

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

## APIs by Example: Realtime Job Monitor

[System iNetwork Programming Tips Newsletter](#)

[Carsten Flensburg](#)

Carsten Flensburg

Thu, 08/19/2004 (All day)

**Q: I need to develop a monitoring system that should be able to send an alert to let people know the following:**

- a) When a pre-defined job (job name only or job name and user name) enters a subsystem
- b) If the job is held by a user or is stuck at MSGW status
- c) How the job ended, either normally or abnormally

I looked at the Job Notification exit program, but it does not help us monitor a job's status changes. Other Work Management APIs ask you to fetch all active jobs at a given point of time, but don't tell if the job has ended.

Is there an API or other workaround available to address this problem?

**A:** The answer to the problem is to use the Job Notification exit program in conjunction with the Retrieve Job Information (QUSRJOBI) API.

For those not familiar with exit programs, an exit program is a program that gets called by the operating system when a particular event occurs. This event is called an "exit point," so the program that is called at that point is known as an "exit program."

For example, if you want to call a program whenever a device description is initialized by the telnet server, you could register a "Telnet Device Initialization" exit program. This program can then have some control over how the device gets initialized.

In the case of monitoring a job entering or exiting the system, the exit point that we'll use is called QIBM\_QWT\_JOBNOTIFY and is described as the "Job Notification" exit point.

The Job Notification exit point is unusual in that it does not call a program like the other exit points do. Instead, it writes a message to a data queue. You can read that data queue from a program to get information about a new job entering the system or an existing job ending.

Your other requirement was to know when a job gets held or gets stuck at message waiting ("MSGW") status. Since you can get the job's identification from the Job Notification data queue, you can then call the QUSRJOBI API to retrieve its status. You'll want to check at regular intervals until the job ends.

For this week's APIs by Example newsletter, Carsten Flensburg has written a utility that does that. It follows these steps:

- a) It waits on the Job Notification data queue for an entry.
- b) It checks a physical file to see if this is one of the jobs that you wanted to monitor. If it's not, it skips this entry and goes back to waiting on the data queue.
- c) If the entry is a "job start" entry, it adds the entry to the list of those that it should monitor and sends an \*INFO message to a message queue.
- d) If the entry is a "job end" entry, it removes the entry from the list of those that it should monitor and sends an \*INFO message to a message queue.

e) If a "time out" occurs while waiting for the entry on the Job Notification data queue, it will check each entry in its list to see what its status is. If the status is "HLD" or "MSGW," it sends an \*INFO message to a message queue. This time-out will occur every 25 seconds.

This program is intended to be submitted to be run as a "never-ending program." If you do want to end it, it checks every 25 seconds to see if a controlled end job request has been issued for it. If it has, it will end gracefully.

You can read about the QIBM\_QWT\_JOBNOTIFY exit point at the following ZZZlink:

<http://publib.boulder.ibm.com/iserics/v5r2/ic2924/info/apis/xjobntfy.htm>

Information about the timestamp when the job entered the system, started, and exited the system was added to this exit point in response to APAR SE06696. You can read this APAR at the following ZZZlink: [http://www-912.ibm.com/n\\_dir/nas4apar.nsf/c79815e083182fec862564c00079d117/afoace822025418786256be7003d5883?OpenDocument](http://www-912.ibm.com/n_dir/nas4apar.nsf/c79815e083182fec862564c00079d117/afoace822025418786256be7003d5883?OpenDocument)

The Realtime Job Monitor utility demonstrates the following APIs:

Clear Data Queue (QCLRDTAQ) <http://publib.boulder.ibm.com/iserics/v5r2/ic2924/info/apis/qclrdaq.htm>

Receive Data Queue (QRCVDTAQ) <http://publib.boulder.ibm.com/iserics/v5r2/ic2924/info/apis/qrcvdaq.htm>

Send Data Queue (QSNDDTAQ) <http://publib.boulder.ibm.com/iserics/v5r2/ic2924/info/apis/qsnddaq.htm>

Retrieve Data Queue Message (QMHRDQM)  
<http://publib.boulder.ibm.com/iserics/v5r2/ic2924/info/apis/qmhrdqm.htm>

Convert Date and Time Format (QWCCVTDt)  
<http://publib.boulder.ibm.com/iserics/v5r2/ic2924/info/apis/qwccvtdt.htm>

Send Nonprogram Message (QMHSNDM)  
<http://publib.boulder.ibm.com/iserics/v5r2/ic2924/info/apis/QMHSNDM.HTM>

You can download the source code for this article from  
<http://www2.systeminetwork.com/noderesources/code/clubtechcode/RealTimeJobMonitor.zip>.

The above source code was written by Carsten Flensburg. If you have any questions, you can contact Carsten at <mailto:flensburg@novasol.dk>.

**Source URL:** <http://iprodeveloper.com/rpg-programming/apis-example-realtime-job-monitor>