


[print](#) | [close](#)

APIs by Example: Library List Management

[System iNetwork Programming Tips Newsletter](#)

[Carsten Flensburg](#)

Carsten Flensburg

Thu, 07/12/2007 (All day)

Managing and controlling library lists is essential when setting up production and testing environments. Typically, the focus is on the user part of the library list because this is usually controlled by the job description assigned to a job. However, the product libraries as well as the current library can play a useful and practical role when you define the application runtime environment.

To stay true to the headline of this column, APIs are of course also involved in today's article. The Change Library List (QLICHGLL) API can change a job's product libraries, current library, and the user library list, and a number of other work management APIs are available to retrieve library list information for jobs, job descriptions, and system values. Putting them all into action is taken care of by three new library list CL commands.

Let me start with the library list basics. A library is the container of objects in the QSYS.LIB file system. All libraries are themselves placed in the QSYS library -- even the QSYS library is placed there. Whenever an object needs to be located, and the object name is unqualified, the library list is searched for an object of the specified name and type. The search starts in the library specified at the top of the library list and continues until a match is found or the end of the library list is reached. A library list is made up of up to four different types of library list entries:

1. the system library list -- a maximum of 15 libraries are allowed
2. the product library list -- a maximum of 2 libraries are allowed
3. the current library -- only one library can be the current library at any time
4. the user library list -- a maximum of 250 libraries are allowed (since V5R1; prior to that, 25 libraries were allowed); the user library list can still be limited to 25 libraries by means of data area QLMTUSRLIB (see IMB documentation below)

Here's an example of where and how all these libraries are located in a jobs library list, as displayed by the Display Library List (DSPLIBL) command:

Display Library List				
				System: WYNDHAMW
Type options, press Enter.				
5=Display objects in library				
ASP				
Opt	Library	Type	Device	Text
	QSYS	SYS		System Library
	QSYS2	SYS		System Library for CPI's
	QUSRSYS	SYS		System Library for Users
	QHLPSYS	SYS		

QGY	SYS	
QPDA	PRD	
COMPUTIL	PRD	Company utilities
COMPDTA	CUR	Company data library
QTEMP	USR	
COMPPGM	USR	Company program library
QMOM	USR	WEBSphere MQ FOR ISeries
QGPL	USR	General Purpose Library
More...		

Note the Type column in the above example, specifying which part of the library list a given library belongs to. The different parts of the library list also have a different origin:

1. The system library list is taken from the system value QSYSLIBL when a job is initiated.
2. The product libraries are temporarily part of the library list whenever the menus or commands for which they were specified as product libraries are active.
3. The current library is defined by the user profile that initiated the job. All user profiles have a CURLIB attribute, which can be either a named library or the special value *CRTDFT. In the latter case, library QGPL is used whenever an object is specified to be created in library *CURLIB. As you will see in a moment, a current library can also come from a menu or command.
4. The user library list comes from the job description that was assigned to the job. The job description, in turn, can point to the system value QUSRLIBL. The system default job description QDFTJOB, when shipped, points to the system value QUSRLIBL.

The following CL commands all have a Product library (PRDLIB) and Current library (CURLIB) attribute: The Create Menu (CRTMNU), Change Menu (CHGMNU), Create Command (CRTCMD), and Change Command (CHGCMD). If a product or current library is assigned by one of these to a menu or command, it will cause the specified library to be in effect as product or current library while the menu or command is active.

This facility is quite useful if, for example, you have created a CL command where the command object resides in a library already in the common library list, but depends on objects (say the command processing program) placed in a library that is not part of that library list. By specifying the special library as the CL commands product or current library, you ensure that the command will be able to locate dependent objects without qualifying the referenced objects. This usefulness also applies to menus that likewise might be dependent on the presence of referenced objects.

Note that if more menus or commands are active at a given point, the menu or command that was activated most recently takes precedence. As the menus or commands are ended, however, the menus or commands that you return to will have their product or current libraries reinstated. The product or current libraries are kept in an internal stack to allow the operating system to restore them upon return.

The following native CL commands are available to manipulate and display library lists for the job executing the command:

- Edit Library List (EDTLIBL) displays and changes the user library list.
- Change Library List (CHGLIBL) displays and changes the user library list and current library.
- Change Current Library (CHGCURLIB) changes the current library.
- Add Library List Entry (ADDLIBLE) adds a library to the user library list, optionally in a specific position.

- Remove Library List Entry (RMVLIBLE) removes a library from the user library list.
- Display Library List (DSPLIBL) displays the full library list.
- Change System Library (CHGSYSLIBL) adds and removes libraries to and from the system library list.
Note that changing the system library list could jeopardize system security and integrity.

As mentioned earlier, libraries and library lists are also part of menus, commands, user profiles, job descriptions, subsystem descriptions, and system values. Use the appropriate CL commands to maintain these object types and their library values.

