65

MQ update

November 2004

In this issue

A itc PUBLICATION

MQ Update

Published by

Xephon Inc
PO Box 550547
Dallas, Texas 75355
USA

Phone: 214-340-5690
Fax: 214-341-7081

Editor

Trevor Eddolls
E-mail: trevore@xephon.com

Publisher

Bob Thomas
E-mail: info@xephon.com

Subscriptions and back-issues

A year's subscription to *MQ Update*, comprising twelve monthly issues, costs $380.00 in the USA and Canada; £255.00 in the UK; £261.00 in Europe; £267.00 in Australasia and Japan; and £265.50 elsewhere. In all cases the price includes postage. Individual issues, starting with the July 2000 issue, are available separately to subscribers for $33.75 (£22.50) each including postage.

Contributions

When Xephon is given copyright, articles published in *MQ Update* are paid for at the rate of $160 (£100 outside North America) per 1000 words and $80 (£50) per 100 lines of code for the first 200 lines of original material. The remaining code is paid for at the rate of $32 (£20) per 100 lines. To find out more about contributing an article, without any obligation, please download a copy of our *Notes for Contributors* from www.xephon.com/nfc.

MQ Update on-line

Code from *MQ Update*, and complete issues in Acrobat PDF format, can be downloaded from our Web site at www.xephon.com/mq; you will need to supply a word from the printed issue.

This issue is dedicated to the memory of Chris Bunyan, co-founder of Xephon and creator of the *Update* journals.

Printed in England.

WMQ security considerations for z/OS and RACF

INTRODUCTION

Our shop runs WebSphere MQ in a z/OS (Version 1 Release 2) environment with RACF. MQ provides messaging and queueing support for CICS, IMS, TSO, and batch amongst others. Security is pretty tight, but years of experience in the mainframe environment, talking to many IBM consultants, and reading much technical literature, have revealed a few points that may be of interest to those working in a similar environment. This is not a complete security overview of WebSphere MQ on z/OS, but it provides some simple guidance for those who run MQ on the mainframe.

Z/OS ISSUES

The following security issues apply to z/OS. The initial default security set-up on z/OS can be very basic, and the security profiles defined do not provide much granularity:

- Change the default user ID – it is strongly recommended that you change the CMDUSER user ID for command security checks on z/OS from its default setting of CSQOPR. This can be changed in CSQ6SYSP. Furthermore, the ID should be given restricted authority.
- Secure command server queues – in z/OS the command server queues such as SYSTEM.COMMAND.INPUT are a major entry point into WebSphere MQ. You should devote considerable attention to their security.
- ISPF panels – access to the ISPF panels available for WebSphere MQ for z/OS should be strictly controlled. This will help ensure that objects are not started or stopped, changed, or defined without adequate authority.

RACF ISSUES

The following security issues apply to RACF in a WebSphere MQ environment:

- Queue naming – giving some thought to the naming conventions for z/OS will go a long way towards simplifying your RACF profiles. A hierarchical system such as ‘enterprise.dept.application’ will allow you flexibility in assigning access to ‘enterprise.*’, ‘enterprise.dept.*’, etc. Don’t put the queue type at the start of the queue.
- Blank user IDs – in RACF, user IDs can be blank. Make sure that access to these user IDs is carefully controlled. Furthermore, z/OS’s default security time-out value is quite large. You should reduce the time interval to decrease the security risk. You can find the value in the queue manager object.
- Generic profiles and the RESLEVEL – if you define too many generic profiles to RACF, or do not set up the RESLEVEL correctly, this will add a high overhead to system maintenance. Too many generic profiles will make it difficult to control access to all your different resources. Some thought needs to be given to what resources need protecting and how this will be achieved before RACF definitions are changed.
- APPC security – like many users, we flow user IDs through a LU6.2 APPC channel. If this is something that you do you should ensure that you define APPC security correctly; so make sure you define conversation security and session security with adequate passwords, and, crucially, the security access list and trusted group names. Also, you can create RACF profiles in the APPCLU class to define more security characteristics for LUs and for conversations between the LUs. These profiles differ, depending on whether or not the LU is a member of a VTAM generic resource group. This is covered in Chapter 10, ‘Setting Up Network Security’, in IBM’s *z/OS V1R5.0*

MVS Planning: APPC/MVS Management (SA22-7599-03).

MQ ISSUES

The following considerations are more general MQ security issues. Attributes you should give attention to are default settings and items, user IDs, and aliases:

- Default settings and items:
  - *delete default items* – as with much software, the default options that are set during the installation procedure can come back to haunt us (like the default user ID for z/OS). In the real world these default settings are often forgotten after installation is complete. The installation of WebSphere MQ can leave a number of default objects on a system. These can then be used as a springboard for security breaches. Check for objects that start with SYSTEM.DEFAULT, and find out what they do. If they are not needed by your system, they should be deleted or protected before you move to a production environment. IBM recommends that if you have any doubts as to why an object is there, 'rename it and see whether the system requires having it back; if not, delete it'.
  - *change the default 'blank' MCAUSER* – another default issue that needs consideration is the default 'blank' MCAUSER definition. If the receiver channel definitions have been set up with PUTAUT = DEF and the MCAUSER attribute left blank, there could be problems. When it comes to channel definitions you should never allow the default 'blank' for MCAUSER. If this is allowed to happen, the only user ID that will be checked for any PUTs will be the channel initiator address space user ID. Channels need a high degree of protection because they are a significant access point into an enterprise.

- User IDs:
  - *ALTUSER ID* – allowing the use of an alternative user ID in the MQMD has the potential to let an attacker subvert object authority. Carefully consider your use of ALTUSER ID.
  - *CHINIT user IDs* – another user ID issue comes with CHINIT user IDs. Running CHINITS under different user IDs in a queue sharing group will allow you to apply different access profiles to different access paths.
- Aliases:
  - *use aliases* – aliases are an extremely powerful tool for setting different permissions on a queue for read and write access. The advantage comes from not having to create large numbers of different profiles or permissions.
  - *alias the DLQ* – if you alias the 'read' Dead Letter Queue (DLQ) it will allow you to give many users browse access but far fewer will have write access. It also makes it more difficult for an attacker to hide their tracks through the DLQ and deleting log files.
- Other considerations:
  - *the MQM group* – do not put all your users who need access to tools such as MQ Explorer into the mqm group. It is best to create a new group with similar access rights to mqm. You can then create a hierarchical structure within this new group. Such a structure could cater for different levels of users by creating groups for categories such as mqadmin, mqops, etc. This gives you much more granularity in your control over different user groups within your enterprise.
  - *object definitions* – a number of interfaces allow WebSphere MQ object definitions to be manipulated

(ie the MQ Explorer interface, the RUNMQSC command line utility, etc). It is essential to identify the users of these products and grant access through the OAM and the ESM on z/OS.

– *encryption and channels* – if you are running WebSphere MQ Version 5.3, you will be able to use the Secure Sockets Layer (SSL) to authenticate queue manager-to-queue manager and MQ client-to-queue manager connections. This provides strong channel authentication before any data is passed. SSL can also be used to provide bulk encryption for all data flowing across a channel. The encryption choices for the secret key are RC2, RC4, DES, T-DES, or AES. The possible key sizes are 40 bits (RC2 and RC4), 56 bits (DES and 56 bit RC4), 112 bits (T-DES), 128 bits (128-bit RC4 and AES), and 256 bits (AES). The strength of the secret key is important because WebSphere MQ re-uses the same symmetric encryption key that it created when the channel started for all message transfers on that channel. The same key will be used until the channel stops. Only when the channel restarts will a new key be created. A lot of channels are either never shut down or are very long running indeed. This gives an attacker a lot of opportunity to hack the key. The first remedy for this is to stop and start channels on a regular basis, which will allow for the creation of a new symmetric key. The other consideration is to increase the key length, if you have not already done so. IBM recommends using a key length of at least 112 bits, which means using RC4, Triple DES, or AES, with AES being the strongest. Unfortunately, z/OS and the mainframe hardware do not support AES. Some systems may have support for Triple DES, which should be used if possible; otherwise choose the longest supported key length.

