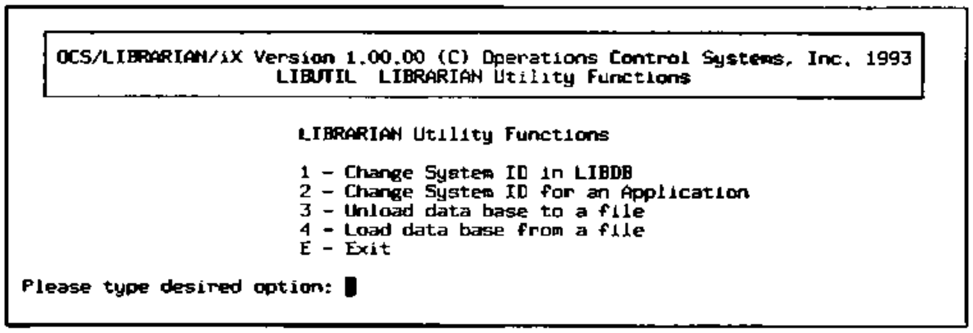
- Change the system ID for all applications.
- Change the system ID for a single application.
- Unload a database to a flat file, @UL.PUB.OCSLIB.
- Load a database from a flat file @UL.PUB.OCSLIB.

Operation

The complete name for the LIBRARIAN utility program is **LIBUTILP.COMP.OCSLIB**. To use LIBUTILP, log on to MGR.OCSLIB on the MPE/iX server and type:

```
:RUN LIBUTILP.COMP.OCSLIB
```

The program presents a menu of options, as displayed in Figure 10–1:



```
OCS/LIBRARIAN/iX Version 1.00.00 (C) Operations Control Systems, Inc. 1993
                   LIBUTIL  LIBRARIAN Utility Functions

                   LIBRARIAN Utility Functions

                   1 - Change System ID in LIBDB
                   2 - Change System ID for an Application
                   3 - Unload data base to a file
                   4 - Load data base from a file
                   E - Exit

Please type desired option: █
```

Figure 10-1. LIBRARIAN Utility Functions Menu

For more information on the LIBUTILP program, refer to Appendix B, "Using the LIBRARIAN Utility Program" in the *LIBRARIAN/iX Administrator's Guide*.

Configuration Program 11

LIBRARIAN provides you with the CONFIGP utility program to perform miscellaneous functions, including updating your LIBRARIAN configuration and changing database passwords.

The LIBRARIAN configuration program, CONFIGP, enables you to perform the following:

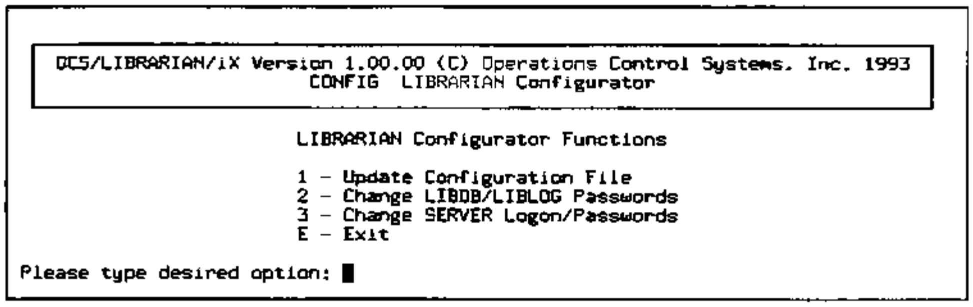
- Update the LIBRARIAN configuration file.
- Change LIBDB/LIBLOG passwords.
- Change server logon and passwords, for client machines.

Operation

The complete name for the LIBRARIAN configuration program is **CONFIGP.COMP.OCSLIB**. To use CONFIGP, log on to MGR.OCSLIB on the MPE/iX server and type:

```
:RUN CONFIGP.COMP.OCSLIB
```

The program presents a menu of options, as displayed in Figure 11–1:



```
 OCS/LIBRARIAN/iX Version 1.00.00 (C) Operations Control Systems, Inc. 1993
                        CONFIG  LIBRARIAN Configurator

                     LIBRARIAN Configurator Functions

                     1 - Update Configuration File
                     2 - Change LIBDB/LIBLOG Passwords
                     3 - Change SERVER Logon/Passwords
                     E - Exit

Please type desired option: █
```

Figure 11–1. LIBRARIAN Configuration Functions Menu

For more information on the CONFIGP program, refer to Appendix C, "Using the LIBRARIAN Configuration Program" in the *LIBRARIAN/iX Administrator's Guide.*

Datasets 12

Dataset capacity requirements are based on factors such as the number of files, users, file movements, and versions. Therefore, capacity requirements vary considerably among installations. The following dataset descriptions will help you estimate dataset sizes for your installation.

