

WebSphere MQ environments: Documentation and Auditing

Introduction

This article is about documenting and auditing your WebSphere MQ environments. As an independent consultant I get to see many MQ sites and often need to re-design or implement new messaging streams.

Without previous knowledge of the current setup of the systems that is almost a mission impossible. Existing documentation is often the initial design and not updated with the latest changes. So the first task at a new site is finding out what is installed and how it is configured.

Another challenge is changes to environments, often many changes are applied to the same system and all of a sudden the system no longer operates as expected, because “something” changed and the guessing game starts. The ability to quickly scan the environment and compare it to a previous scan is the icing on the cake.

Lastly when my job is done I need to leave the environment properly documented.

To address the challenges above and handle them with great speed and the least effort (read: time), I have built a tool called MQDocument that helps me do my job.

Exploring (new) environments

When entering a new site there is no time to spend a day or two installing a complicated piece of software, an exploration tool needs to be light weight, installed and “up and running” in minutes. In some cases installing anything is out of the question!

So what was needed was something light, easy and sometimes non installable... but still do the job.

For light weight and easy I decided on developing a number of java classes (develop once, run anywhere), installation is done in seconds and “up and running” came with it. All what was needed was access to the machine put some files in a temp directory and run them.

For non installable I looked at what was there and see if I could reuse it. The most basic form is getting the output of DIS QMGR, DIS Q(*) ALL, DIS CHL(*) ALL, etc and put it in a file... Other sites used Supportpac MS03 (SaveQmgr) or MO71 (GUI Explorer) or MO72 (MQSC Client), z/OS systems produced MAKEDEF output and so on.

Over time all of the above formats were included into the MQDocument modules and all produce the same output format which I call XML/MQ. Converting your existing configuration information to XML/MQ opens a number of new possibilities.

To demonstrate the capability I copy/pasted page 5 to 11 (as is) of the article “Setting up SSL within WebSphere MQ clusters - part 2” in MQUpdate Issue 84 and ran it through MQDocument. With a few minor corrections to the XML/MQ output, it didn’t like the used font for the 0 with dash and didn’t handle the page

numbers and document footers, I had all the MQSC definitions in XML/MQ format in a minute.

The first two definitions on Page 5:

```
*
* Channel definitions for first ABC queue manager.
* The queue manager is a member of cluster SL5ABCL.
*
DEFINE CHANNEL('SL5ABCL.QMABCØ1L') CHLTYPE(CLUSRCVR) +
DESCR('ABC Queue Manager Cluster Receiver') +
CLUSTER('SL5ABCL') +
CONNAME('ABCØ1L.my.domain(1414)') +
SSLCIPH('TRIPLE_DES_SHA_US') +
SSLPEER('CN="LIV.QM*"') +
REPLACE
DEFINE CHANNEL('SL5ABCL.QMGWØ1L') CHLTYPE(CLUSSDR) +
DESCR('Repository Cluster Sender') +
CLUSTER('SL5ABCL') +
CONNAME('GWØ1L.my.domain(1414)') +
SSLCIPH('TRIPLE_DES_SHA_US') +
SSLPEER('CN="LIV.QM*"') +
REPLACE
```

Translated to XML/MQ:

```
<CLUSRCVR Name="SL5ABCL.QMABCØ1L">
  <DESCR>ABC Queue Manager Cluster Receiver</DESCR>
  <CLUSTER>SL5ABCL</CLUSTER>
  <CONNAME>ABCØ1L.my.domain(1414)</CONNAME>
  <SSLCIPH>TRIPLE_DES_SHA_US</SSLCIPH>
  <SSLPEER>CN="LIV.QM"</SSLPEER>
</CLUSRCVR>
<CLUSSDR Name="SL5ABCL.QMGWØ1L">
  <DESCR>Repository Cluster Sender</DESCR>
  <CLUSTER>SL5ABCL</CLUSTER>
  <CONNAME>GWØ1L.my.domain(1414)</CONNAME>
  <SSLCIPH>TRIPLE_DES_SHA_US</SSLCIPH>
  <SSLPEER>CN="LIV.QM"</SSLPEER>
</CLUSSDR>
```

Documentation

XML/MQ can be processed by any XML capable system, without knowing anything about MQ. I have added the ability to “compare” the XML (more on that later) and to generate “standard” documentation from the XML using XSL (Extensible Stylesheet Language).

XML and XSL were chosen because it is flexible and open so you can use any XSLT engine to do your transformations (even use a browser!) and make changes to the look and feel of the “base” supplied stylesheet to match for example corporate standards.