NEW THREATS

One of the biggest security problems we have encountered in recent years is the increasing use of wireless LANs using the WiFi (802.11b) standard. At first glance this might appear somewhat remote from WebSphere MQ security in the z/OS environment. However, wireless access points, or stations, announce their presence across the airwaves by broadcasting a packet containing their service set identifier (SSID) and all security turned off is the default setting. Unfortunately, an attacker outside a building using tools, usually called 'stumblers', to find wireless access points can easily determine whether an access point exists and whether the security is turned on.

Often the wireless access point will route traffic to a DHCP server that will provide the machine with a TCP/IP address, a gateway address, and the DNS suffix of the parent company. A port scan focusing on WebSphere MQ ports such as 1414, 1415, etc, will often produce a number of queue manager listeners. If these are pinged they will often yield a DNS name that is the same name as the queue manager's. From there the attacker can access MQ Explorer and their newly-found queue and begin to manipulate objects, implement remote queue manager definitions, and make other such mischief.

While the concept of the secure perimeter is a little fuzzy these days, this kind of exposure cannot be allowed. You may think that that this is not a problem in your enterprise because, for example, you do not use WiFi, or you have a security policy that says that staff cannot use WiFi. Think again. The small size of WiFi products means that it is easy for staff to bring them into work, and, if not properly secured, there will be gaping holes in your security.

Try this experiment – download a copy of one of the popular 'stumbler' utilities used to detect wireless access points. Most major operating systems have stumbler programs. For Windows there is NetStumbler (www.netstumbler.com), for Linux there is Kismet (www.kismetwireless.net), for MAC OS

there is MacStumbler (www.macstumbler.com), and for the PocketPC there is MiniStumbler, which is a port of NetStumbler. A detailed list of stumblers can be found at www.wardriving.com. Install the software onto your laptop and do your own mini-'wardrive' around your enterprise. (Wardriving is a term used by enthusiasts who gather data about wireless networks in a given area. They do this by driving around with laptops running stumbler programs listening for the broadcast beacons of APs.) Of course, in your case you will be walking around the enterprise with a laptop or PDA, but the concept is the same. You might be surprised to find what turns up! It is reminiscent of the early days of the Internet with staff sneaking dial-up modems into the office. I am afraid that this might involve some footwork; you may have to visit some of the offenders and explain the error of their ways, and remove the rogue access points.

At the present we believe that removal of the unauthorized access points is probably the best policy, because WiFi security is still poor. Wired Equivalent Privacy (WEP) is part of the IEEE 802.11 standard and is used to secure wireless networks. However, several serious weaknesses have been identified by cryptographers, and WEP was superseded by Wi-Fi Protected Access (WPA) in 2003, and then by the full IEEE 802.11i standard (WPA2) in 2004. Remember WEP provides the minimum level of security needed to deter casual snooping, it should not form part of an enterprise security set-up.

Furthermore, this is not just a one time job – you will need to monitor the enterprise for rogue access points on a regular basis. The crucial point is to create and distribute a security policy and educate your staff about the dangers of unencrypted wireless access within an enterprise.

In the context of WebSphere MQ you have to make sure that the Access Control Lists are implemented and MCAUSER is used properly, otherwise the enterprise is entirely exposed. Using SSL (Secure Sockets Layer) encryption between queue

managers will have absolutely no effect because the attacker is already within the firewall!

CONCLUSIONS

The principal issue that has been raised by the above overview is the acceptance of the default settings during set-up. It would be beneficial to review the default settings on a regular basis to make sure that they are still appropriate to the running of your systems.

MQ security has often been overlooked in the past. However, there is a greater awareness today that message queueing infrastructure can provide a gateway into an organization if it is not protected. Certainly, IBM itself has been steadily increasing its emphasis on middleware security in the last few years. The release of Secure Sockets Layer support in WebSphere MQ Version 5.3, and the release of WebSphere MQ Extended Security Edition Version 5.3, in June 2003 for some platforms, both highlight IBM's concern over security.

We are fortunate that WebSphere MQ is probably the most secure and resilient of the message-oriented middleware products in the enterprise market today, but it would be beneficial for organizations to review their MQ security once in a while.

John Edwards
Systems Administrator (UK)

Early experiences on MQ 5.3 for Linux

This article attempts to raise the level of awareness of people using WebSphere MQ (referred to as MQ) on the Linux platform. It is, however, restricted to the non-mainframe platforms and specifically to RedHat Linux and SuSE Linux

using Intel processors. It won't teach you Linux, although some basic commands are shown.

BACKGROUND

IBM officially supports the Linux operating system on both the mainframe and distributed platforms.

Support for MQ on the Linux platform came with MQ Version 5.2 (October 2000).

The number of companies looking at Linux as an alternative to Windows, or other Unix offerings (Solaris, AIX), and even the mainframe, is on the increase.

From a personal perspective, there have now been two projects using MQ on Linux during the last year.

The reasons customers are looking at, and migrating to, Linux are varied and complicated and beyond the scope of this article. However, perceived cost savings, flexibility (eg being able to add 'blades'), and 'internal politics' form a large part of the equation.

HOW TO PROCEED

I believe the best way to get a feel for Linux, and run MQ, is by actually using it. A rudimentary knowledge of Unix is preferable but not a strict prerequisite.

The best way to approach this is to accept that mistakes will be made (after all, that's how knowledge is obtained).

WHICH LINUX?

There are many distributions (also known as 'distros') of Linux; all are free (downloadable from the Internet), although a charge is made by companies who copy it to a CD for you. Free software is great, but would you bet your business on it? Who is going to support it? Which databases are supported? Will it scale? What about failover? Is it secure?

At one client site, RedHat was chosen as the supplier and RedHat Advanced Server V2.1 and V3 as the operating system. For personal use, SuSE V9.1 (either the personal or professional edition) is very easy to install. The KDE desktop looks very similar to Windows and the YAST utility on SuSE is very good at applying fixes (similar to Windows Update).

PREREQUISITES FOR MQ ON LINUX

Check the IBM Web site (http://www-306.ibm.com/software/integration/mqfamily/platforms/supported/wsmq_for_linux_intel_5_3.html) to ensure that the hardware and software available meet the criteria.

IBM is very conservative on its support page, which lists relatively old versions of Linux software. For example:

- SuSE SLES-7. The latest release is 9.1.
- RedHat Professional 7.3. The latest non-commercial release is 8. The commercial release (not mentioned by IBM) is currently 2.1, and the latest 3.1.

To be fair, however, it must be realized that in the Linux world what is important is the level of the so-called 'kernel' and some of the software libraries.

OBTAIN MQ FOR LINUX

Unless a licensed copy of the software is already present, IBM makes a 90-day trial version available for download on Intel processors from http://www14.software.ibm.com/webapp/download/preconfig.jsp?id=2004-02-26+15%3A37%3A22.610455R&S_TACT=104AH%20W42&S_CMP=&s=.

INSTALLATION

After registering and downloading the software, there will be a file called *mq53_trial_lin_int.tar.gz,* which is around 130MB.

The ‘gz’ extension means it is a zipped file created by a Unix utility called gzip.

The Windows Winzip utility in fact ‘understands’ this format.

Login as ‘root’ on the Linux machine.

Create a suitable working directory, eg **mkdir mqinstall**.

Unzip the file (increases to about 150MB), eg **gunzip mq53_trial_lin_int.tar.gz**.

Untar it, eg **tar -xvf mq53_trial_lin_int.tar**.

This creates a number of files with the .rpm ending. These files are in RedHat Package Manager format used by both RedHat and SuSE – other distros may use a different file format.

Read the install file:

```
cd README
cd en_US
vi readme
```

(When finished, use **ESC : q !** to get out of vi.)

Check kernel level: **uname –r**.

Response from the system: 2.6.5-7.95-default (for example).

At this point stop and read the section entitled *Specific issues* because you may need to set an important environment variable.

Accept the licence before the install: *./mqlicense.sh*.

It responds with:

```
Displaying license agreement on :0.0
Exited with: 9
Agreement accepted: Proceed with install.
```

Install the rpm packages in the following order:

- rpm -i MQSeriesRuntime-5.3.0-1.i386.rpm
- rpm -i MQSeriesSDK-5.3.0-1.i386.rpm

- rpm -i MQSeriesServer-5.3.0-1.i386.rpm
- rpm -i MQSeriesClient-5.3.0-1.i386.rpm
- rpm -i MQSeriesSamples-5.3.0-1.i386.rpm
- rpm -i MQSeriesJava-5.3.0-1.i386.rpm
- rpm -i MQSeriesMan-5.3.0-1.i386.rpm.