For information on database structure, data items, and initial capacities, refer to the schema text files which are loaded during the LIBRARIAN installation:

LIBDB	SCHEMA.PUB.OCSLIB
LIBLOG	LOGSCHEM.PUB.OCSLIB

This chapter describes each of the datasets which comprise the two LIBRARIAN databases:

- LIBDB Database
- LIBLOG Database

LIBDB Database

The LIBDB database contains the following datasets:

M-APPLICATION	Application names and associated master filesets. One record per application. Maintained by the Applications (AP) screen.
M-FILE-SET	Fileset names and descriptions. One record per fileset. Maintained by the Filesets (FS) and Projects (PR) screens and by FMAINT commands.
M-USER	User names, user data, and passwords. One record per user. Maintained by the Users (US) screen.
D-AHFSET-COMP	Component (hierarchy) definitions for user-defined filesets. One record for each component for each fileset. Maintained by the FMAINT subsystem of LIBRARIAN.
D-AHFSET-FILE	Filenames within a user-defined fileset. One record per file per fileset. Maintained by FMAINT.

D-APPL-VERSION	Versions for an application. One record per version, per application. Maintained by the **VERSION** command. Fileset is included in this record because the master fileset definition for the application can change between versions. The fileset recorded here is the fileset as it was at the time the version was established.
D-AUTO-FSET	Wildcard descriptors used for including files including/excluding files from this fileset. One record for each descriptor, for each fileset. Used by the Auto Update (AUTOUPDP) program to add new files automatically to fileset when introduced. Maintained by the Auto Filesets (AF) screen.
D-COM-LINKS	Defines the actual device names for communicating between systems, if the device name differs from the system name. Maintained by the System-to-System Table (SS) screen.
D-FILE	Contains most of the file-specific information in the database. Contains statistics on number of generations, versions, copies out, last route/step performed, etc. One record per revision per file defined to LIBRARIAN. Maintained by the File Access (FA), Files in Filesets (FF), Pending Master Files (PF) screens, and updated by LIBRARIAN and other utilities.
D-FILE-TABLE	All filenames in LIBRARIAN are tracked by tokens. This dataset is a cross-reference between the token and the actual filename.
D-FORWARD-VER	Defines previous version locations for forward version search. One record per search level. Maintained by the Forward Versioning (FV) screen.
D-FSET-COMPONENT	Defines fileset hierarchy. One record for each component fileset. Maintained by the Fileset Components (FC) screen.
D-FSET-FILE	Contains the filenames within a master fileset. One record per file per fileset. Maintained by the Files-in-Fileset (FF) screen and **AUTOUPDATE**.

D-MESSAGES	Contains messages sent to users, using the **MAIL** and **MEMO** commands. One record per each line of each message.
D-NETWORK-CONFIG	Contains one record for defining the type of network and default logon information for connecting to remote machines. Maintained by the Network Configuration (NC) screen.
D-NO-COMPRESS	Contains one record. Array of 40 filecodes to be excluded from compression. Maintained by the Compress Exclusions (CE) screen.
D-PENDING-AREA	Defines areas from which new files can be introduced. One record per area per step. Maintained by the Pending Production Areas (PP) screen.
D-PRESTEPS	Defines composite presteps for a step. A maximum of 16 presteps can be defined for a composite prestep. Maintained by the Composite Presteps (CP) screen.
D-PROJECTS	Defines projects and specifies the routes for which each project is valid. Maintained by the Projects (PR) and Project Status Change (PS) screens.
D-REFINED-STEP	Defines different destinations and/or movement types for subsets of the source location defined in the related D-STEP record. One record per refinement defined. Maintained by the Step Refinements/Exceptions (SR) screen.
D-ROUTE	Defines routes within an application. One record per route. Maintained by the Routes (RT) screen.
D-STEP	Contains the basic definition of a step, including the source and destination locations, type of movement, prestep, etc. One record per defined step. Maintained by the Steps (ST) screen.
D-SYSTEM-ID	Defines system data including login with appropriate passwords. One record per system. Maintained by the Systems (SY) screen.
D-SYSTEM-PROFILE	Defines global parameters for the LIBRARIAN installation. One record. Maintained by the System Profile (SP) screen.

