

Organisation

Alphacourt Ltd. © 2000.

Author

Ruud van Zundert, on behalf of Alphacourt.

Purpose

This SupportPac gives a working example of how the current functionality of the MQSeries clustering workload exit can be extended.

This particular Cluster workload exit only writes messages to the queue manager with the highest network priority on its cluster receiver channel. This can be a useful way of including/excluding queue managers in a cluster.

If there are several queue managers with the same priority, then workload is distributed amongst them in a round-robin fashion.

If there is no appropriate target destination, it sets return code 2188 (rejected by cluster exit).

Background

MQSeries clustering using software is an excellent way of reducing system administration, improving availability and distributing workload.

The workload exit that comes supplied with the base product interrogates the status of the queue managers in the cluster, their channel status, and the status of the cluster queues.

The following applies when using this exit:

- If the cluster queues are defined with the 'default bind' (DEFBIND) set to 'NOT FIXED', then the workload exit distributes the messages in a round-robin fashion.
- If DEFBIND is set to 'ON OPEN', then the exit chooses one of the available queue managers, and all messages are routed to it.

If there is a local copy of the cluster queue, then no workload distribution takes place, and messages are written to the local queue.

If one of the available queue managers fails, then the exit ensures that messages are only sent to 'active' queue managers.

Writing your own exit

If you want the exit to do something different, then you'll need to write your own. Unfortunately, IBM do not give you the source of the default exit.

On the MQ Windows server platform, a 'C' programming sample can be found in directory \MQSeries\tools\c\samples called amqswlm0.c.

This sample is NOT the same as the default exit, but does provide you with an insight into the workings of the exit.

A good source of reference are the following IBM manuals which can be found on the web : <http://www-4.ibm.com/software/ts/mqseries/library/manualsa/>

manuals	MQSeries Queue Manager Clusters	csqzah03.pdf
	MQSeries Intercommunication	csqzae05.pdf

Pre-requisites/Dependencies

This source supplied with this SupportPac is based on the sample obtained at MQSeries for Windows NT/2000 V5.1 CSD6. At this service level, IBM introduced a new method of browsing the data in the cluster cache via module MQXCLWLN; details can be found within the memo.ptf member after CSD6 has been applied.

In fact, future releases will only work by calling this module, so if you have a current exit that browses the cluster cache by chasing pointers, now is the time to change it.

In order to keep track of the last chosen destination, the exit uses the first byte of the 'ExitUserArea' to store it. Please ensure that the cluster workload exit length is at least 1 (in fact, the default is 100).

You must alter the sending queue manager by issuing

```
ALTER QMGR CLWLEXIT('path\<exitname>(clwlFunction)')
```

and then restart the queue manager.

Note that the suffix '.dll' has been left off <exitname>. Also note the case of the data in brackets.

Environments tested

Operating System	Windows 2000, service pack 1 & service pack 2
MQSeries	Windows V5.1 CSD6 & Windows V5.2 CSD1
Compiler	Microsoft Visual C++ V6

Compiling & Linking

The settings for Microsoft Visual C++ V6 are :

- Create a new project as a 'Win 32 Dynamic Link Library'
- Select Project, then Settings
- Click the 'C/C++' tab
- Select Category : Code generation
- Select Run-time library : Multithreaded DLL
- Click the 'Link' tab
- Add mqutl.lib in the 'object/library modules' window
(If you forget to do this, you get this error message :

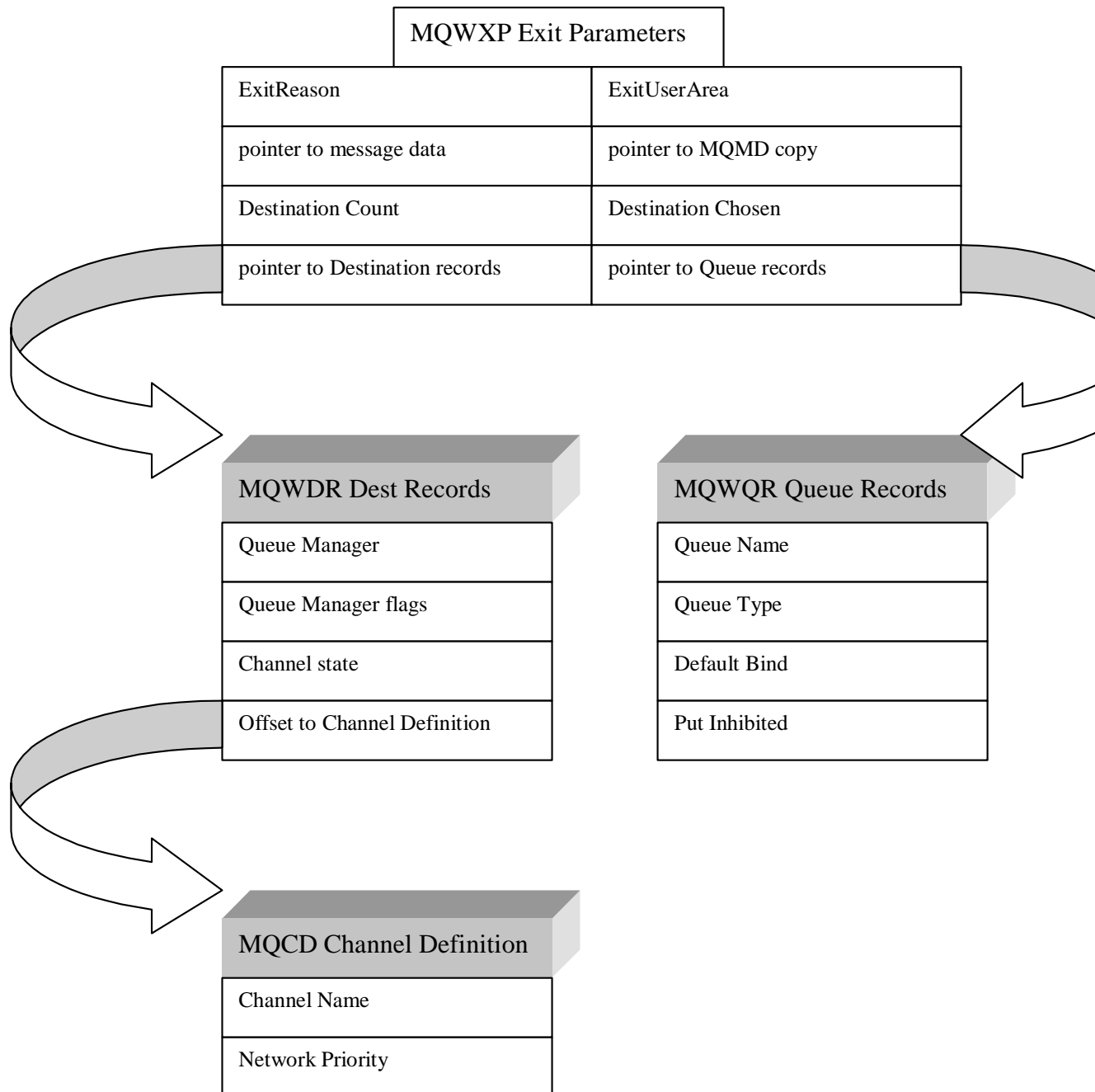
error LNK2001: unresolved external symbol _MQXCLWLN