Failing to use the correct order will result in the following type of error:

```
~/mqinstall # rpm -i MQSeriesServer-5.3.0-1.i386.rpm
error: Failed dependencies:
        MQSeriesRuntime = 5.3.0-1 is needed by MQSeriesServer-5.3.0-1
        MQSeriesSDK = 5.3.0-1 is needed by MQSeriesServer-5.3.0-1
```

Check RPM database:

```
rpm -qa | grep -i mq
```

This should return the same list as above.

Check the version of MQ (this shows that no CSDs have been applied):

```
javsuse9:~/mqinstall # mqver
Name:        WebSphere MQ
Version:     530
CMVC level:  p000-L021028
BuildType:   IKAP - (Production)
Apply CSD7.
```

Download the fix, unzip and untar it. Apply it:

```
rpm -i MQSeriesRuntime-U496732-5.3.0-7.i386.rpm
rpm -i MQSeriesSDK-U496732-5.3.0-7.i386.rpm
rpm -i MQSeriesServer-U496732-5.3.0-7.i386.rpm
rpm -i MQSeriesClient-U496732-5.3.0-7.i386.rpm
rpm -i MQSeriesSamples-U496732-5.3.0-7.i386.rpm
rpm -i MQSeriesJava-U496732-5.3.0-7.i386.rpm
rpm -i MQSeriesMan-U496732-5.3.0-7.i386.rpm
```

A check against the RPM database shows that the previous releases are still there. This is correct and allows MQ to be restored to a previous level if required.

Check the version and fix level – it should show ‘Version: 530.7 CSD07’.

SPECIFIC ISSUES

1 Running *mqlicense.sh* results in a segmentation fault.

On RedHat Linux AS3, the *mqlicense.sh* script may cause a segmentation fault and may happen even if the LD_ASSUME_KERNEL (see issue 2 below) has been set.

If this happens, check the following line in the script:

```
# Set JRE location
JRE=${PROGPATH?}/lap/jre/bin/java
```

and change it to the correct Java runtime library.

2 Running standard MQ commands like **dspmq** and **crtmqm** on RedHat V3 results in MQ internal errors:

```
> dspmq
AMQ6090: WebSphere MQ was unable to display an error message 20006220.
```

This is accompanied by an FDC file in */var/mqm/errors*:

```
WebSphere MQ First Failure Symptom Report
=========================================
Date/Time          :- Tuesday August 24 08:31:31 BST 2004
Host Name          :- xldn0351dap (Linux 2.4.21-9.0.1.ELsmp)
PIDS               :- 5724B4104
LVLS               :- 530.6  CSD06
Product Long Name  :- WebSphere MQ for Linux for Intel
Vendor             :- IBM
Probe Id           :- XY439010
Application Name   :- MQM
Component          :- xcsProgramInit
Build Date         :- Feb 11 2004
CMVC level         :- p530-06-L040211
Build Type         :- IKAP - (Production)
UserID             :- 00029398 (UNKNOWN)
Program Name       :- dspmq
Thread-Process     :- 00011386
ThreadingModel     :- Unknown
Major Errorcode    :- MQRC_ENVIRONMENT_ERROR
Minor Errorcode    :- OK
Probe Type         :- MSGAMQ07DC
Probe Severity     :- 4
Probe Description  :- AMQ6090: WebSphere MQ was unable to display an
                                                    error message 7DC.
FDCSequenceNumber  :- 0
```

```
No MQM Function Stack Available
```

Converting X'7DC' into decimal gives a value of 2012, which is the MQ error, eg:

```
> mqrc 2012
      2012  0x000007dc  MQRC_ENVIRONMENT_ERROR
```

A search on the Internet shows that a lot of users have hit this problem.

It is related to the introduction of a new ‘threading’ library called NPTL, which stands for Native POSIX Threading Library. It was announced by RedHat in September 2002 and it is not only standardized but a lot faster.

Unfortunately, it is not supported by MQ V5.3 (nor by DB2 UDB or Oracle).

To overcome this problem an environment variable has to be set that tells the operating system to use the older LinuxThreads library.

Two options are available:

– LD_ASSUME_KERNEL.

– AMQ_THREADMODEL_RESET (needs CSD5 as a minimum).

The main difference between these is that the first one affects all processes, whereas the second affects only MQ. Remember to set the environment variable for all MQ-related work: it is best to create a script, or better still, set the variable when a user logs on.

If all MQ work runs under the ‘mqm’ userid as well as the ‘bash’ shell, use the ‘.bashrc’ file (in mqm’s home directory) to set up the variable. This file is typically also used to add items to the PATH and CLASSPATH, eg:

```
LD_ASSUME_KERNEL=2.4.19
export LD_ASSUME_KERNEL
```

Alternatively, set AMQ_THREADMODEL_RESET to any

value (eg =1). The presence of this variable effectively sets the kernel to 2.4.19.

Log off and log on again and check that the variable is set.

3 MQ produces FDCs when using ‘crtmqm’, ‘strmqm’, and ‘strmqcsv’.

Whenever any of the above commands were issued, MQ created an FDC with component string ‘XlsLateEventAllocation’ – but the command did work. Unfortunately, the error was spotted when MQ was tested under RedHat AS3 and it was thought that the latest RedHat release was to blame.

Closer investigation, however, showed that this happened not only under AS2.1, but also under all MQ CSDs (1 to 7).

After the usual traces were sent to IBM, the response was that this can happen if a non-zero return code from a system call to get the ‘group id’ is obtained.

One thing that all the Linux servers had in common was the use of NIS (Network Information Service). In a ‘traditional’ Unix environment, all userids and groups are locally defined in */etc/passwd* (userid mqm) and */etc/group* (group mqm) and any user/process issuing MQ administration commands must be a member of the mqm group. It was during this checking that MQ produced the FDC.

In the Unix world, behind each userid and group is its numeric equivalent and the ‘convention’ is to allocate numbers higher than 500 for non-system ids.

In a NIS environment, different commands have to be issued to find out about users and groups (note: yp stands for yellow pages).

To list the attributes of the user mqm:

```
ypmatch mqm passwd
```

To list the attributes of the group mqm:

```
ypmatch mqm group
```

It turned out that the mqm NIS userid was a member of a group whose gid (group id number) was 75. The system call to find the name associated with that gid first looks on the local system and unfortunately there was one called ‘rpcuser’. Because this wasn’t ‘mqm’ an FDC was produced. This isn’t an MQ and Linux problem, but a set-up problem with NIS.

DIFFERENCES BETWEEN LINUX AND OTHER UNIX SYSTEMS

One of the big differences under Linux is how system ‘processes’ and ‘threads’ are used. Issue the following display command on a Sun Solaris system to list all the processes associated with queue manager RUUD:

```
> ps -ef | grep RUUD
     mqm 16176     1  0   Jul 30 ? 0:00 runmqlsr -m RUUD -t tcp -p 14141
     mqm 25455 25453  0   Aug 18 ? 0:00 amqhasmx RUUD /var/mqm
     mqm 25460 25453  0   Aug 18 ? 0:56 amqzlaa0 -mRUUD -fip0
     mqm 25457 25453  0   Aug 18 ? 0:00 /opt/mqm/bin/amqrrmfa -t2332800
-s2592000 -p2592000 -g5184000 -c3600 -m RUUD
     mqm 25456 25453  0   Aug 18 ? 0:00 amqzllp0 -mRUUD ?
     mqm 25453     1  0   Aug 18 ? 0:00 amqzxma0 -m RUUD
     mqm 25454 25453  0   Aug 18 ? 0:00 /opt/mqm/bin/amqzfuma -m RUUD
     mqm 25492     1  0   Aug 18 ? 0:29 amqpcsea RUUD
     mqm 25459 25453  0   Aug 18 ? 0:00 /opt/mqm/bin/runmqchi -m RUUD
vanzunru  8753  8734  0 08:56:57 pts/8    0:00 grep RUUD
     mqm 25458 25453  0   Aug 18 ? 0:00 /opt/mqm/bin/amqzdmaa -m RUUD
```