D-USER-CAPS	Defines users with special capabilities. One record per capability per user. Maintained by the User Capabilities (UC) screen.
D-USER-FSET	Contains one record for each user fileset and it's creator, establishing a cross-reference between users and their filesets. Maintained by FMAINT.
D-USER-PROJECT	Defines user project authorizations. One record per authorized user per project. Maintained by the Project Authorizations (PA) screen.
D-USER-STEP	Defines user step authorizations. One record per authorized user per step. Maintained by the Step Authorization (SA) screen.
D-VFSET-COMP	Retains D-FSET-COMPONENT data at the time a new version is established. Contains fileset components. Maintained by the **VERSION** command. Number of records is approximately number of D-FSET-COMPONENT records for an application multiplied by number of versions.
D-VFSET-FILE	Retains D-FSET-FILE data at the time a new version is established. Contains fileset members. Maintained by the **VERSION** command. Number of records is approximately the number of D-FSET-FILE records for an application multiplied by the number of versions.

LIBLOG Database

The LIBLOG database contains the following datasets:

D-FILE-TABLE	All filenames in LIBRARIAN are tracked by tokens. This dataset is a cross-reference between the token and the actual filename.
D-SYSTEM-PROFILE	Defines the next file ID to be used. One record.
D-TRANSUM	Contains a log of transactions performed. One record for each transaction. New records are added automatically by LIBRARIAN when logging is turned on.

D-TRANSDTL	Contains detail of file operations. One record for each file within a transaction. New records are added automatically by LIBRARIAN.
D-MEMO	Contains memo text related to a transaction. Related to D-TRANSUM and D-TRANSDTL by internal TRANS-ID. One record per line of text. New records are added automatically by LIBRARIAN.

Flush log records from the LIBLOG database by using the FLUSH utility or **SHOWLOG>FLUSH**.

Note

To check LIBDB and LIBLOG dataset capacities, use the LIBRARIAN command **CHECKDB**.

LIBRARIAN/iX Glossary of Terms

Note Terms that appear in *italics* in the following definitions have separate glossary entries.

A

Access Control

The attribute of a *master file* that determines how many *read/write mode* copies are allowed. The four access control levels are: *exclusive, read only, serial write, and multiwrite.*

Access Mode

The attribute of a *secondary file* that determines whether or not it can be checked in and replace its associated *master file*. A secondary in *write mode* can replace a master. A *read mode* can only replace a master through an *emergency checkin* that is configured to use the *PUSHREAD* parameter. A file's access mode is determined by *access control*, user request, *step* definition, and *default access mode* (precedence is in order listed).

Aging Policy

A *system profile* value that indicates how long log records are kept. When the *FLUSHLOG* utility is run, audit trail records that are older than the number of days specified in the aging policy are deleted.

Transactions associated with projects override this policy and are deleted only when the project status is flush pending.

Alternate prestep

A *prestep* that can be performed as an alternative to the defined prestep. Up to three alternatives can be defined for a *step*.

Annotate

Comments inserted by LIBRARIAN into source listings that indicate which lines were inserted/deleted for which *revision*. Date/time, related project and user who made the change are included.

Application

A site–defined organizational unit including a set of *master files* that are being controlled by LIBRARIAN, a set of *steps* for file movement/approval, and, optionally, a set of *projects* for tracking file changes associated with a particular work activity.

Application Manager

A *special user capability* assigned to the user responsible for the files and *steps* within an *application*.

Application fileset

The highest level *fileset* for an *application.*

Approval step

A *null step* that is required as a prerequisite for a subsequent step.

Authorization

The process of determining which files have been requested in a *transaction* and whether or not the rules permit the operation to be performed on each of these files. Authorization is based on the user who initiated the request and the current status of each file requested.

AUTOXEQ file

A *macro* that is executed before the first prompt/main menu appears. A file called AUTOXEQ that exists in the product account is executed prior to any AUTOXEQ file that might exist in the user's home directory.

Auto fileset descriptors

General locations that describe how *master files* are assigned automatically to *master filesets.* Descriptors can include or exclude files from filesets using *wildcards.* When you run *AUTOUPDATE,* introduce new files with a *pending master,* or perform a *checkin step* with the AUTOUPDATE parameter turned on, any previously *untracked files* in these locations get added to the appropriate master filesets.

Automatic Login ID

The login used when transactions require automatic logging in to a remote system.