Below you will find the look and feel of the transformed XML/MQ from above using the “base” supplied stylesheet in a browser:

| Channel Definition | | |
|--------------------|------------------------------------|----------|
| Name | SL5ABCL.QMABC01L | Name |
| Cluster | SL5ABCL | CLUSTER |
| Connection Name | ABC01L.my.domain(1414) | CONNNAME |
| Description | ABC Queue Manager Cluster Receiver | DESCR |
| CipherSpec | TRIPLE_DES_SHA_US | SSLCIPH |
| Distinguished Name | CN="LIV.QM" | SSLPEER |

| Channel Definition | | |
|--------------------|---------------------------------|----------|
| Name | SL5ABCL.QMGW01L | Name |
| Cluster | SL5ABCL | CLUSTER |
| Connection Name | GW01L.my.domain(1414) | CONNNAME |
| Description | Repository Cluster Sender | DESCR |
| CipherSpec | TRIPLE_DES_SHA_US | SSLCIPH |
| Distinguished Name | CN="LIV.QM" | SSLPEER |

Needles to say (or maybe not) this can be copied into or imported into a Word document (as I am writing this article in Word) or any other word processor like for example OpenOffice.

Another “application” is to simply put these files on a file server and so providing “read only” access to the actual MQ configuration details using a browser.

Security / Access Control Documentation

The sample definitions used above did not include security definitions, to show the capability in this area I will use some output from one of my test Queuemanagers.

Since MQ 5.2 CSD (Fixpacks and Refreshpacks nowadays) 4 the amqoamd command has the –s switch to output the OAM information in setmqaut format. With this option you can either use the actual output of the amqoamd command or use the original scripts you used to set the authorities.

The sample QMGRA amqoamd –s output:

```
setmqaut -m QMGRA -t qmgr -p Michael@LAPTOP1 +altusr +connect +inq +set +setall +setid
+chg +crt +dlt +dsp
setmqaut -m QMGRA -n SYSTEM.DEFAULT.LOCAL.QUEUE -t queue -p Michael@LAPTOP1
+browse +get +inq +passall +passid +put +set +setall +setid +chg +clr +dlt +dsp
setmqaut -m QMGRA -n SYSTEM.DEFAULT.ALIAS.QUEUE -t queue -p Michael@LAPTOP1
+browse +get +inq +passall +passid +put +set +setall +setid +chg +clr +dlt +dsp
setmqaut -m QMGRA -n SYSTEM.DEFAULT.REMOTE.QUEUE -t queue -p Michael@LAPTOP1
+browse +get +inq +passall +passid +put +set +setall +setid +chg +clr +dlt +dsp
setmqaut -m QMGRA -n SYSTEM.DEF.RECEIVER -t channel -p Michael@LAPTOP1 +inq +chg
+dlt +dsp +ctrl +ctrlx
setmqaut -m QMGRA -n SYSTEM.DEF.SENDER -t channel -p Michael@LAPTOP1 +inq +chg
+dlt +dsp +ctrl +ctrlx
setmqaut -m QMGRA -n TEST123 -t queue -p Michael@LAPTOP1 +browse +get +inq +passall
+passid +put +set +setall +setid +chg +clr +dlt +dsp
setmqaut -m QMGRA -n TEST456 -t queue -p Michael@LAPTOP1 +browse +get +inq +passall
+passid +put +set +setall +setid +chg +clr +dlt +dsp
```

Translated to XML/MQ:

```
<ACL Type="qmgr" Name="" Principal="Michael@LAPTOP1">
<Connect>1</Connect><Inquire>1</Inquire><AltUser>1</AltUser><Set>1</Set><SetAll>1</SetAl
l><SetId>1</SetId><Change>1</Change><Create>1</Create><Delete>1</Delete><Display>1</
Display></ACL>
```

```

<ACL Type="queue" Name="SYSTEM.DEFAULT.LOCAL.QUEUE"
Principal="Michael@LAPTOP1">
<Browse>1</Browse><Inquire>1</Inquire><Get>1</Get><Put>1</Put><PassAll>1</PassAll><PassId>1</PassId><Set>1</Set><SetAll>1</SetAll><SetId>1</SetId><Change>1</Change><Clear>1</Clear><Delete>1</Delete><Display>1</Display></ACL>
<ACL Type="queue" Name="SYSTEM.DEFAULT.ALIAS.QUEUE"
Principal="Michael@LAPTOP1">
<Browse>1</Browse><Inquire>1</Inquire><Get>1</Get><Put>1</Put><PassAll>1</PassAll><PassId>1</PassId><Set>1</Set><SetAll>1</SetAll><SetId>1</SetId><Change>1</Change><Clear>1</Clear><Delete>1</Delete><Display>1</Display></ACL>
<ACL Type="queue" Name="SYSTEM.DEFAULT.REMOTE.QUEUE"
Principal="Michael@LAPTOP1">
<Browse>1</Browse><Inquire>1</Inquire><Get>1</Get><Put>1</Put><PassAll>1</PassAll><PassId>1</PassId><Set>1</Set><SetAll>1</SetAll><SetId>1</SetId><Change>1</Change><Clear>1</Clear><Delete>1</Delete><Display>1</Display></ACL>
<ACL Type="channel" Name="SYSTEM.DEF.RECEIVER" Principal="Michael@LAPTOP1">
<Control>1</Control><Reset>1</Reset><Change>1</Change><Delete>1</Delete><Display>1</Display></ACL>
<ACL Type="channel" Name="SYSTEM.DEF.SENDER" Principal="Michael@LAPTOP1">
<Control>1</Control><Reset>1</Reset><Change>1</Change><Delete>1</Delete><Display>1</Display></ACL>
<ACL Type="queue" Name="TEST123" Principal="Michael@LAPTOP1">
<Browse>1</Browse><Inquire>1</Inquire><Get>1</Get><Put>1</Put><PassAll>1</PassAll><PassId>1</PassId><Set>1</Set><SetAll>1</SetAll><SetId>1</SetId><Change>1</Change><Clear>1</Clear><Delete>1</Delete><Display>1</Display></ACL>
<ACL Type="queue" Name="TEST456" Principal="Michael@LAPTOP1">
<Browse>1</Browse><Inquire>1</Inquire><Get>1</Get><Put>1</Put><PassAll>1</PassAll><PassId>1</PassId><Set>1</Set><SetAll>1</SetAll><SetId>1</SetId><Change>1</Change><Clear>1</Clear><Delete>1</Delete><Display>1</Display>
</ACL>