Counting the number of these processes showed there were 10:

```
> ps -ef | grep -v grep | grep -c RUUD
10
```

If this same command is issued under RedHat Linux it gives a total of 51 processes and if you have scripts that check on these, changes may be required.

```
bash-2.05b$ ps -ef | grep -v grep | grep -c RUUD
51
```

The MQ listener processes on Linux clearly highlight the difference with Solaris:

```
mqm 12047 11765 09:00 pts/1 00:00:00 runmqlsr -m RUUD -t tcp -p 14141
mqm 12048 12047 09:00 pts/1 00:00:00 runmqlsr -m RUUD -t tcp -p 14141
mqm 12050 12048 09:00 pts/1 00:00:00 runmqlsr -m RUUD -t tcp -p 14141
mqm 12051 12048 09:00 pts/1 00:00:00 runmqlsr -m RUUD -t tcp -p 14141
mqm 12052 12048 09:00 pts/1 00:00:00 runmqlsr -m RUUD -t tcp -p 14141
```

SUPPORT FROM MQ ISVS

Although it is perfectly possible to manage a small number of Linux systems using scripts, many companies prefer to have a single solution for their MQ monitoring and administration functions.

Without going into the merits of specific products, here is a short list of contenders that were evaluated:

- BMC with Patrol for MQ. Note that support for Linux started on 6 August 2004.
- MQSoftware with Qpasa!
- Candle (now owned by IBM) with Pathway XE for MQ.
- Micromuse with DCM.

REFERENCES

Quick Start Guide for Linux with MQ: http://www-106.ibm.com/developerworks/ibm/library/l-ss3-mq.

Discussion on the NPTL threads: http://kerneltrap.org/node/view/422.

Some useful commands:

- List version of the ‘kernel’:

  ```
  uname -r
  ```

- List version of the operating system:

  ```
  cat /etc/redhat-release
  ```

  which may return:

  ```
  Red Hat Linux Advanced Server release 2.1AS (Pensacola)
  ```

or:

```
Red Hat Enterprise Linux AS release 3 (Taroon Update 1)
```

Use **cat /etc/SuSE-release** for SuSE.

- List environment variables:

```
env
```

- Check what is installed via RPM:

```
rpm -qa | grep XXX
```

where *XXX* is the software name.

How do you know whether NPTL is being used?

- If the kernel level is higher than 2.4.19.
- NPTL was part of glibc Version 2.3, so check using:

```
rpm -qa | grep glibc
```

CONCLUSION

To all intents and purposes, MQ on the Linux platform looks and feels very similar to MQ on the other Unix platforms.

The recommendation is to start small with a proof-of-concept and ensure all hardware and software levels are compatible.

Ruud van Zundert (ruudvz@btclick.com)
Independent Consultant (UK)

Migrating from WebSphere MQ Integrator Broker Version 2.1 to WebSphere Business Integration Message Broker Version 5.0

This article describes migration strategies for message flows and message sets from WebSphere MQ Integrator Broker

(hereafter called WebSphere MQ IB) Version 2.1 to WebSphere Business Integration Message Broker (hereafter called WebSphere BI MB) Version 5.0. It assumes a basic understanding of the message broker and differences between both product versions. When comparing these migration strategies the article discusses advantages and disadvantages between different methods.

INTRODUCTION

Last year, IBM made available WebSphere BI MB Version 5.0, the successor product to WebSphere MQ IB Version 2.1. In the meantime, end of service for WebSphere MQ IB has also been announced for September 2005. Therefore it is time to think about migrating existing brokers that are processing production messages to the new version. Some possible ways to migrate a broker domain have been evaluated for WebSphere Business Integration for Financial Networks (hereafter called WebSphere BI for FN) on z/OS. Also considered are the implications these choices have, for example, regarding necessary changes required for the migration, the time needed to perform the migration of the message flows, or the fall back capabilities in case there are problems in the migration process. Note: for details about WebSphere BI for FN see http://www.ibm.com/software/integration/wbifn.

WebSphere BI for FN is an integration platform that consists of a base product and network-specific extensions. The base product is an infrastructure that eases the delivery of products on top of WebSphere MQ IB. It provides common services, for example:

- Customization – ways to adapt message flows and their resources, for example WebSphere MQ messages queues and database tables, to different run-time environments.
- Configuration – ways to dynamically influence the processing of message flows.

- Security – ways for message instance-based security.
- Predefined nodes – for commonly-needed functions for exploiting message flows, for example auditing, message warehouse, monitoring, and timer functions.

WebSphere BI for FN extensions provide value-added access to different financial networks, for example the Extension for SWIFTNet (ESN) allows customer applications access to the Secure IP Network provided by SWIFT (see http://www.swift.com for details about SWIFT). These extensions deliver message flows that invoke base product functions. An extension can be delivered by IBM or Independent Software Vendors, or any customer can develop their own extension.

The new version of the broker product not only offers new functionality, it also introduces significant changes to some components and procedures. Exploiting new functionality is usually not involved while migrating to the new broker version. This should be done after successful migration.

Most of the changes between the two broker versions are related to the replacement of the WebSphere MQ IB Control Center with the WebSphere BI MB Message Brokers Toolkit for WebSphere Studio (workbench for short). Coming with this change is the move of the repository for all development artefacts, for example message flows and message sets, from the central Configuration Manager to de-central tooling workstations. Nevertheless, exploiting the Eclipse-based tooling capabilities, a source code control system can be used to revert back to a central location for version control of the artefacts. (For details about Eclipse see http://www.eclipse.org.)

There are also changes to the development artefacts themselves. For example, a message flow is represented as one XML file in WebSphere MQ IB while in WebSphere BI MB the message flow can consist of different files, eg the message flow itself, the ESQL for compute, filter, and database nodes, mappings, and similar. Coming with these changes is some

new functionality, and also some restrictions and changes that are incompatible with WebSphere MQ IB nodes.

In contrast to the changes to the tooling and the Configuration Manager in the new product, the broker is, apart from additional functionality, very similar to the previous broker. All plug-ins developed for WebSphere MQ IB still work with the new version. On z/OS, recompiling them with XPLINK options is recommended for performance reasons. For details about the differences between both products have a look at the WebSphere BI MB documentation.

To discuss the migration strategies, it is assumed that there is a broker running that is processing production messages on z/OS. This broker should be migrated with the same functionality as before. The Version 2.1 broker is already running using prerequisite software versions that are included in the list of supported software for WebSphere BI MB. If necessary, prerequisite software should be upgraded before the migration of the broker starts.

WEBSPHERE BI MB DEFINED MIGRATION STRATEGIES

As part of the WebSphere BI MB documentation, there is already a defined migration path for a broker domain. The migration path works as follows. First a user needs to prepare for the migration followed by the actual migration that switches to the new version.

The migration preparation involves recording the topology and assignments. The topology includes the names of all message brokers that need to be migrated and the name of every execution group for each broker. The assignments include the names of all message flows in each execution group and all assigned message sets for each broker. For every message flow assignment it is required to record the message flow properties. This is the information about whether the flow is transactional, the commit count, the commit interval, and the additional instances parameter. All the information

has to be recorded manually, ie no tools are provided by WebSphere BI MB to support a user in this task. After all the information is written down, eg on a sheet of paper, the message flows and message sets should be locked on a Control Center workspace and exported to a file.

When this is done, the actual migration starts. The first step in the migration process is that all Control Centers, the Configuration Manager, and all brokers should be stopped. It has to be ensured that the Configuration Manager and the broker are not deleted because this would remove information that is used later.

The next steps are to uninstall the WebSphere MQ IB product and install WebSphere BI MB. The reason for uninstalling WebSphere MQ IB is that on most platforms the WebSphere BI MB product cannot be installed on the same operating system instance that the WebSphere MQ IB product is installed on. After the installation of the new product, the Configuration Manager and the broker should be re-created with the new version. They need to use the same resources, ie the same database instances and the same WebSphere MQ queue managers. The Configuration Manager is recreated by migrating the Configuration Manager database. This can be done using the WebSphere BI MB delivered program **mqsicreatetables** command. After this is done, the Configuration Manager can be started and all assignment, topology information, and topic data should be available again. To be able to work with the new Configuration Manager and later on with the brokers, a connection to the Configuration Manager must be defined. Be aware that with WebSphere BI MB the Configuration Manager no longer holds the repository for the development artefact message flows and message sets, and therefore these databases are no longer needed. This database can therefore be deleted once the migration has completed successfully.