Autoupdate

The process used to add *master files* to *master filesets* automatically based on predefined *auto fileset descriptors* that include or exclude files from filesets, typically using wildcards. *Pending masters* and masters not currently assigned to required filesets are added, typically during *checkin, new steps* and/or running of the AUTOUPDATE utility.

B

Baseline

The *master library* at a particular point in time. An *application manager* establishes a baseline by creating a *version.* This marks and protects all of the files in an application at that time, so that the application or any part of the application can be restored to that baseline any time in the future.

Base Revision

A *revision* that was current at the time a *baseline version* was created. The *version count (VCOUNT)* for a base revision is always zero and cannot be flushed until the version(s) of which it is a part is made *obsolete.*

Branch

A set of *revisions* that are made as a divergence from the main development path for a master file. A branch is created automatically when a previous revision is checked out. A branch can also be forced from the latest revision if the master is already checked out in *write mode,* or the user does not intend to check the file back in on the *trunk.* Whenever a new branch is created, a branch counter and *leaf* counter (both starting at 1) are appended as a pair to the original *revision ID.*

Branch revision

A *revision* that appears on a *branch.*

C

Checkin step

Any *step* which copies or moves a file from a *secondary location* into the *master library,* either retaining and replacing the existing master, introducing a new one or establishing a new branch .

Checkout step

Any *step* which copies a file from the *master library* into a *secondary location,* generally for modification by programmers.

Client

An MPE or UNIX implementation of LIBRARIAN where the LIBRARIAN data bases reside on a different system, but the user is able to perform all LIBRARIAN functions.

Command Mode

In command mode, the user enters LIBRARIAN commands at a command line prompt. Users can switch between command mode and *menu mode* by pressing the F2 function key.

Component filesets

Filesets that are subsets of higher–level filesets.

Composite prestep

A collection of *presteps* that must be performed before a subsequent step can be performed. Composite presteps also permit the specification of a date prerequisite.

D

Default access mode

The *access mode* that is assigned to a *secondary file* when neither the user or *step* explicitly specify the mode. The *access control level* for a file determines which access modes are allowed.

Delta file

A privileged (MPE) or hidden (UNIX) file that contains the history of changes made to an associated *master file*.

Deltas

A method for retaining and reconstructing previous revisions of *master files* that involves storing only the changes to files over time.

Dependency

A file that *make* evaluates with respect to some target to determine whether to invoke some action, such as a compile or link.

Destination

The target location when copying or moving a file.

Dummy target

A *make target* that does not correspond to an actual file. *Dependencies* of dummy targets are actual files that are always evaluated as targets themselves to determine whether they are out of date and need to be rebuilt.

E

Edit mask

A file expression that uses special editing characters to map one filename into another; e.g., source to destination name for a copy or move or *secondary* to *pending master name* for introduction of a new file.

Emergency checkin

A checkin that moves a *read mode secondary file* into the *library* with the *PUSHREAD* option. If a *write mode* copy exists, the *owner* is notified via a LIBRARIAN *mail* message, and an *exception* is recorded.

Exception Flag

An indicator that something special has happened related to a file such as an *emergency checkin, merge conflict* or previous *master revision* was restored at a time when the file was checked out. The exception flag must be cleared before any further operation on the file is allowed.

Exception message

A LIBRARIAN *mail* message that indicates that an exception flag has been placed on a file. This message is sent to the *owner* of the *write mode* copy of the file.

Exclusive access

The *access control level* that prevents *secondary* copies of a *master file* from being made.

Expiration date

The date when after which a file can be flushed using the *FLUSH* utility.

Expired file

A *read mode secondary* or *retained file* that is eligible to be flushed by the *FLUSH* utility.

Explosion

The creation of a list of files by expanding a *fileset, listfile,* or *wildcard* file specification for LIBRARIAN to *authorize*.

External

A file that resides on a system on which LIBRARIAN is not running, typically an unsupported platform, or system which is not on an accessible network. LIBRARIAN steps can be used to record movement to an external location, but cannot physically move the file or verify its existence. Users are responsible for transferring files (via tape or other means) for any transaction using the EXTERNAL option.

F

Fileset

A collection of files identified by a unique name assigned by the *Librarian Manager* (*master filesets*) or any user (*user filesets*). When requesting files, filesets can be referenced by preceding the fileset name with a percent sign (%). Because filesets contain collections of files that are related by some criteria other than physical location, and can span directories and systems, they are often referred to as *logical filesets*.