To add a bit to the functionality provided by the native library list CL commands, and make it easier to test programs requiring a specific library list setup during development and deployment, I have created three new CL commands that let you save and restore a job's product libraries, current library, and user libraries, as well as replace the user library list based on a job description. Here are the command prompts and a brief description of each command:

```

                                Save Library List (SAVLIBL)
Type choices, press Enter.

User profile . . . . . *CURRENT      Name, *CURRENT

```

The Save Library List (SAVLIBL) command saves a job's product libraries, current library, and user libraries to the user profile specified. The library information is stored with the user profile using the User Application APIs described earlier and used in this column (see links to the previous articles below). Note that special authority *SECADM and *ALLOBJ is required if you specify a user profile that is different to the current job's user profile.

```

                                Restore Library List (RSTLIBL)
Type choices, press Enter.

User profile . . . . . *CURRENT      Name, *CURRENT

```

The Restore Library List (RSTLIBL) command does the exact opposite of the SAVLIBL command -- it restores the library information previously saved with the specified user profile and replaces the current job's product libraries, current library, and user library list with the saved libraries. Special authority *SECADM and *ALLOBJ is required if you specify a user profile that is different from the current job's user profile.

```

                                Set Library List (SETLIBL)
Type choices, press Enter.
Job description . . . . . *NONE      Name, *NONE
Library . . . . .                Name, *LIBL, *CURLIB
Product library 1 . . . . . *SAME    Name, *SAME, *NONE
Product library 2 . . . . . *SAME    Name, *SAME, *NONE
Current library . . . . . *SAME      Name, *SAME, *CRTDFT

```

The Set Library List (SETLIBL) command can set the current job's user library list as it replaces the current list with that of the specified job description. If the job description specifies *SYSVAL as its Initial library list (INLLIBL), the system value QUSRLIBL is retrieved and used as a replacement. If *NONE is specified as the job description's initial library list, the current job's user library list is removed. This could, of course, mean that some user objects and commands are no longer immediately available, so proceed with care.

This APIs by Example includes the following sources:

```

CBX175  -- RPGLE  -- Library List Management - Services
CBX175B -- SRVSRC -- Library List Management - Binder source

CBX1751 -- RPGLE  -- Save Library List - CPP
CBX1751H -- PNLGRP -- Save Library List - Help
CBX1751V -- RPGLE  -- Save Library List - VCP
CBX1751X -- CMD    -- Save Library List

CBX1752 -- RPGLE  -- Restore Library List - CPP
CBX1752H -- PNLGRP -- Restore Library List - Help
CBX1752V -- RPGLE  -- Restore Library List - VCP
CBX1752X -- CMD    -- Restore Library List

CBX1753 -- RPGLE  -- Set Library List - CPP
CBX1753H -- PNLGRP -- Set Library List - Help
CBX1753V -- RPGLE  -- Set Library List - VCP
CBX1753X -- CMD    -- Set Library List

CBX175M -- CLP     -- Library List Management - Build commands

```

To create all above objects, compile and run CBX175M. Compilation instructions are found in the source headers as usual.

User Application API examples previously published:

APIs by Example: User Application Information APIs, Part 1:

<http://www.systeminetwork.com/article.cfm?id=52288>

APIs by Example: User Application Information APIs, Part 2:

<http://www.systeminetwork.com/article.cfm?id=52387>

APIs by Example: User Application Information APIs, Part 3:

<http://www.systeminetwork.com/article.cfm?id=52457>

IBM Library List Related Documentation:

Library lists:

<http://publib.boulder.ibm.com/infocenter/iserics/v5r4/topic/rbam6/ulibl.htm>

Functions of using a library list:

<http://publib.boulder.ibm.com/infocenter/iserics/v5r4/topic/rbam6/objuseliblist.htm>

OS/400 Library List Search Support:

http://www-03.ibm.com/servers/enable/site/os400/lib_list.html

V5R2+ Data area support of 25 library limit in user library list (QLMTUSRLIB):

[https://www-](https://www-012.ibm.com/s_dir/slkbase.NSF/1ac66549a21402188625680b0002037e/cdb71315ede7ac5586256c1d005631db?OpenDocument)

[012.ibm.com/s_dir/slkbase.NSF/1ac66549a21402188625680b0002037e/cdb71315ede7ac5586256c1d005631db?OpenDocument](https://www-012.ibm.com/s_dir/slkbase.NSF/1ac66549a21402188625680b0002037e/cdb71315ede7ac5586256c1d005631db?OpenDocument)

This article demonstrates the following library list related APIs:

Change Library List (QLICHGLL) API:

<http://publib.boulder.ibm.com/infocenter/iserics/v5r3/topic/apis/qlichgll.htm>

Retrieve Job Information (QUSRJOBI) API:

<http://publib.boulder.ibm.com/infocenter/iserics/v5r3/topic/apis/qusrjobi.htm>

Retrieve Job Description Information (QWDRJOB) API:

<http://publib.boulder.ibm.com/infocenter/iserics/v5r3/topic/apis/qwdrjobd.htm>

Retrieve System Values (QWCRSVAL) API:

<http://publib.boulder.ibm.com/infocenter/iserics/v5r3/topic/apis/qwcrsval.htm>

You can retrieve the source code for this API example from:

http://www.pentontech.com/IBMContent/Documents/article/55065_246_LibApi.zip.

Source URL: <http://iprodeveloper.com/rpg-programming/apis-example-library-list-management>