To get a configuration similar to the WebSphere MQ IB broker domain, the message flows and message sets must be

migrated. This is supported by WebSphere BI MB migration programs **mqsimigratemsgflows** and **mqsimigratemsgsets**. These programs take the WebSphere MQ IB development artefacts and generate files out of them that can be imported into the new workbench.

In parallel with the Configuration Manager, the new broker can be set up. This is done by defining the broker as if a new broker were being created. The definition should be done using the same WebSphere MQ queue manager and the same database with the same database subsystem, schema name, and any other defined attributes like the lil-path (where to find the customer plug-ins). In this process the broker queues are updated as required by the new version.

When ready with the definition, the broker can be started. In the operations view, the broker should now become active. The broker still shows all the execution groups and message flows, but the topology and the assignments need to be recreated.

Once the message flows and the message sets are migrated and imported, the next steps are performed in the workbench. Here the domain should be recreated with the same configuration as recorded in the first step. This process starts with deleting all definitions of the broker – message flow assignments, message set assignments, and execution groups.

Afterwards, the brokers define the execution groups that were deleted in the previous step. Then the message flows and message sets can be compiled. The message flows need to be assigned the message flow properties as recorded in the first step, and these development artefacts can be deployed to the execution groups. There are significant differences in this process compared with WebSphere MQ IB; for example, message sets must now be assigned to execution groups rather than to a broker. To be sure that the message flows continue to work with the message sets, it is safer to assign the message sets to all execution groups of the broker where

the message set was assigned before. Once you are certain which flows require which message sets, the number of message set assignments can be reduced.

Another difference is the deployment process itself. Instead of the resources being directly assigned to the broker, they first have to be compiled into so-called broker archive files (BAR files). In these BAR files attributes for the message flows, like the additional instances parameters, can be set. Afterwards the BAR files can be deployed to an execution group. For details of how to work with the workbench, for example how to import and compile message flows and message sets, see the appropriate WebSphere BI MB documentation.

Once all the above operations are successful, verification can be begun of whether the message flows processing still gets the same results as before. For further details about the proposed WebSphere BI MB migration strategy, please read the appropriate parts of the WebSphere BI MB documentation.

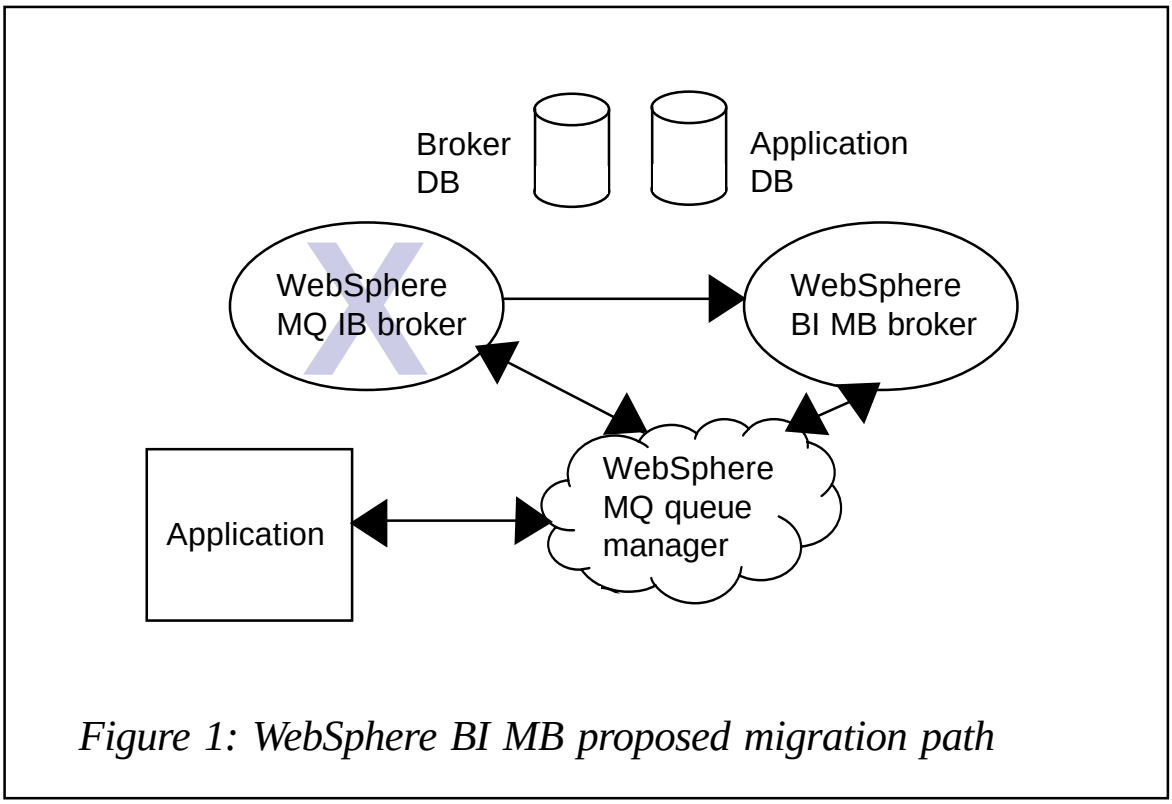

Figure 1: WebSphere BI MB proposed migration path

ADVANTAGES AND DISADVANTAGES OF THIS METHOD

The WebSphere BI MB migration procedure has the advantage that it is supported and documented by the broker product. It also reuses the resources of the old installation as shown in Figure 1 and therefore the need for additional resources is minimal. Another advantage of this strategy is that all applications working with the brokers by sending and receiving messages to and from the message flows are unaware of this product upgrade.

The main disadvantage of this migration method results from the fact that it involves large changes. In the migration documentation there are no hints on what do to if the migration to WebSphere BI MB fails. There are a lot of reasons why a migration could fail, for example:

- *Problems with the subsystems* – the broker (and also Configuration Manager and the workbench) has other software prerequisites. The new installation may not work with the installed WebSphere MQ or database manager, or they require additional fixes.
- *Message flow and message set migration problems* – the migration programs delivered with WebSphere BI MB have a set of restrictions. The migration programs report on the results of each resource they migrate.
- *Missing artefact* – because the message flows and message sets to migrate have to be started manually, some of them may be overlooked and therefore will be missing. Another reason why some message flows may be missing or not assigned according to the old configuration is that in the first step the record could be incomplete or contain typos. Since the old configuration is no longer available, it cannot be used as a reference for checking the assignments.

Usually, if the system is used for production and the migration is not successful, a user would like to switch back to their ‘old’ system running WebSphere MQ IB. But, because of the large

set of changes, this is not easily possible. The configuration of the complete machine could be backed up before starting the migration, as recommended in the documentation. In the case of problems, the old system could then be restored from this back-up. But if the machine is used for multiple purposes, this could disturb other users. So from a fallback perspective, this approach could result in problems.