Note: In MPE, a fileset is any set of files that can be referred to using wildcards in name, group and/or account. LIBRARIAN refers to this as a *physical fileset*.

File structure (hierarchy)

The relationship of filesets, subsets and physical files within an application library.

Flush policy

The *system profile* policy that determines how many previous file *generations* to keep when the *FLUSH* maintenance utility is run.

FLUSHLOG

The maintenance utility that purges old log records that have aged beyond the *aging policy* specified in the *system profile*.

FLUSH

The maintenance utility that purges *expired files* and *obsolete versions*.

Flushed project

When a project is closed and then assigned a status of flush pending, log records associated with that project get flushed the next time the *FLUSHLOG* utility is run. After FLUSHLOG has been run, the project status is changed to flush, and the project can be deleted, if desired.

Flushed version

When a *version's* status has been changed to *obsolete,* base *revision* files that are a part of that version are flushed if they are not also part of a subsequent version. After *FLUSH* has been run, the version status is changed to flush, and the version can be deleted, if desired.

Flush pending

A *project status* that indicates that log records for the *project* should be purged when the *FLUSHLOG* utility is run.

FMAINT

The facility for creating and maintaining *user filesets.*

Forward versioning

An option on *checkout* to automatically search alternate *libraries* (usually previous versions) when a *master file* is not found in the expected *location* as defined by the checkout step. If the file is then found in an alternate location, it is brought forward as a *secondary* of a new *pending master* for the primary *application.*

G

Generation

Each time a file is checked in, a new generation is created. Previous generations of *master files* are stored in the *library* as *retained files* (usually compressed) or as *deltas.*

Generation count (GCOUNT)

A sequential number assigned to each *master file generation.* The current GCOUNT is the total number of times a master file has been replaced. When specifying GCOUNT as an option in a file request, a negative number indicates a generation relative to the latest generation.

Generic rule

A *target–dependency* relationship in *make* that uses *wildcards* (target) and edit masks (dependency) to determine what is out of date. Actual target and dependency names are substituted into the rebuild commands using *make macros.*

I

Indirect file

Also called a *listfile,* an indirect file is a text file that includes a list of filenames. This file can be used in *LIBRARIAN* commands as a convenient way of referencing files. Indirect files can be created in a text editor or through *LIBRARIAN's LMAINT* facility.

INPROGRESS

A parameter used with a *checkout step* that instructs LIBRARIAN to record the existence of a *write mode secondary* without physically copying the file from the *library*. This parameter is most often used when LIBRARIAN is initially implemented and some files are already being worked on or tested.

Intermediate revision

Master files that are retained between versions. The version count (VCOUNT) for intermediate revisions is always greater than 0.

L

Leaf Revision

Each *revision* on a *branch* is called a leaf, sequentially numbered from the start of the branch. Whenever a new branch is created, a branch counter and leaf counter (both starting at 1) are appended as a pair to the original *revision ID*.

LIBRARIAN

The program that controls and processes all file operations maintaining an audit trail of activity.

LIBRARIAN Manager

A *special user capability* assigned to the person responsible for configuring LIBRARIAN and defining site rules. The LIBRARIAN Manager has unrestricted access to all LIBRARIAN functions for all files.

Library

A library is the repository from which files are *checked out*, and to which they are subsequently *checked in*. Files are also distributed to production locations from the library. It is the 'official' collection of files that are under LIBRARIAN's control. Files in the library are called *master files*. The library provides a central point of control for changes to production source, object and data.

Listfiles

Also called an *indirect file*, a listfile is a text file that includes a list of filenames. This file can be used in *LIBRARIAN* commands as a convenient way of referencing files. Listfiles can be created in a text editor or through *LIBRARIAN's LMAINT* facility.

LMAINT

The facility for creating and maintaining *listfiles (indirect files)*.

Location

The group/account (MPE) or directory (UNIX) and *system* where a file exists or should be created.

Logical fileset

A meaningful name assigned to a collection of files not bound by physical boundaries. See *fileset*.

!LOGON, !LOGIN

A special wildcard that can be used in defining step source and destination *locations* to indicate that the user's login data should be substituted as appropriate. For MPE, this wildcard can be used for group, account and/or *system*. For UNIX, this wildcard is equivalent to '.' for current working directory and can also be used for *system*.