```

Below you will find the look and feel of the transformed XML/MQ from above using the “base” supplied stylesheet in a browser:

| Access Control Summary for Queue Manager: | | | | | | | | | | | | | |
|---|---|---------------------------------|---------------------------------|---------------------------------|-------------|---------------------------------|----------------------------|-------------|----------------------------|-----------------------|----------------------------|----------------------------|---------------------------------|
| QMGR | | | | | | | | | | | | | |
| (principal) Michael@LAPTOP1 | | | | | | | | | | | | | |
| | | C o n t r o l | I n q u i r e | A l t U s e r | | C o n t r o l | R e s e t | S e t | S e t A l l | S e t I d | C h a n g e | D e l e t e | D i s p l a y |
| qmgr | | ✓ | ✓ | ✓ | | | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| channel | SYSTEM.DEF.RECEIVER | | | | | ✓ | ✓ | | | | | ✓ | ✓ |
| channel | SYSTEM.DEF.SENDER | | | | | ✓ | ✓ | | | | | ✓ | ✓ |
| | | B r o w s e | I n q u i r e | G e t | P u t | P a s s A l l | P a s s I d | S e t | S e t A l l | S e t I d | C h a n g e | D e l e t e | D i s p l a y |
| queue | SYSTEM.DEFAULT.ALIAS.QUEUE | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| queue | SYSTEM.DEFAULT.LOCAL.QUEUE | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| queue | SYSTEM.DEFAULT.REMOTE.QUEUE | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| queue | TEST123 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| queue | TEST456 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

Note: this is output from a V6 Queuemanager, new controls are added in V6, including channel authorizations.

Auditing

When an environment is documented as shown in the samples above, people without knowledge of MQSC or setmqaut formats can more easily understand what is there and what is not. (at least the documentation gives them the impression they can understand it...). The ability to produce these reports on the “fly” gives great confidence in the Security Officer’s department :-)

Besides visually attractive reports, you still have the base XML/MQ data which can be kept for safe keeping and can be refreshed used over and over (the XML/MQ information can be kept in CVS for example).

Let’s pretend I make some changes to the fictive system I just documented (afterall the MQSC came from another article and the ACL information from my test Queuemanager)

On the Cluster Receiver channel SL5ABCL.QMABC01L I change the CONNAME to 127.0.0.1(1414) and on the ACL’s I change Put authorization to No for the TEST123 and TEST456 Queues.

I “rerun” the MQDocument collection routines and a new snapshot of the environment is created. As I kept the old snapshot I can now compare the two snapshots taken at different points in time and see what changed:

Below you see the output window of the mqdocdiff class where -i is supplied as the new XML/MQ snapshot and -p is supplied as the previous XML/MQ snapshot. It also reports the QMGR for which differences are reported as multiple Queuemanager configurations can be put into a single XML/MQ snapshot file.

```
ca: Command Prompt
D:\Mqmd\mqdocument>java mqdocdiff -i xephon.sample.new.xml -p xephon.sample.xml
mqdocdiff 1.0.5
Program license:
-Contractor License- Michael Dag

Ready to report on differences for QMgr: QMGRA
Changed CLUSRCUR SL5ABCL.QMABC01L on QMGRA
      CONNAME changed from ABC01L.my.domain(1414) to 127.0.0.1(1414)
Changed ACL for Principal Michael@LAPTOP1, queue TEST123 on QMGRA
      Put changed from Allowed to not Allowed
Changed ACL for Principal Michael@LAPTOP1, queue TEST456 on QMGRA
      Put changed from Allowed to not Allowed

D:\Mqmd\mqdocument>
```

The mqdocdiff program “knows” about MQ Objects and it’s structure, so for example ALTDATE and ALTTIME Attributes on Transmission Queues and Channels are ignored if no other changes can be found.

Conclusion

Having the MQDocument tools available to me makes me as an independent consultant more flexible, better equipped and don't waste my precious time copying and pasting information, in addition I don't spend a lot of time doing boring repetitive jobs like producing reports or documentation ...