Another drawback with this approach is the time needed to perform the migration. There are many steps that need to run in sequence. At least for the period from stopping the complete domain until after verifying whether the new system is working properly, no production work can be processed. Many of the steps involve manual interactions that are inherently slow. The whole migration can take many hours or days. This problem could be reduced if the migration is first tried on a test

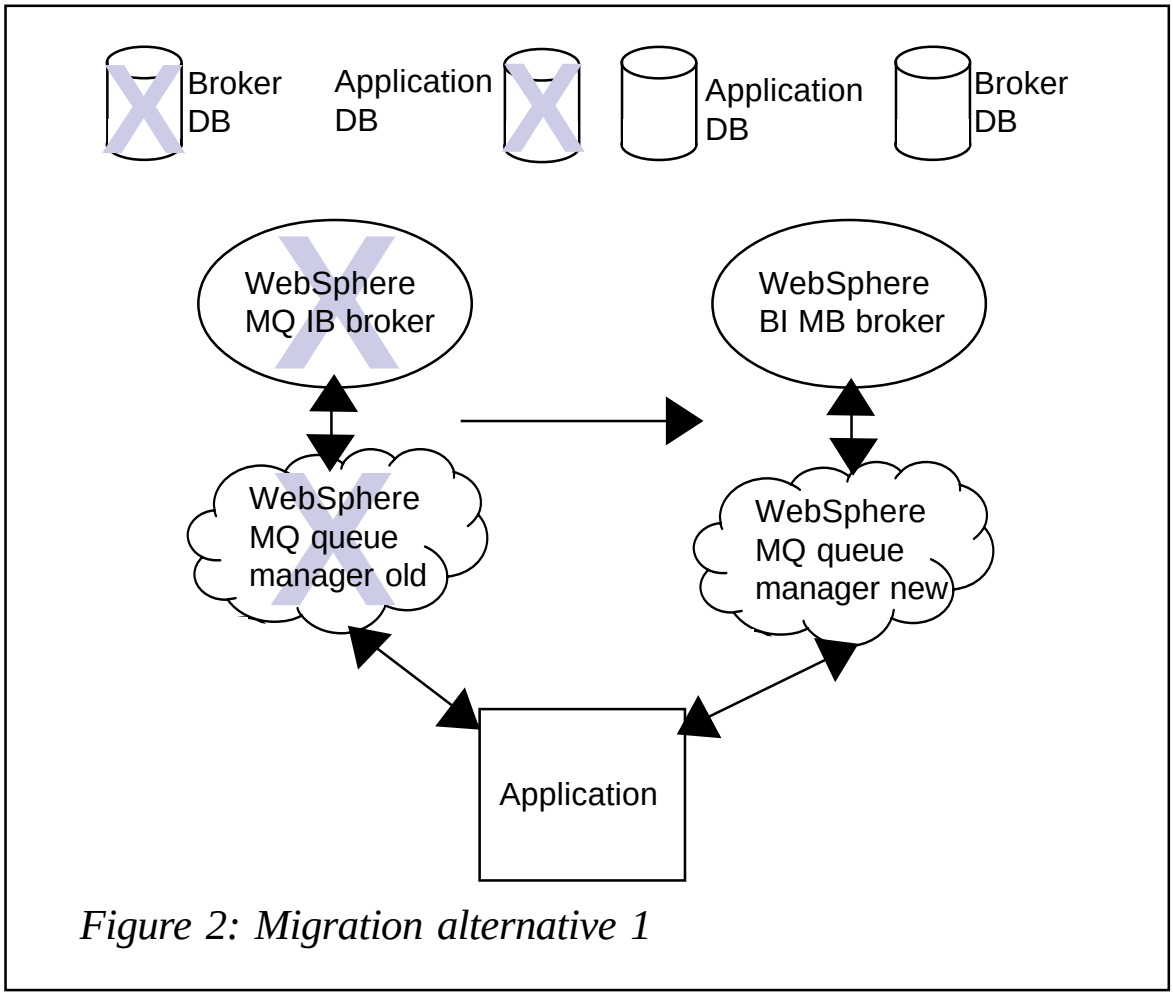

Figure 2: Migration alternative 1

system, but, because there are usually only a small number of test systems and the fallback is difficult, not many practice runs are possible.

There's another possible way to reduce the time needed to perform the migration. The message flows could be migrated on a separate workstation where only the WebSphere BI MB tooling is installed. This could be done as far as preparing the BAR files (including all message flows attributes). A BAR file could be prepared for each execution group for each broker.

The whole migration process relies on many manual processes, for example recording the important topology and assignment information and recreating these from scratch. After the WebSphere MQ IB product has been uninstalled there is no way to control what has been recorded or to get information about whether something is missing.

A FIRST ALTERNATIVE

When looking at the main problems of the WebSphere BI MB migration procedure, namely the time needed to switch to the new version and the huge effort to fall back to the WebSphere MQ IB version if the migration is not working as expected, an obvious alternative is to build the new broker in parallel on a new system as shown in Figure 2. The Version 5.0 product could be installed and the brokers, Configuration Manager, and tooling could be created and configured. The message flows could be taken from the WebSphere MQ IB system, migrated, compiled, enriched with the message flow properties, and deployed to the new brokers as described in the WebSphere BI MB defined approach. During the complete process, all the definitions of the 'old' system are still available and they can be used to control what is deployed on the new system. This way the new system could be completely tested before being put into production.

In this alternative, it is not strictly required to upgrade the prerequisite software to the levels supported by WebSphere

BI MB. If a broker is running different sets of message flows, another advantage of this approach is that each set could be migrated when necessary. It is not necessary to migrate all message flows and message sets at once.

With this alternative, switching to the new broker would mean changing all systems and applications around the broker to a new environment. The amount of effort required for this depends on the number of such applications. Depending on the kind of processing in the message flow additional migrations could be necessary – for example to guarantee that the messages can be processed in the new environment, all the state and processing data from the old system may need to be moved to the new system. This could involve just some information in a database, but it could also mean that the content of some WebSphere MQ queues needs to be moved. The latter could be avoided if there is a possibility of stopping sending new messages from the applications before the migration starts. This gives the broker a chance to process all outstanding messages. Verify that all important message queues are empty, eg using the WebSphere MQ command **DIS QL(*) CURDEPTH**. The current depth of the queues of interest should be 0.

Changing applications to a new environment could mean for a simple case that the application just needs to connect to the new queue manager. In more complicated cases, this could include moving databases and table data or WebSphere MQ messages in the same way as for moving the broker's processing data, as described above.

If switching to the new environment is not successful for any reason, the fallback is to reactivate the old environment containing the WebSphere MQ IB broker. But, depending on when the problem is detected, data may need to be migrated back to the old systems. If now the versions of the prerequisite software are different, this may not work. For this reason it would be very helpful also to upgrade the software levels on the broker Version 2.1 systems before starting the migration.

For the case of WebSphere BI for FN ESN, there is an additional difficulty in changing the applications to the new environment. The problem involves the application software for accessing the SWIFT network. This software, called SWIFT Alliance Gateway (SAG), would also be needed to test the new broker. Each SAG has a separate identity that is reflected in the processing data for the broker. Since this information is needed for the old system for production, the existing SAG could hardly be used for testing the new broker environment. Ordering a second one takes a lot of time and is relatively expensive because the second SAG should have the same capacity as the existing SAG in order to take over the production workload once the migration is complete.

Besides the extra SAG, a lot of other additional resources are also required, at least for the migration period. These resources are, for example, the system where the new product is installed, the databases, queue managers, disk space, and similar. Taking into account also all the difficulties mentioned in the previous sections, this approach was not deemed acceptable as a migration method for WebSphere BI for FN.