M

Macro

A set of *LIBRARIAN* and operating system commands for LIBRARIAN to execute. A macro *control language* provides programmatic control (conditions and loops) and parameter substitution. Parameter values can be system-defined or provided by the user via prompts and/or customized menus. Macros are analogous to MPE command files and UNIX scripts. Multiple macros can be combined in a single *procedure file*. Macros are also referred to as *XEQ files*.

Macro Control Language

The set of special commands and keywords that are used in macros to control flow of execution (IF...THEN...ELSE, REPEAT,

WHILE, LOOP, GOTO) and allow for parameter substitution (tokens preceded by %%).

Mail

Mail includes messages that are sent from one LIBRARIAN user to another, or from *LIBRARIAN* notifying a user that an *exception* condition has occurred that affects that user's work.

Make

A utility that automatically rebuilds/recompiles components of an *application* when they change. Make reads a *makefile* that shows *dependencies* between application components and evaluates which components are out of date. Based on which components are out of date, make issues only the commands necessary to bring the application up to date.

Makefile

A text file that contains make rules. This file can have any name and can be created and maintained using any text editor. This file includes *target-dependency* relationships and commands required to bring each target up to date whenever their dependencies are changed. *Make macros* and *generic rules* can be used to reduce the size and complexity of a makefile.

Make macros

A shorthand that simplifies creating *makefiles*. Macro references are substituted with either user–defined or system–defined values when the makefile is processed. For example, out–of–date *dependency* names can be substituted in generic command descriptions.

Master file

A file that is part of a defined *library* and reflects the most current production version.

Master fileset

A *fileset* defined by the LIBRARIAN Manager that includes *library* files.

Master library

The hierarchy of *master filesets* and associated *master files* for an *application*.

Memo

Text that provides documentation for a *transaction*. Memos are stored in the audit trail database and can be reviewed using *SHOWLOG*.

Menu Mode

The mode of *LIBRARIAN* operation in which users select LIBRARIAN functions from a set of pull–down menus. Users can switch to the command line prompt at any time by pressing the F2 function key.

Merge

An option available on *checkout steps* to combine source code changes from one or more *branches*. Conflicting changes are highlighted with comments in the source code, and should be resolved prior to the next step. Merge is only available if the *delta* feature is being used.

!MSUSER

A special *wildcard* that can be used in defining step destination *locations*. When the step is executed, the wildcard is replaced with the user ID of the user who originally checked out the file. For MPE, this wildcard can be used to fill in group or account. For UNIX, this wildcard can appear anywhere in the path name. This wildcard is typically used to reject files and move them from a test area back to the appropriate developer's work area.

Multi–write

The *access control* level that allows multiple *secondary files* with *write–mode* access.

N

New step

A *step* that introduces a previously *untracked file* to *LIBRARIAN* as a *secondary file*. The file is linked to a pre–existing *master file* or a *pending master* record is created. Rules governing introduction of new files on a step are configured on the PP (Pending Production Areas) screen.

Node

The actual device name associated with a system in a network. This name may or may not be the same as the LIBRARIAN *system ID*.

Null step

A *step* not involving any file movement. A null step is used to reflect some external action such as an approval. Null steps are used to control dependencies between steps; that is, they are used as *presteps*.

O

Obsolete version

When the *LIBRARIAN Manager* or *Application Manager* change the status of a *version* to obsolete, any *retained base revisions* associated with that version will be flushed the next time the *FLUSH* utility is run. Once a version is flushed, it can be deleted, if desired.

Operator

A *special capability* assigned to a user who can *flush* records in the log database and can restore previous revisions of files.

Orphan

Any file not currently being tracked by *LIBRARIAN* or a *master file* not associated with an *application*. Orphans can be created by a LIBRARIAN operation that causes a tracked file to become untracked (unknown to LIBRARIAN), or by operations that use the orphan option to create files in destinations that are not to be tracked.

!OWNER

A special *wildcard* that can be used in defining step destination *locations*. When the step is executed, the wildcard is replaced with the user ID of the user who currently owns the file. For MPE, this wildcard can be used to fill in group or account. For UNIX, this wildcard can appear anywhere in the path name. This wildcard is typically used to approve files in multiple developer work areas.

P

Parent Fileset

A *fileset* that includes *component* filesets.

Pending master file

A file that is being *tracked* as a *master library file*, but, because it is new, does not physically exist in the *library* yet. The associated *secondary* is called a *pending production file* and was introduced through a *new step* or through the use of LIBRARIAN's *forward versioning* feature.

