

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

APIs by Example: Finding IBM i Message Descriptions

[System iNetwork Programming Tips Newsletter](#)

[Carsten Flensburg](#)

Carsten Flensburg

Thu, 04/23/2009 (All day)

Last time, I discussed and demonstrated some of the uses for which IBM i message descriptions provide a flexible and versatile foundation (["Copying System i Message Descriptions,"](#) March 26, 2009, article ID 57897 at SystemiNetwork.com). Today, I continue the coverage of IBM i message descriptions and focus on some of the purposes the operating system and its components use message descriptions for. I hope the information inspires you and demonstrates the usefulness that message descriptions offer an application programmer.

As usual, I also provide a utility involving the use of one or more APIs related to the topic at hand. Because today's article investigates IBM i's employment of message descriptions, I've created a CL command that helps you find message descriptions based on generic message IDs as well as a text string in either the message description's message text or second-level text. I've named the command Find Message Description (FNDMSGD).

As for the comprehensive operating system use of message files and message descriptions, I came across the following excerpt in [IBM's Info Center for release 5.2](#) while researching today's topic:

Following is a representative sample of the message identifier prefixes that are used to identify messages in several of the IBM-supplied message files. Shown are some of the prefixes for the OS/400 operating system, ILE COBOL/400*, ILE RPG/400*, the IDU and SDA utilities, and keyboard and machine interface (MI) messages.

```
CPA -- OS/400 system operator action
CPC -- OS/400 system completion messages
CPD -- OS/400 system diagnostic messages
CPI -- OS/400 system informational messages
CPX -- OS/400 system titles and texts
CPZ -- OS/400 system abnormal termination
CBE -- COBOL run time
CBL -- COBOL compiler
CBX -- COBOL titles and texts
CSC -- COBOL syntax checker
QRG -- RPG language compiler
RPG -- RPG run time
RPT -- RPG auto report
RSC -- RPG syntax checker
RTX -- RPG auto report titles and texts
RXT -- RPG relational diagnostic texts
IDU -- Interactive database utilities (IDU)
IDX -- IDU titles and text
SDA -- Screen design aid (SDA)
SDX -- Screen design aid titles and texts
```

```
KBD -- Keyboard
MCH -- iSeries 400 machine instruction interface
```

All the message descriptions beginning with CP as well as MCH are found in the system QCPFMSG message file. The CPF abbreviation is a legacy from the System/38 operating system, which was called Control Program Facility (CPF). And on a second note, apart from the FNDMSGD command accompanying today's article, I recently wrote a Find Message File (FNDMSGF) command for my own use, and it enables me to locate all message files containing a specific message ID. I doubt that this command has a broader appeal, so I chose not to publish it, but in case you'd find it useful, just let me know and I'll send you a copy.

Anyway, I've collected a number of the QCPFMSG messages containing category CPX (system titles and texts) message descriptions to demonstrate how message files and message descriptions can be used to solve the programming challenge of storing and retrieving pieces of information and texts to be used in your applications and programs. One of these examples is familiar to everyone who has ever evoked the command entry display by, for example, pressing function key F10 from a PDM display—or directly calling the system request processing program QCMD—as it relates to the command entry panel title. I use my new FNDMSGD command to locate the panel title and, assuming that it is stored in the QCPFMSG file, I prompt the FNDMSGD command to enter the search criteria—and note the following criteria supported by the command:

Find Message Description (FNDMSGD)			
Type choices, press Enter.			
Message file		Name	
Library	*LIBL	Name, *LIBL,	
*CURLIB			
Message identifier	*ALL	Name, generic*,	
*ALL			
Include text	*ALL		
Search text	*MSG	*MSG, *SECLVL,	
*BOTH			

The *Message file* parameter designates the message file in which to look for the message descriptions that meet the selection criteria defined by the subsequent three parameters: a generic *Message ID* specifying the subset of messages to include in the search or the special value *ALL if all messages should be included; an *Include text* parameter allowing you to specify a text string to be used in a scan operation performed against the text elements of the message description; and finally a *Search text* parameter telling the FNDMSGD command whether to search for the text string specified in the first-level message text, the second-level message text, or both.

To continue my investigation, I run the command below to find out which CPX-messages contain the words "Command Entry":

```

FNDMSGD MSGF(QCPFMSG)
        MSGID(CPX*)
        INCLUDE('command entry')
        SEARCH(*BOTH)

```

Note that the INCLUDE text parameter is not case sensitive; all strings are converted to uppercase by the CPP before comparing the strings searched. Running the aforementioned command eventually leads to the display of the following panel:

Find message description				
WYNDHAMW		18-04-09		
14:36:27				
Message file . . . :	QCPFMSG	Include string . .		
command entry				
Library :	QSYS	Search text . . .	*BOTH	
Message ID	CPX*	Position to . . .		
Type options, press Enter.				
2=Change 3=Copy 4=Delete 5=Display 8=Work with message description				
Opt	Message ID	Message text		
entry	CPXB9C1	F13=Clear F17=Top F18=Bottom F21=CL command		
	CPX1EB0			
	CPX2423			
Bottom				
Parameters or command				
===>				
F3=Exit	F4=Prompt	F5=Refresh	F6=Add message description	
F9=Retrieve				
F11=Second level		F12=Cancel	F21=Print list	

