

## APIs By Example: Where Is That Module Bound?

[Carsten Flensburg](#)

Fri, 04/26/2013 - 4:01pm

Retrieve module information with the aid of the WRKMODREF command



I've presented several commands in the APIs by Example column that aim to let you resolve cross-references in an ILE development environment. These commands include [Work with Service Program References](#) (WRKSPGREF), [Work with Program Imports](#) (WRKPGMIMP), and [Work with Program Exports](#) (WRKPGMEXP). Today, I'll add the Work with Module References (WRKMODREF) command to this collection. For instructions on how to create WRKMODREF, see the "How to Compile" section below.

### How WRKMODREF Works

Ideally, a module is bound into only one program or service program object. If you need a given module's functionality in more than one program, you'd place that module in a service program. This technique makes the module's function available to all programs that need it by letting these programs import the encapsulating subprocedure from the service program. For whatever reason, though, sometimes the ILE design guideline of keeping a module in one place isn't always followed. But sooner or later, you'll have to locate the programs or service programs containing a specific module because of a pending revision of the module, or ensure that all versions of the bound module are at the current level.

The WRKMODREF command employs two Program and CL Command APIs to retrieve program and module information: Retrieve Program Information (QCLRPGM) and Retrieve Module Information (QBNRMODI), respectively. The QCLRPGM API retrieves the program type to identify whether a program is an OPM- or ILE-type program; the QBNRMODI API returns module source, timestamp, and library attributes. Both APIs have the same interface and define a total of five required parameters (Figure 1).

Figure 1: The QCLRPGM API and QBNRMODI API required parameter group

1	Receiver variable	Output	Char (*)
2	Length of receiver variable	Input	Binary (4)
3	Format name	Input	Char (8)
4	Qualified object name	Input	Char (20)
5	Error Code	I/O	Char (*)

Default Public Authority: \*USE

The first parameter, *Receiver variable*, defines the program variable that will receive the requested program or module information. The second parameter, *Length of receiver variable*, specifies the length of the variable that the API can safely address. The *Format name* parameter defines the format of the returned data. The *Qualified object name* parameter indicates the name of the program or module for which the API should return information. The final parameter, *Error Code*, is the standard API error data structure.

### Building the *Work-with* List

To build the WRKMODREF command's program and service program *work-with* list, follow these steps:

1. Call the Open List of Objects (QGYOLOBJ) API to list all programs and service programs identified by the WRKPGMEXP command's REFPGM parameter.
2. Verify that each program in the list is an ILE program, and use the List ILE Program Information (QBNLPGM) API to display all modules bound into the program. For each identified service program, use the List Service Program Information (QBNLSPGM) API to show all modules bound into it.
3. Perform the following verification process for each bound module listed:

- a. Check whether the module name corresponds to the searched-for module name.
  - b. If it does, check that the searched module library name is either \*ALL or corresponds to the library name of the module that you verified in step 3a.
4. If step 3b produces a match, the WRKMODREF command’s work-with list will include the module information, and a check will verify whether the module’s source and timestamp attributes match the corresponding attributes of the module being evaluated. The work-with list contains a column showing the found *Module state* as either \*MATCH or \*DIFF—the latter value indicates that one or more of the verified attributes failed to produce a match.

You can see the WRKMODREF command’s prompt panel in Figure 2.

Figure 2: Work-with Module References (WRKMODREF) command prompt

```
Work-with Module References (WRKMODREF)

Type choices, press Enter.

Module . . . . . Name
Library . . . . . *LIBL Name, *LIBL, *CURLIB
Reference program . . . . . Name, generic*, *ALL
Library . . . . . *LIBL Name, *LIBL, *CURLIB...
Reference program type . . . . . *ANY *ANY, *PGM, *SRVPGM
Bound module library . . . . . *MODULE Name, *MODULE, *ANY
Sort order . . . . . *OBJLIB *OBJLIB, *TYPOBJ, *LIBOBJ...
Output . . . . . * *, *PRINT
```

Here, you specify the qualified name of the module to locate as the primary parameter and the range of programs or service programs to search for the module as the second parameter. The command’s third parameter lets you narrow the search to only programs or service programs, if applicable.

The fourth parameter identifies the name of the bound module library to evaluate against the module library name indicated in the program’s or service program’s bound module list. The special value \*MODULE says to use the library name specified or resolved for the module name that you designated as the primary parameter. Alternatively, the special value \*ANY means that any library name will qualify. You can also enter a specific library name of your own choosing to use in the module library name test.