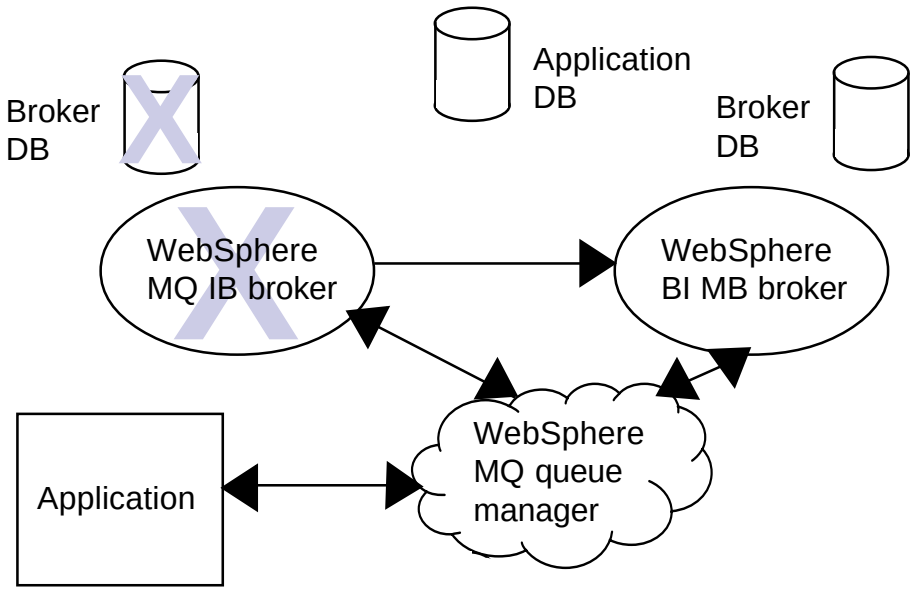

Figure 3: Migration alternative 2

A SECOND ALTERNATIVE

The WebSphere BI MB defined migration approach and the alternative described in the previous section have some significant drawbacks. To overcome these drawbacks a new alternative as shown in Figure 3 was developed. This is in fact a combination of both approaches and is based on the fact that on z/OS both versions of the broker can be installed and brokers of different versions can be defined and active at the same time. This alternative works by exporting all message flows as in the WebSphere BI MB approach. Instead of removing the Configuration Manager and the broker, a second system is used to install the WebSphere BI MB product with the workbench and to create the Configuration Manager. On this second system the message flows and message sets can be migrated. For every execution group of each broker in the old system, a BAR file is created containing all message flows and message sets corresponding to the assignments of the old system. The queue manager of the new Configuration Manager should be connected to the queue managers of all brokers that need to be migrated. This is all preparation work and doesn't disturb the production system. Other preparation work involves preparing the broker on z/OS. This includes customizing the broker until it can be started. It must be done with parameters similar to the WebSphere MQ IB broker it should replace. This means it should have, for example, the same paths for product extensions (plug-ins) and it should use the same WebSphere MQ queue manager. A significant difference should be that the database schema name used for broker internal tables needs to be changed.

After these preparations, switching to the new version can take place in a relatively short timeframe. For this switch, the WebSphere MQ IB broker should be stopped. It must be ensured that there is no outstanding deployment to the broker. This can be verified by checking all broker queues named SYSTEM.BROKER.*. These should all have a current queue depth of 0. Now the WebSphere BI MB broker can be started. Using the new tooling, the broker can be defined and the

necessary execution groups added. Afterwards, the prepared BAR files should be deployed to the corresponding execution groups. If this is successful, the new message flows are ready to be tested to see whether they still return the same results as before. When successful, the new message flows are ready for production.

Compared with the previous alternatives, the time during which production messages can be processed is relatively short. In the case of the migration not being successful, a fallback is simple. Depending on when the problem occurred, the following action brings back the WebSphere MQ IB broker:

- Stop the Version 5.0 broker.
- Ensure that the broker queues are empty.
- Start the Version 2.1 broker.

Afterwards you should test that the message flows are processing messages correctly to be sure that the flows still work. If this is successful, the broker Version 2.1 can be used to process production messages while off-line the migration problem can be analysed and corrected before a new attempt to migrate the broker can be started.

Once the migration is successful and you are sure that no fallback is needed, some clean-up action should be performed. This includes throwing away the broker Version 2.1 and dropping its database tables. The broker should not normally be de-configured. This process leaves an unusable broker definition in the Configuration Manager of the WebSphere MQ IB domain. To avoid any unwanted configuration messages for the new broker, the connection to the broker's queue manager should be dropped. A better approach, if all brokers have been successfully migrated, is just to delete the Version 2.1 Configuration Manager. Then the system can be freed and reused for other software.

The main advantage of this migration approach is that the migration happens within the environment in which the old

broker runs and no external systems and applications need to be changed. The time taken to actually switch to the new version can be a few hours, but if it is first tried in some test environments, several minutes. A fallback to the old environment is also possible without too much effort. No data or messages need to be moved between systems.

But the approach has also some disadvantages. Compared with the WebSphere BI MB proposed migration path, an additional system with the WebSphere BI MB product for the new Configuration Manager and the workbench is needed. If not processed carefully, a broker could get configuration messages from the wrong Configuration Manager. The biggest disadvantage is that the approach works only on z/OS.

CONCLUSION

Migrating an application in a production environment that involves message flows and message sets from WebSphere MQ IB Version 2.1 to WebSphere BI MB Version 5.0 is difficult. The main reason for this is the significant changes introduced with the new message broker product. There are different approaches to performing the migration. Every approach has advantages and disadvantages, and the decision about which approach to take has to be made carefully. The amount of time that the production system is unavailable for processing messages, the effort that has to be spent to fall back to the old system if the migration is not successful, and the additional resources that are required to perform the migration must all be taken into account. Basing the choice on these criteria and the concrete applications and message flows should allow the best approach to be selected.

Michael Groetzner
IBM (Germany)

Rotating application log files in WebSphere Application Server Version 4.1

INTRODUCTION

During a recent assignment, while migrating an application into a production environment, my client raised a concern about the size of the application log files. During the quality assurance phase of the application life-cycle, both saturation and stress testing were completed. These tests were designed to simulate a limited number of users. However, even with the reduced user population, the application log files quickly grew to between 100 and 400MB.

The resulting problem was that the log files were too large to support the timely resolution of application problems. As the size of the log files increased, two issues came to the fore:

1 The hardware that supported WebSphere and the application code were isolated from the internal LAN, and moving files of more than 3–5MB in size became a frustrating, time-consuming process.

2 When the support desk personnel actually had to review the application log files, the sheer volume of data to be processed extended the time that an analyst required to isolate the relevant information

Prior to releasing the application into production, it was decided that a process was required that would facilitate the switching and archiving of the application log files in an automated fashion. In order to understand the alternatives considered, it is first necessary to understand the layout of the application infrastructure.

THE ENVIRONMENT

The corporate infrastructure for this client involved the following components:

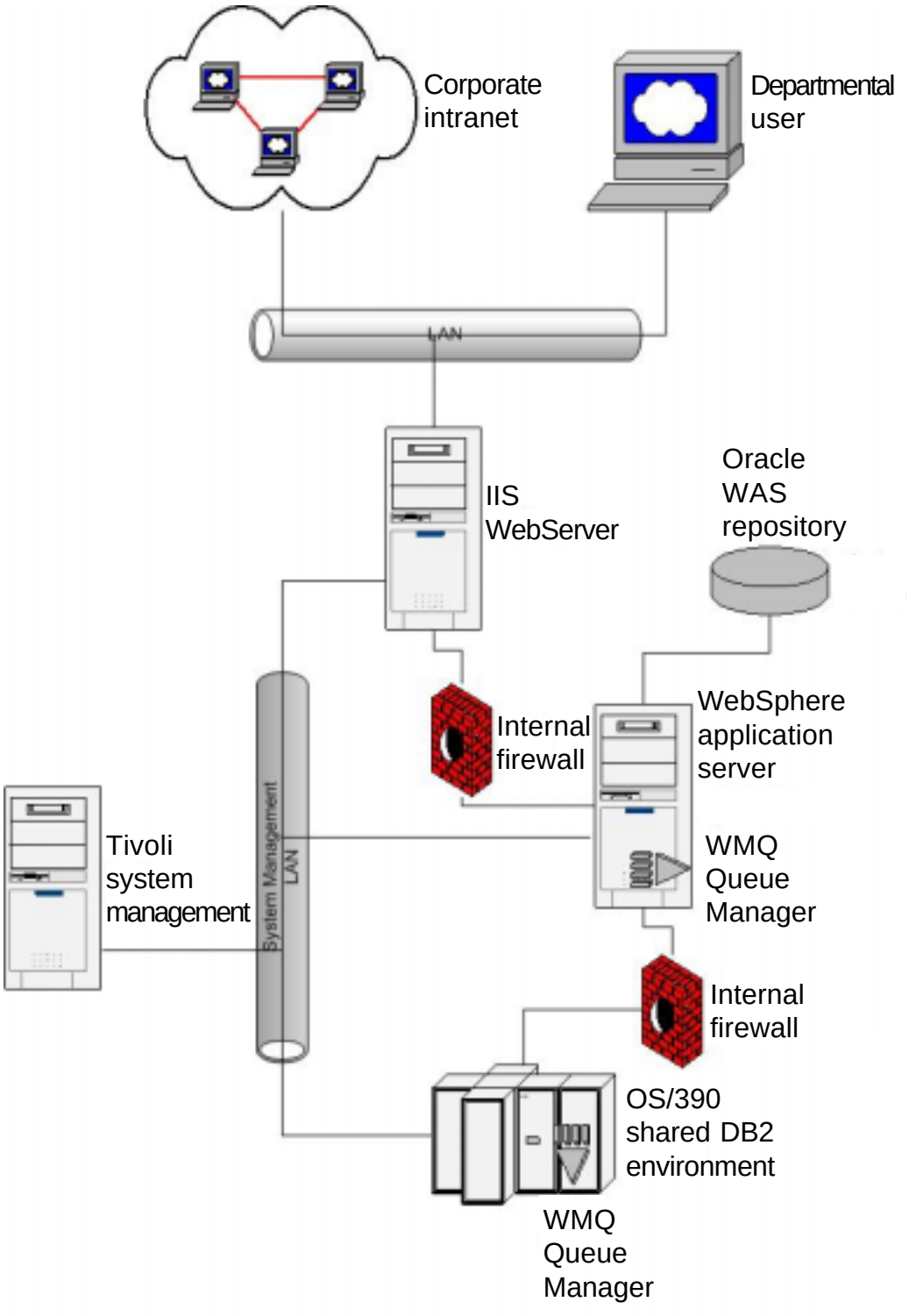

Figure 1: The enterprise infrastructure

- IIS 5.1
- WAS 4.1
- Oracle 8
- WebSphere MQSeries 5.2
- DB2 OS/390.