Pending master mask

An *edit mask* used to automatically derive a *pending master file* name based on the name of the *secondary file* being introduced through a *new step*.

Pending production area

Any *location*(s) defined for a *step* where previously *untracked files* can be introduced as new *secondary files*. Steps with pending production areas are considered to be *new steps*.

Pending production file

A *secondary file* that was introduced using a *new step*. The *master file* does not currently exist in the *library*.

Permissions

A UNIX term used to indicate file access rights; a matrix of read, write, and execute access for owner, group and world.

Physical fileset

A collection of files that exist in a particular *location*. Physical fileset references include specific filenames, or names using standard operating system/shell *wildcards*.

Prestep

A *step* that must be completed successfully for a file before the next step in the *route* can be performed. Presteps are often *null approval steps*.

Procedure

A *macro* that is included in a file with other macros with a procedure header.

Procedure file

A file that contains multiple *macros*. Each macro has a procedure header indicating the name of the macro. Procedure files can be loaded and unloaded while using *LIBRARIAN*.

Project

A way of organizing *transactions* and associated files with a specific work activity.

Project fileset

A *user fileset* that is created automatically when defining a *project*. The fileset is maintained automatically when files are *checked out* or introduced as new files for the project. Files can also be added to this fileset in advance by a *Project Manager* using the *FMAINT* facility.

Project manager

A *special user capability* assigned to users who can create projects, modify project status and authorize users to work on projects.

Project menu

Whenever *projects* are associated with a particular *route*, users are asked to select the project that they are working on from a menu when checking files out or introducing new files.

Project status

A flag that determines what activities can be associated with a *project*.

PUSHREAD

A *step* option which allows a *read mode* copy to replace a *master file* or *write mode secondary* which has not been checked in yet. This option is typically used for *emergency steps*.

R

Read mode

The attribute of a *secondary file* that indicates it cannot replace the *master*. Read mode copies expire after a configured period of time and can be flushed using the *FLUSH* utility.

Read only

An *access control* level that only allows *read mode* copies of a file.

Read step

A *step* that copies a *master file* to a *secondary* location in *read mode*, with no intention for modification. An *expiration* policy can be applied, so that read mode copies created by the step can be cleaned up automatically with the *FLUSH* utility.

Receiver

A *system* that can receive files from other systems, but from which *LIBRARIAN transactions* cannot be initiated.

Release Step

Similar to a *read step*, a release step copies files from the *library* to a production *location* in *read mode*. Typically, these files do not expire, and the previous version is often *retained*.

Retained file

A previous *generation* of a file saved under a *LIBRARIAN*–generated name "G#######". Files are retained when the retain parameter is used on a *step* and the destination file is a *tracked master* or *secondary file*. *Base revisions* are always retained. If *deltas* are being used, changes to the previous generations are stored.

Revision

Any set of changes made to a *master file* through a *checkin* step. Revisions include all *generations* of a master file including the most current. *Leaves* and *branches* also make up the set of revisions for a file.

Revision ID

Revisions are identified by version name followed by a colon (:) followed by version count. If the revision is on a branch, branch and leaf count pairs are appended delimited with periods (.).

Route

A set of automated procedural controls for managing file changes and distribution. A route consists of a predefined file–movement path that reflects an established cycle. The route includes *steps* for all allowable movements of the files for that cycle.

Route Alias

When defining *projects*, a route alias can be defined to indicate that the project only applies to a particular *route*. The project name can be used in place of the route name when performing a step (i.e., step.project) to bypass the *project menu*.

Rule Administrator

Similar to the *LIBRARIAN Manager*, the Rule Administrator is a user with *special user capability* who can define *LIBRARIAN* rules such as steps and filesets, but is not automatically authorized to perform LIBRARIAN functions, and cannot create *user authorizations*.

S

Scan/Replace

A *LIBRARIAN* function that searches files for patterns of text, and optionally replaces the matches with user–defined text.

Scope

The attribute of a *step* that restricts which files the user can request. When copying or moving files, the scope specifies where files come from and where they can be copied. Steps can restrict by fileset, from location and to location.

Secondary file

Any copy of a *master file* or another secondary file. All secondaries are linked to a master (or *pending master*) either directly or indirectly, and are in *read or write mode*.

Secondary location

Any *location* where *secondary files* can be created.

Serial write

The *access control* level that allows only one *secondary file* at a time to have *write mode* access, preventing concurrent modifications.

