TechTip: WebSphere MQ Administration with PCF

GIUSEPPE COSTAGLIOLA / 01 FEBRUARY 2007

An MQ administrator not only is responsible for the administration and maintenance of the MQ objects (queue managers, queues, channels, listeners, authorizations, etc.), but also should provide some tools that ensure all MQ objects are operational, generate alerts in case of failure, give statistics with number of messages sent or received, and more.

By monitoring availability and collecting critical IBM WebSphere attributes, the MQ administrator can quickly identify performance problems and easily alert the appropriate personnel to fix the problem.

The WebSphere MQ Programmable Command Formats (PCF) is a set of commands that allow administration tasks to be issued directly from a user program. Instead of using APIs, PCF uses a special format to define a set of messages that are sent to the queue manager's command queue, where they are processed by the command server. Replies are returned to the designated reply-to queue.

When your program sends messages in the PCF format, you can query or update MQ resources, such as queues, channels, queue managers, etc. However, PCF commands and their replies are not in a text format that you can read; instead, each PCF command is a data structure that is embedded in the application data part of a WebSphere MQ message. Each command is sent to the target queue manager using the Message Queue Interface (MQI) function MQPUT in the same way as any other message. The command server on the queue manager receiving the message interprets it as a command message and runs the command. To get the replies, the application issues an MQGET call, and the reply data is returned in another data structure. The application can then process the reply and act accordingly.

These are some of the things your program must supply to create a PCF command message:

- Message Descriptor This is a standard WebSphere MQ message descriptor, in which message type (MsqType) is MQMT_REQUEST and message format (Format) is MQFMT_ADMIN.
- Application Data This contains the PCF message, including the PCF header, in which the PCF message type (Type) specifies MQCFT_COMMAND and the command identifier specifies the command for example, Inquire Channel Status (MQCMD_INQUIRE_CHANNEL_STATUS) or Inquire Queue (MQCMD_INQUIRE_Q).

For a complete description of the PCF data structures and how to implement them, see **WebSphere MQ Programmable Command Formats and Administration Interface**.

PCF may seem difficult at first glance, but when you understand how to set up the request command and how to get the response, you can do almost everything with your MQ.

Just remember that, before using the examples that are part of this article, you must do the following:

- Create a BndDir that includes LIBMQM in library QMQM.
- Create a queue for response SYSTEM.ADMIN.REPLYQ.JYMON.
- Unless the user is a member of the QMQMADM group or has *ALLOBJ authority, grant the user who runs the program access to queue manager *MQM, queue SYSTEM.ADMIN.COMMAND.QUEUE, queue SYSTEM.ADMIN.REPLYQ.JYMON, and any other MQ object you want to query. For more information about the authority to work with WebSphere MQ objects, see the **WebSphere MQ for iSeries V6 System Administration Guide**.
- Start MQ Command Server (**STRMQMCSVR**).

The enclosed RPGLE program **JYMONMQ** is an example on how you can monitor queue manager, channels, and queue using PCF. This program accepts two input parameters and returns three output parameters. The following table gives an example of input and output parameters:

Sample Input and Output Parameters	
Monitor Q Manager MQOBJNAME = ' ' MQOBJTYPE = '*MQM'	If Active... MQSTATUS = 'ACTIVE ' MQINFO = 'MQ manager for development system' MQERROR = *OFF
Monitor Channel MQOBJNAME = 'ABC.TO.XYZ' MQOBJTYPE = '*CHL'	If Inactive... MQSTATUS = 'INACTIVE ' MQINFO = 'MQRCCF_CHL_STATUS_NOT_FOUND' MQERROR = *ON If Active... MQSTATUS = 'RUNNING ' MQINFO = 'Msg(s):0 SENDER 128.127.xxx.xxx(1404)' MQERROR = *OFF
Monitor Queue MQOBJNAME = 'XML_INPUT' MQOBJTYPE = '*Q'	MQSTATUS = 'LOCAL ' MQINFO = 'Cur:58 Inp:0 Out:0 Local queue to accept XML messages'

Another example is the **MQPRTEV** command that accompanies this article; you can use it to print the events from an internal queue.

Create a physical file named JYMONMQC from the DDS enclosed and populate it with JYMONMQC.dat. Then run this command:

```
MQPRTEVQNAME(SYSTEM.ADMIN.CHANNEL.EVENT)CLEARQ(*NO)
```

This command will produce a report like the one shown in Figure 1:

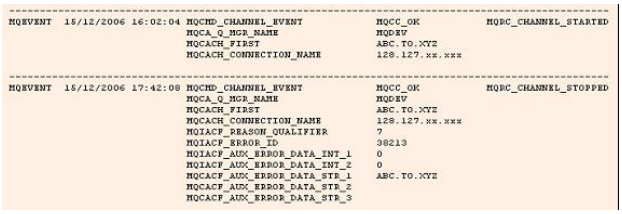

Figure 1: Your report will look like this. (Click image to enlarge.)

These two examples show how to use PCF to query the most common MQ objects, but as I already said, there is lot more in PCF. For example, you can write a program to issue PCF commands to any queue manager in the network from a single node. In this way, you can both centralize and automate your administration tasks such as creating queues, processing definitions and channels, or changing attributes of these objects directly from your program. Or you can simply enhance my JYMONMQ channel monitor to gather channel performance information by collecting the number of messages sent or received, the number of completed batches, the number of batches sent or received, the number of bytes sent or received, and other data.

Giuseppe Costagliola is a programmer in Turin, Italy. You can reach him at **beppecosta@yahoo.it**.

Giuseppe Costagliola is a programmer in Turin, Italy. You can reach him at**giuseppe.costagliola@gmail.com.**