- You could unclick the 'generate debug info' to make the DLL smaller
- Create a <exitname>.DEF file, and add to the project

Input to the exit

The exit gets driven for a number of reasons (see header file cmqxc.h).

The input to the exit can be shown by the following diagram. Note that only a subsection of fields has been shown here :



You'll notice from the source that a number of specific checks have been added, for example

1. At initialisation time, the first byte of the exit user area is initialised

2. During testing it was noticed that the exit gets called when MQ tries to write to a non-existent (and hence non-clustered) queue.
3. The exit really only deals with queue 'open' and 'put' requests. It leaves all other requests to MQ's decision. In fact, on entry to the exit, the field 'Destination chosen' has already been set by MQ. The coding overrides that field for MQOPEN and MQPUT requests.
4. The field index is used to check the array of queue managers. If, say, the 'search' started at element 3, and there were 3 eligible queue managers, then the code resets index to 0 to allow for the rest to be searched.
5. During testing it was noticed that if all instances of a cluster queue were 'put-disabled' then instead of leaving the message on the cluster xmitq, it ended up on one of the target's DLQ with reason '2268' (MQRC_CLUSTER_PUT_INHIBITED). It was deemed more appropriate to set the exit response to MQXCC_SUPPRESS_FUNCTION in this case.

Known Issues

Issue 1 – warning messages at LINK time

The Intercommunication Guide suggests that the DEF file should contain the following entries :

```

LIBRARY priority
PROTMODE
DESCRIPTION 'Provides MQ Cluster exit'
CODE SHARED LOADONCALL
DATA NONSHARED MULTIPLE
HEAPSIZE 4096
STACKSIZE 8192
EXPORTS clwlFunction

```

When you compile/link the module under Windows 2000 with MS VC++ V6 you get the following warnings which you can ignore :

Linking...

.\priority.def : warning LNK4017: PROTMODE statement not supported for the target platform; ignored

.\priority.def : warning LNK4017: CODE statement not supported for the target platform; ignored

.\priority.def : warning LNK4017: DATA statement not supported for the target platform; ignored

Creating library Debug/priority.lib and object Debug/priority.exp

priority.dll - 0 error(s), 3 warning(s)

Issue 2 – warning message when queue manager is restarted with new exit

When you restart your queue manager , having just added the exit, you may see the following warning message in the /MQSeries/qmgrs/<qmgrname>/errors/amqerr01.log :

06/01/2001 14:31:00

AMQ5010: The system is restarting the WorkLoad Management Server process.

EXPLANATION:

The system has detected that the WorkLoad Management server process (amqzlw0, pid:3252) has stopped and is restarting it.

ACTION:

Save the generated output files which may indicate the reason why the WorkLoad Management process stopped. If the reason the WorkLoad Management Server process stopped is a problem in a WorkLoad Management user exit, correct the problem, otherwise contact your IBM support center.

The process being mentioned (pid 3252) is not in fact a server process. This error appears under MQ V5.1 CSD6, but not under MQ V5.2 CSD1.

Issue 3 – exit does not appear to handle DEFBIND=OPEN as expected

The default processing for DEFBIND=OPEN is to select a queue manager and send all requests to that one until disconnect time. On the next run, MQ selects the next queue manager, etc.

When the SupportPac exit is used under MQ V5.1 CSD6 with cluster queues defined with DEFBIND=OPEN, the exit only ever sends messages to one queue manager. This could possibly be a bug in the base MQ code related to the update of the 'Destination Chosen' field. If on entry to the exit MQ has chosen queue manager number 1, but the exit overrides this to number 2, then on the next invocation, MQ should set the destination to the next queue manager, but this is not happening. Fortunately, this does not occur under MQ V5.2 CSD1.

Problems/Feedback

This is a category 4 SupportPac, and as such has been provided 'asis'. However, if you have any bug reports or suggestions for improvements, then please mail them to:

Admin@Alphacourt.com