Server

A system that has an implementation of *LIBRARIAN* which includes the LIBRARIAN databases. *Clients* access this database and other LIBRARIAN functions remotely.

Settings

LIBRARIAN session-level parameters that control the user's working environment.

Special user capability

See *user capabilities*.

Standard Rule

A *make* rule that associates specific *target*(s) with specific *dependencies*.

Step

A rule governing the copying and moving of files from one *location* to another. Steps are the basic building blocks of the *LIBRARIAN* file movement and control system. Steps are grouped into *routes* and are performed using system– and/or site–defined names.

Step parameter defaults

Options that control the behavior of a step, by default.

Step parameter overrides

If allowed, users can override *step parameter defaults* by specifying desired overrides.

Step refinements/exceptions

A *step* definition that includes rules for altering the destination *location* based on the from location, filecode (MPE), and/or *fileset* membership. The same criteria can be used to alter the type of movement (copy, move or null) or exclude files altogether from the step.

Step type

There are three types of steps: master–to–secondary (MS), secondary–to–secondary (SS) and secondary–to–master (SM). MS steps are steps that checkout or distribute files. SM steps are steps that check files in. SS steps encompass all steps in between, such as move to test and approvals.

System

A unique *node* within a network identified to *LIBRARIAN* with a unique *system ID*.

System ID

Used to *identify* systems to *LIBRARIAN* within a network. Optionally appears as a prefix to a filename delimited by ':' to indicate the appropriate system.

System Profile

A set of global parameters maintained by the *LIBRARIAN manager* that control how LIBRARIAN operates. Includes items such as flush policy, aging policy, date formats, etc.

T

Tag

A user–defined name for a particular *revision* of a file or files that can be used to identify them at a later time, even after they have been *retained.*

Target

Component of a make rule that is built from one or more dependencies using one or more commands. Object code and executables are examples of targets.

Tracked file

A file for which there is a record in the LIBRARIAN data base. Tracked files are *masters, secondaries* or *retained files* and movement operations are controlled by *LIBRARIAN* rules. All other files are *untracked files.*

Transaction

Any LIBRARIAN operation attempted either successfully or unsuccessfully on a set of files. Except for commands which provide information, all transactions are logged in the LIBRARIAN audit trail.

Trunk revision

A *revision* that is not checked in on a *branch.*

U

Untracked file

A file for which there is no record in the *LIBRARIAN* database. Ad hoc operations on these files conform to normal operating system security. Steps cannot be performed for untracked files.

User authorizations

The mechanism for determining who can do what. Authorizations can be defined for *steps* and *projects. Special user capabilities* can be assigned so that specific authorization is not required in some cases.

User capabilities

Grants users certain privileges that transcend standard *user authorizations*. These include *LIBRARIAN Manager, Application Manager, Project Manager, Operator, Rule Administrator* and *X capability*. If no special capability is assigned, authorization is required for steps, and other commands conform to normal operating system security.

User fileset

A *fileset* created and maintained by a user through the *FMAINT* user fileset module. User filesets allow users to group files for their convenience. Like *master filesets*, precede user filesets with % when referencing them in commands.

!USERID

A special *wildcard* that can be used in defining step source and destination *locations*. When the step is executed, the wildcard is replaced with the user ID of the user performing the step. For MPE, this wildcard can be used to fill in group or account. For UNIX, this wildcard can appear anywhere in the path name. This wildcard is typically used to check out file's into the developer's work area.

User ID

A unique identifier for a LIBRARIAN user that is password protected. Users are prompted for their User ID when initiating the *LIBRARIAN* program.

User password

Used to protect against unauthorized use of the *LIBRARIAN* system. Passwords are required and can be changed by the individual users.

V

Verify

The *LIBRARIAN* facility for reviewing file information on–line or off–line.

Version count (VCOUNT)

The sequential number that tracks the number of *generations* since the current *version* was defined.

Version

All the files in an *application*, as they were at a specific point in time.

Version ID

The name given to a version by a *LIBRARIAN* or *Application Manager.*

W

Wildcards

Special characters or tokens used in filenames to request multiple files that match a pattern, and/or to determine destination *locations*.

Work-in-progress

Untracked files that were in development and/or test prior to LIBRARIAN implementation. These files can be handled using the INPROGRESS parameter with a checkout step.

Write mode

The attribute of a *secondary file* indicating that it can replace its *master file* through an authorized *checkin step.*

X

XEQ file

A text file that contains the commands for a single *macro*. These macros are executed by filename.

Index