Coldfusion MX and Mqseries Revisited

Posted At : April 24, 2006 10:38 PM | Posted By : Mark Kruger
Related Categories: Coldfusion MX 7, Coldfusion Upgrading

In my previous post on the topic of integrating MQSeries with Coldfusion MX I included some sample code for sending and retrieving messages to "put" and "get" queues using an MQManager object. This post has a correction and addition to that original sample code.

In the original sample I included 3 "close()" calls to kill the 2 queue objects and the manager object. The original code looked like this:

```
<cfscript>

// CLOSE OBJECTS
// close the put queue
put.close();
//close the get queue
get.close();
// close the manager object
manager.close();
</cfscript>
```

if you were to run this code on a lightly loaded server it is likely that it would run perfectly fine and not give you any trouble. Under the hood, however, the "manager.close()" method call does not "disconnect" the connection. This leaves orphaned connections that are created and not reused correctly by the queue manager. You can see the results of this behavior by watching the process list. You will see a process created over and over again and not destroyed. Over time it could lead to hundreds of spawned but innactive processes.

Looking at "active connections" in MQ Explorer will also give you a clue. What you will see is connections climbing and never dropping. Over time these "active connections" could end up in the thousands in a very short time. Obviously this is a resource consumptions issue.

The Fix

The fix is amazingly simple. leave the close() command for the queues alone, but change the manager method call to "disconnect()". The new code looks like this:

```
<cfscript>

// CLOSE OBJECTS
// close the put queue
put.close();
//close the get queue
get.close();
// close the manager object
manager.disconnect();
</cfscript>
```

Now take a look at the process list. A few processes will spawn (a pool) and then even out. MQ Explorer should show "active connections" going up *and down* as needed.