WRKMODREF lets you specify a variety of sort orders for the produced list, as well as print the command’s output instead of displaying it in a *work-with* panel. The command’s online help text panel group provides more details about the command and its parameters. To show you an example of the list panel, I ran the following command on my system:

```
WRKMODREF MODULE (PRODLIB/WEB100)
REFPGM (PRODLIB/WEB*)
REFPGMTYP (*ANY)
BNDMODLIB (*MODULE)
ORDER (*OBJLIB)
OUTPUT (*)
```

Figure 3 displays the resulting *work-with* panel.

Figure 3: Work-with Module References list panel

```
Work-with Module References WYNDHAMW

22-02-13 20:03:10

Module . . . . : WEB100 Creator . . . : CARSTEN
Library . . . : PRODLIB Module created : 19-11-12 09:25:10
Attribute . . : RPGLE Source changed : 16-11-12 14:36:20
Text . . . . : Web File Service Functions

Type options, press Enter.
2=Update 4=Delete 5=Display 7=Rename 8=Program reference 9=PDM
```

Program			-----Source-----			Module	
Opt	Name	Library	Type	Member	File	Library	State
	WEB100	PRODLIB	*SRVPGM	WEB100	QRPGLESRC	WEBDEVLIB	*MATCH
	WEB155	PRODLIB	*PGM	WEB155	QRPGLESRC	WEBDEVLIB	*DIFF
	WEB170	PRODLIB	*PGM	WEB170	QRPGLESRC	WEBDEVLIB	*DIFF
	WEB172	PRODLIB	*PGM	WEB172	QRPGLESRC	WEBDEVLIB	*DIFF
	WEB200	PRODLIB	*PGM	WEB200	QRPGLESRC	WEBDEVLIB	*MATCH
	WEB210	PRODLIB	*PGM	WEB210	QRPGLESRC	WEBDEVLIB	*MATCH
							More...
Parameters or command							
==>							
F3=Exit	F4=Prompt	F5=Refresh	F8=Display module	F9=Retrieve			
F11=Display date and time	F12=Cancel	F17=Top	F18=Bottom				

Here, the F11 key lets you toggle between the various types of program and module information exposed by the list panel, including the initial columns, member create and source change date and time, and module text.

Using the available list options, you can run a number of program and service program commands against selected programs or service programs. The list of commands includes UPDPGM/UPDSRVPGM, DLTPGM/DLTSRVPGM, DSPPGM/DSPSRVPGM, DSPMOD, WRKMBRPDM, and DSPPGMREF. The online, cursor-sensitive help text further details the list panel sections, columns, options, and function keys.

For more information about the APIs and concepts presented here, see the references listed in the “Find Out More” section below. If you have a topic that you’d like me to cover in future installments of APIs by Example, please let me know at [flensburg@novasol.dk](mailto:flensburg@novasol.dk).

## How to Compile

Below, you’ll find instructions detailing how to create the Work with Module References command. The following sources are included with the [code download](#) associated with this article:

CBX260—RPGLE: Work with Module References—CPP

CBX260E—RPGLE: Work with Module References—UIM Exit Program

CBX260H—PNLGRP: Work with Module References—Help

CBX260P—PNLGRP: Work with Module References—Panel Group

CBX260V—RPGLE: Work with Module References—VCP

CBX260X—CMD: Work with Module References

CBX260M—CLP: Work with Module References—Build Command

To create all of the command objects, compile and run the CBX260M CL program, following the instructions in the source header. You’ll also find compilation instructions in the respective source headers of the individual sources.

## Find Out More

[“ILE Concepts for the Impatient Programmer”](#)

**IBM i 7.1 Information Center documentation**

[List ILE Program Information \(QBNLPGMI\) API](#)

[List Service Program Information \(QBNLSPGM\) API](#)

[Retrieve Module Information \(QBNRMODI\) API](#)

[Retrieve Program Information \(QCLRPGMI\) API](#)

**iPro Developer articles**

“[Analyzing ILE Programs](#)” (December 2002)

“[APIs by Example: Identifying and Working with Service Program References](#)” (September 2008)

“[APIs by Example: Keeping Track of Your Exports](#)” (February 2013)

“[APIs by Example: Locating and Working with Module Imports](#)” (October 2008)

**Source URL:** <http://iprodeveloper.com/application-development/apis-example-where-module-bound-o>