


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

## APIs by Example: Data Queue APIs and CL Commands, Part 4

[System iNetwork Programming Tips Newsletter](#)

[Carsten Flensburg](#)

Carsten Flensburg

Thu, 02/08/2007 (All day)

This week, I'm joining the threads from all three previous installments of this article series in a new, alternate Work with Data Queues (WRKDTAQ2) command. The WRKDTAQ2 command offers a Work-with interface to the new data queue commands that I've published so far:

The Send Data Queue Entry (SNDDTAQE), Copy Data Queue Description (CPYDTAQD), Clear Data Queue (CLRDTAQ), Display Data Queue Description (DSPDTAQD), and Display Data Queue Entries (DSPDTAQE) commands are all available from the Work with Data Queues panel, as are the native commands Create Data Queue (CRTDTAQ), Delete Data Queue (DLTDTAQ) and Change Object Description (CHGOBJD).

Today, I also briefly discuss some of the data queue interfaces that various system functions use to communicate with user-written applications. These add some interesting options to help programmers solve technical challenges difficult to overcome otherwise. I also include links to IBM documentation as well as specific code examples.

The WRKDTAQ2 command's purpose is to offer an interface that makes it easier and faster to work with data queues during development of data queue-driven applications. The command also helps overcome data queue administration difficulties. The WRKDTAQ2 command lets you perform all available operations against data queues from one panel and keeps all data queue CL commands in one spot.

The WRKDTAQ2 command also provides parameters that control the type of data queues to include in the list, as well as the presentation order of the list.

Here's how the WRKDTAQ2 command prompt looks:

Work with Data Queues (WRKDTAQ2)

Type choices, press Enter.

Data queue . . . . .		Name, generic*,
*ALL		
Library . . . . .	*LIBL	Name, *LIBL,
*CURLIB...		
Data queue type . . . . .	*ALL	*ALL, *STD, *DDM
Sort order . . . . .	*DTAQ	*DTAQ, *LIB

When run, the WRKDTAQ2 command displays a panel similar to this one:

```

                                Work with Data Queues

WYNDHAMW
                                02-02-07

20:14:04
Type options, press Enter.

    1=Send entry          3=Copy      4=Delete          5=Display description
    8=Display entries    13=Change description  14=Clear

Opt  Data queue  Library      Type      Text
----
  QSYSDTAQ      QSYS        *STD      Exit point: QIBM_QWT_JOBNOTIFY
  RMTAPP01      QGPL        *DDM      Remote RDB connection - get

  RMTAPP02      QGPL        *DDM      Remote RDB connection - put

  TEST01        QGPL        *STD      Test FIFO data queue

  TEST02        QGPL        *STD      Test keyed data queue

  WEBINB01      QGPL        *STD      Web application inbound

  WEBOUT01      QGPL        *STD      Web application outbound

Bottom
Parameters or command

===>

  F3=Exit    F4=Prompt    F5=Refresh    F6=Create data queue
F9=Retrieve
  F11=Display full text  F12=Cancel    F17=Top      F18=Bottom

```

As usual, both the command and display panel are explained in more detail in the cursor-sensitive online help text included with the WRKDTAQ2 command.

Besides the efficient and flexible program-to-program communication support that data queues provide, they are also useful for enabling system-to-program communication. I've collected information and programming examples at the end of this article, documenting the following operating system usage of data queues:

1. Display File (DSPF): Specifying a data queue for the CRTDSPF, CHGDSPF, or OVRDSPF commands' DTAQ parameter lets your interactive application be event driven (i.e., wait for user input and data queue input at the same time), as well as lets it time out the display file if no action occurs within a user-defined time limit. For further documentation and a programming example, please follow the link provided at the end of this article.

2. Intersystem Communications Function File (ICFF): Specifying a data queue for the CRTICFF, CHGICFF, or OVRICFF commands' DTAQ parameter lets your communication application be event driven (i.e., wait for local program input and remote system input to the communications file at the same time), as well as lets it time out the communication if no action occurs within a user-defined time limit. A link to documentation is at the end of this article.
3. Spooled file support 1: Specifying a data queue for the CRTOUTQ or CHGOUTQ commands' DTAQ parameter generates a predefined data queue entry to the specified data queue every time a spooled file enters the ready (RDY) status on that output queue. Links to documentation and a programming example are at the end of this article.
4. Spooled file support 2: Adding the QIBM\_NOTIFY\_CRTSPLF environment variable at either system or job level generates a predefined data queue entry to the data queue specified in the environment variable, whenever a spooled file is created on the system or in the job for which the environment variable was added. Links to documentation and a programming example are at the end of this article.
5. Job event notification: Registering a data queue for the QIBM\_QWT\_JOBNOTIFY exit point and specifying a combination of event types and subsystems generates a predefined data queue entry whenever a specified job event (e.g., job placed on job queue, job starts, job ends) takes place in the specified subsystem. The programming example at the link provided at the end of this article reveals the details of setting up a job monitor using this exit point, and a link to the exit point documentation is also provided.

