

Using Natural Libraries

This chapter covers the following topics:

- General Information
- Steplibs
- Search Sequence for Object Execution
- Logging on to a Library

For more information on libraries, see *SYSMAIN Utility* in the *Tools and Utilities* documentation.

General Information

Libraries are used to store Natural objects such as programs, subprograms, subroutines, help routines or text.

Some libraries are delivered with Natural, such as libraries beginning with the prefix "SYS". The library `SYSTEM`, for example, contains system-related programs, error messages and DDMs (data definition modules).

When you first logon to Natural, your default library will in most cases be `SYSTEM`. Since this library generally contains system-related objects, you will probably want to use a different library to store your programs. You can add your own libraries to the Natural environment or use libraries set up for this purpose.

Note:

You can alter your default library from `SYSTEM` to any other library by specifying the Natural profile parameter `INIT-LIB` in the Configuration Utility.

Steplibs

A steplib is a Natural user library or system library that is concatenated with the current user or system library. This avoids redundant storage of identical objects and helps organize applications. In addition, if Natural Security is installed, a steplib can be used to restrict access to particular objects.

Natural searches in a steplib when an object is not found in the current library (see *Search Sequence for Object Execution* for further information). The standard steplibs are the libraries `SYSTEM` in the `FUSER` and the `FNAT` system files.

If Natural Security is active, you can define additional steplibs in the security profile of each library. The entries in a library security profile override any definitions made outside Natural Security.

If Natural Security is not active, you can specify additional steplibs in the Configuration Utility, with the profile parameter `STEPLIB`. In addition, you can define further steplibs by using one of the application programming interfaces (for example, `USR1025N` or `USR3025N`) that are supplied for this purpose in the Natural system library `SYSEXT`.

The additional steplibs are searched for an object before the standard steplibs SYSTEM (FUSER and FNAT).

The currently active steplibs are shown when you enter the system command TECH.

Search Sequence for Object Execution

This section describes the sequence in which Natural libraries and system files are searched for a requested object that is to be executed from either a user library or a system library.

Note:

If the profile parameter BPSFI is set to "ON" (the default setting is "OFF"), objects are searched for in the buffer pool first.

The search sequence for a user-written object to be executed from a user library is as follows:

1. The current library in the FUSER system file as defined by the system variable *LIBRARY-ID.
2. The steplibs (in sequence) as specified in the Natural Security profile for the current library or in the steplib table.
3. The default steplib as defined by the system variable *STEPLIB.
4. The library SYSTEM in the FUSER system file.
5. The library SYSTEM in the FNAT system file.

The search sequence for a Natural object to be executed from a system library is as follows:

1. The current "SYS" library in the FNAT system file as defined by the system variable *LIBRARY-ID.
2. The steplibs (in sequence) as specified in the Natural Security profile for the current library or in the steplib table.
3. The library SYSLIBS in the FNAT system file, which contains objects shared by system commands and utilities.
4. The library SYSTEM in the FNAT system file.
5. The library SYSTEM in the FUSER system file.

Since the FUSER system file is searched last, you must provide an object that is used in both the FUSER and the FNAT system files (for example, a user-exit routine for a Natural utility) only in one location, namely in FUSER.

Logging on to a Library

To log on to a library, select the **Library** menu and press ENTER. A window appears showing the command <LOGON> followed by a list of all existing libraries. For example:

```
+-----+
| <LOGON> |
| DEMO1   |
| DEMO2   |
| DEMO3   |
| SAMPLES |
| SYSEXP  |
| SYSEXS  |
| SYSTEM  |
+-----+
```

To create a new library, choose the **<LOGON>** command. A window is then displayed in which you enter the name of the library to be created. See also *Library Naming Conventions*.

To use an existing library, you can either choose the library from the list, or choose the **<LOGON>** command and then enter the name of the desired library. The contents of the library are then displayed. For example:

```
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| Cmd Name      Type      SM S/C User ID   SRC Date      GP Date      |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| <DIRECT COMMAND> |
|   ARRAYD      Program    S  S   SAGPC    07:45 02-02-07 |
|   ARRAYE      Program    S  S/C SAGPC    07:45 02-02-07 09:25 02-02-07 |
|   BREAK1      Program    S  S/C SAGPC    07:45 02-02-07 09:25 02-02-07 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
```

Field	Explanation
Cmd	<p>In this field, you can enter a function code. Enter "?" or press F2 to display the list of the available function codes for the selected object. The function codes are:</p> <p>C Check the object's source code for syntax errors.</p> <p>D Read the object's source code into work area.</p> <p>E Edit the object's source.</p> <p>L List the object's source code.</p> <p>I Display the directory information about the object.</p> <p>H Print hardcopy of the object's source.</p> <p>R Run (that is, compile and execute) the object's source.</p> <p>X Execute the object.</p> <p>S Stow the object in source and object form.</p> <p>U Scratch (that is, delete) the object's source and object form.</p> <p>. End.</p>
Name	Object name.
Type	Object type.
SM	<p>Programming mode:</p> <p>S Structured mode.</p> <p>R Reporting mode.</p>
S/C	<p>S The object exists in source form.</p> <p>C The object exists in object (cataloged) form.</p>
User ID	The ID of the user who saved/cataloged the object.
SRC Date	Source program date. The time and date when the object was last saved.
GP Date	Generated program date. The time and date when the object was last cataloged.

Note:

It is also possible to enter the system command LOGON *library-name* in the **Direct Command** window.