The above panel allows you to alter all available search criteria and continue or refine your search if necessary. Although it is not immediately clear why the second and third message IDs are included in the above list, pressing the list panel's F11 function key reveals that the text string "Command Entry" is found in message ID CPX2423's second-level help text. How can I now find out whether this is the actual text used by the operating system to display the command entry panel's panel title? And how can I do that without altering the QSYS version of the QCPFMSG file, which I sincerely recommend that you **do not** change in any way. I opt for running the following series of commands to verify that I've indeed located the right message ID:

```
1. CRTMSGF MSGF(QTEMP/QCPFMSG)

2. CPYMSGD MSGID(CPX2423)
   MSGF(QSYS/QCPFMSG)
   TOMSGF(QTEMP/QCPFMSG)

3. When prompted by the CPYMSGD command, for the second-level text,
   overwrite Command Entry
   with your name, preserving the left apostrophe as well as the
   position of the words
   following Command Entry as in the following example:

   Second-level message text . . . 'Carsten Flensburg

   System/36 Command Entry
System/38 Com
mand Entry'

4. OVRMSGF MSGF(QCPFMSG)
   TOMSGF(QTEMP/QCPFMSG)
   SECURE(*YES)

5. CALL QCMD
```

If my assumption is correct and all above commands complete normally, you'll now see your name heading the command entry display. To recover the panel's normal appearance, delete the message file, the message description, or the message file override. Note that the Override Message File (OVRMSGF) command is different from other override commands in the sense that if a specified message ID is not found in the message file being overridden to, the overridden message file will instead be searched. This behavior is quite useful because you're not forced to copy all message IDs from one message file to another in order to change only one or a few message IDs. In the above example, all other QCPFMSG message IDs than CPX2423 will be retrieved from the QSYS version of the message file. At the end of this article, I've included a link to an article written by Gary Guthrie explaining the general concept of file overrides in more detail.

Another example of message description usage is the well-known approach of storing CL commands for a menu, in this case the system attention menu, in a message description. Using the FNDMSGD command against CPX*-message IDs and for example the DSCJOB (Disconnect Job) command quickly locates the CPX2313 message ID:



Select Message Details to Display							
							System:
WYNDHAMW							
Message ID		:	CPX2313				
Message file		:	QCPFMSG				
Library		:	QSYS				
Message text		:	ENDRQS	DSPJOB	DSPMSG		
SNDMSG		SI					
GNOFF	DSPMSG	DSCJOB	DSPWSUSR	ENDRDBRQS			
Select one of the following:							
1. Display message text							
5. Display message attributes							
30. All of the above							
Selection							
F3=Exit F12=Cancel							

Following the OVRMSGF approach described earlier, you could easily replace the DSPJOB command with the WRKJOB command and this way gain access to both change job functions as well as a command line by evoking the attention menu. Placing the OVRMSGF in your user profile's initial program will make your version of the CPX2313 message ID immediately available in all your interactive jobs.

As yet another example of how message descriptions can be utilized, you'll notice that when displaying a journal entry by running the Display Journal (DSPJRN) command followed by option 5 for a journal entry, the journal entry code and type is followed by the textual description of the actual type and code:

```

                                Display Journal Entry

Object . . . . . :                               Library . . . . . :

Member . . . . . :

Incomplete data . . :   No                      Minimized entry data :
*NONE
Sequence . . . . . :   572218304

Code . . . . . :   T  - Audit trail entry

Type . . . . . :   JS - Job data

```

Again the FNDMSGD command will offer a number of possible candidates for the storage location of the descriptive texts. My immediate guess points to the CPX7064 message ID for the journal code texts and for release 5.4 the message ID interval CPX7065 to CPX7094 for the journal type texts. The appearance of the message ID's second-level text reveals that a certain level of formatting has been applied to the texts. I've included a sample program (CBX202T) with this article to give you an example of how the information in question can be retrieved. If you simply compile and run the CBX202T program, the retrieved information will be visible in both the message subfile of your currently displayed panel as well as in the job log of your current job.

I've created a few utilities around the audit journal on my system and find it convenient to use this system-supplied journal entry information as opposed to defining and storing this information elsewhere myself. Note though that system information stored in message descriptions does not constitute a formal interface, so release upgrades should always prompt a verification effort on your part to be able to rely on this information beyond the point of upgrade.

The mentioned examples are by no means exhaustive as far as the many ways to exploit message descriptions in your daily programming tasks is concerned, but hopefully they add an angle or two to your perception of the functionality and techniques supported by message files and message descriptions. I suggest you use the FNDMSGD command to look up further examples of system information and facilities hidden in the message files on your IBM i system.

This APIs by Example includes the following sources:

```

CBX202T  -- RPGLE  -- Retrieve Journal Entry Codes and Types - sample
program

CBX202   -- RPGLE  -- Find Message Description - CPP
CBX202E  -- RPGLE  -- Find Message Description - UIM General Exit Pgm

```

```
CBX202H  -- PNLGRP  -- Find Message Description - Help
CBX202L  -- RPGLE   -- Find Message Description - UIM List Exit Program
CBX202P  -- PNLGRP  -- Find Message Description - Panel Group
CBX202V  -- RPGLE   -- Find Message Description - VCP
CBX202M  -- CLP     -- Find Message Description - Build command
```

To create all these FNDMSGD command objects, compile and run CBX202M, following the instructions in the source header. As always, you'll also find compilation instructions in the respective source headers. If you're interested in a copy of the Find Message File (FNDMSGF) command, please send me an e-mail at flensburg@novasol.dk

IBM documentation:

[CL message concept](#)

[Message ID overview](#)

[System/38 Wiki](#)

Previously published related articles:

[APIs by Example: Copying IBM i Message Descriptions](#)

[So You Think You Understand File Overrides](#)

This article demonstrates the following Message Handling API:

[Retrieve Message \(QMHRTVM\) API](#)

[You can retrieve the source code for this API example here.](#)

Source URL: <http://iprodeveloper.com/rpg-programming/apis-example-finding-ibm-i-message-descriptions>