So far, this article series has brought you a set of new data queue CL commands and an overview of the many ways to exploit data queues in application design and development. In the next and final installment, I present a sample utility, demonstrating a client/server scenario using data queues as the communication layer. This also includes a discussion of the data protocol foundation necessary to provide the crucial robustness and flexibility in such an architecture.

### **This APIs by Example includes the following sources:**

```
CBX168  -- Work with Data Queues - CCP
CBX168E -- Work with Data Queues - UIM Exit Program
CBX168H -- Work with Data Queues - Help
CBX168P -- Work with Data Queues - Panel Group
CBX168V -- Work with Data Queues - VCP
CBX168X -- Work with Data Queues
CBX168M -- Work with Data Queues - Build command
```

To create all these objects, compile and run CBX168M. Compilation instructions are in the source headers, as usual.

### **IBM documentation for data queue support:**

Use Data Queues:

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

Examples: Using Data Queues or User Queues:

<http://publib.boulder.ibm.com/infocenter/iserics/v5r4/topic/apiref/ExTaskDataque.htm>

ICFF & Display File — Example Using a Data Queue:

<http://publib.boulder.ibm.com/infocenter/iseriess/v5r3/topic/apis/apiexusque.htm>

Data Queue Support for Spooled Files:

<http://publib.boulder.ibm.com/infocenter/iseriess/v5r3/topic/rzalu/rzaludataq.htm>

Job Notification Exit Point:

<http://publib.boulder.ibm.com/infocenter/iseriess/v5r3/topic/apis/xjobntfy.htm>

### **Programming examples of data queue usage (ProVIP membership required):**

Mystifying Timeout Messages — Tech Corner, July 1996:

<http://www.systeminetwork.com/article.cfm?id=1539>

INVITE a Timeout — Tech Corner, May 1996:

<http://www.systeminetwork.com/article.cfm?id=1592>

Automatic Spool Rerouting — March 31, 2004:

<http://www.systeminetwork.com/article.cfm?id=18333>

Automatic Notification When Spool Files Created — March 17, 2004:

<http://www.systeminetwork.com/article.cfm?id=18244>

Realtime Job Monitor — August 19, 2004:

<http://www.systeminetwork.com/article.cfm?id=19252>

Synchronizing Passwords Between Systems — July 21, 2004:

<http://www.systeminetwork.com/article.cfm?id=18986>

### **The previous installments of this article series:**

Data Queue APIs and CL Commands, Part 1

<http://www.systeminetwork.com/article.cfm?id=53542>

Data Queue APIs and CL Commands, Part 2

<http://www.systeminetwork.com/article.cfm?id=53685>

Data Queue APIs and CL Commands, Part 3

<http://www.systeminetwork.com/article.cfm?id=53850>

### **This article series demonstrates the following data queue APIs:**

Retrieve Data Queue Description (QMHQRDQD) API:

<http://publib.boulder.ibm.com/infocenter/iseriess/v5r3/topic/apis/qmhqrdqd.htm>

Send Date Queue (QSNDDTAQ) API:

<http://publib.boulder.ibm.com/infocenter/iseriess/v5r3/topic/apis/qsnddtaq.htm>

Receive Date Queue (QRCVDTAQ) API:

<http://publib.boulder.ibm.com/infocenter/iseriess/v5r3/topic/apis/qrcvdtaq.htm>

Clear Data Queue (QCLRDTAQ) API:

<http://publib.boulder.ibm.com/infocenter/iseriess/v5r3/topic/apis/qclrdtaq.htm>

Retrieve Data Queue Message (QMHRDQM) API:

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

All data queue APIs are documented here:

<http://publib.boulder.ibm.com/infocenter/iserics/v5r3/topic/apis/obj2.htm>

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

[http://www.pentontech.com/IBMContent/Documents/article/54001\\_161\\_DataQueue4.zip](http://www.pentontech.com/IBMContent/Documents/article/54001_161_DataQueue4.zip).

**Source URL:** <http://iprodeveloper.com/rpg-programming/apis-example-data-queue-apis-and-cl-commands-part-4>