THE ENTERPRISE INFRASTRUCTURE

The enterprise infrastructure is illustrated in Figure 1.

IIS

The Web server chosen for the client front end was IIS. In this case, IIS was designed to act as a gate from the corporate intranet to allow secured departmental access to applications. This particular application was designed to be a combination of static HTML pages backed with ASP JavaScript. EAI connectors written in Java deployed Enterprise Java Beans for access to the legacy DB2 environment.

Because of the nature of the legacy systems involved, and the requirements of application integration of both legacy and departmental programs, WebSphere Application Server was chosen. Following the deployment of the business application into the WebSphere instance, the IIS plug-in was generated within WebSphere and installed on the IIS server instance as an ISAPI filter.

WAS

WebSphere Application Server was chosen as the server instance, based on Enterprise IT standards. In cases where applications required more complex processing than could easily be supported by either static HTML or embedded scripting, Java application code in the form of both applets and servlets was created to support the design requirements.

Since many of the business designs required access to the

data generated by existing legacy systems, Enterprise Java Beans were created to provide a bridge between the client applications and the existing database environments.

Oracle

In order to support the WAS processing environment, Oracle on AIX was used to provide the WAS repository facilities. By installing the repository on a Unix server platform, the enterprise could take advantage of enhanced disk space and improved disk management, as well as increased operating system support for the Oracle server.

Since the majority of the Oracle repository access required by WebSphere is initiated when the WAS server is first started, the amount of traffic generated across the network link was minimal, and therefore did not impede application or subsystem performance.

WebSphere MQ

In order to integrate the newly designed front end with the existing legacy application, WebSphere MQ was chosen as the middleware layer. In addition to direct table look-up processing by components of the new user interface, extraction of larger volumes of legacy data to satisfy business user requests was accomplished by executing existing mainframe application code.

The database result set was generated by existing mainframe applications. Their target was changed from a mainframe terminal to a message buffer. This could then be accessed by an MQSeries queue manager application, which was designed to move data into the MQSeries middleware environment. This made it possible to leverage the existing applications into a much larger user community without the impact of redesigning and rewriting large amounts of application code.

DB2 OS/390

The majority of the existing legacy data, as well as the existing

legacy application code, were part of a Walker application installation. Since the Walker installation was DB2 based, EJBs were created to provide direct access to the existing data store and to authorize access to sensitive corporate information.

Since the Walker application itself includes a robust reporting capability, as well as providing options for application customization based on enterprise standards and business requirements, re-using the CICS-based Walker transactions to access the stored DB2 data saved significant amounts of application development time and resource.

POSSIBLE SOLUTIONS

Whenever an application server is started, or restarted, within WebSphere, it is possible to create a new copy of the application

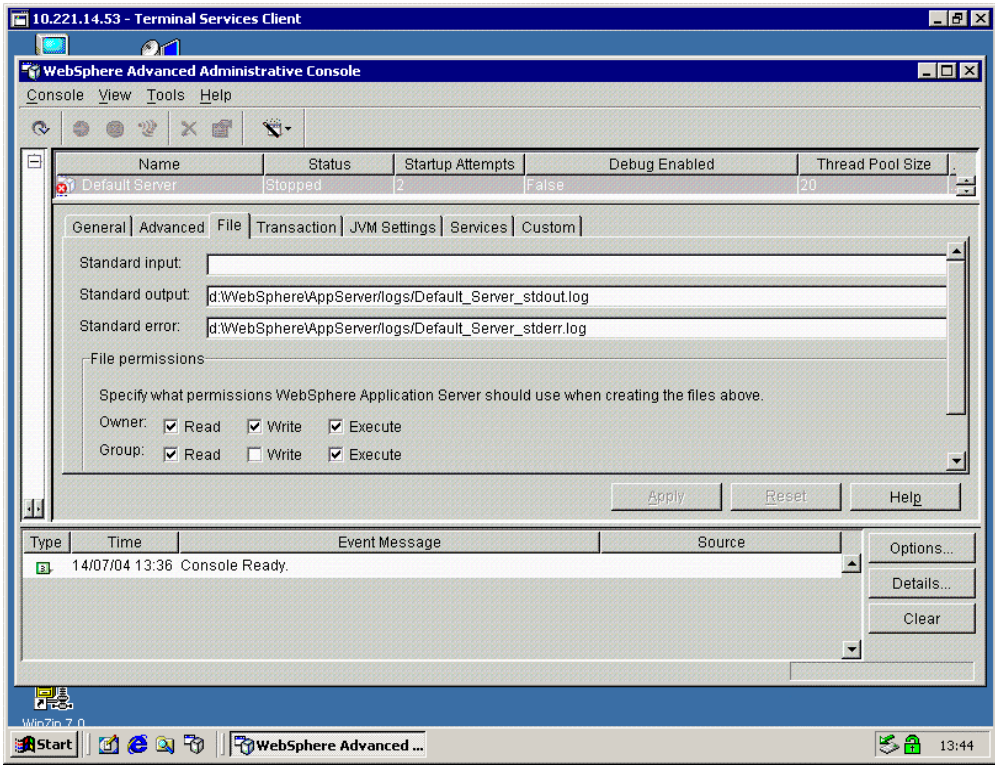

Figure 2: Fields for for stdout and stderr

standard output and standard error logs. These files can be identified within the WAS Application Server instance by using the WebSphere Advanced Administrative Console, by selecting the chain **WebSphere Administrative Domain,** then **Server Groups,** then the application server group definition. On the properties panel select the **File** tab and you will see fields for various file entries, two of which are for stdout and stderr, as shown in Figure 2.

If the physical datasets, which are referenced above, do not exist, the WAS application server instance will create new copies of the logs and start writing to them.

This is a standard feature of WebSphere, and therefore, the initial decision by the client business analysts was to rename and archive the application log files each time the application server was restarted. This would allow the resulting log files to be broken into time-based segments, which could then be published for access by the Help Desk staff in order to resolve reported application problems.

The enterprise management framework used by the client was Tivoli. A suggestion was put forward that a Tivoli script should be created, which would manage timed maintenance of the application including the shutdown, archival of log files, and restart of the application server instance.

The initial deployment of the application into a limited production status allowed the client community to begin working while the Tivoli solution was implemented. Using the various Tivoli infrastructure agents, measurements were made of various application performance metrics, including the growth of the log files. These numbers were then extrapolated into performance projections of resource consumption once the application reached full deployment.

While the numbers from the performance projections created some concern, it was the actual file figures experienced during the limited deployment that confirmed that the Tivoli solution would not be viable. With a limited client population

of between 100 and 200 users, the application log files grew at an approximate rate of 2MB per business day. Given that the actual application client population on full deployment would be between 7,000 and 9,000 users, the projected growth of the log files would be as much as 140MB per day.

As was previously mentioned, the size of the log files that was most effective was around 5MB. Following through with the maths, if the log archival process was to rely on starting and stopping the application server in order to close, archive, and recreate the log files, keeping them to a manageable size, it would be necessary to stop and restart the application 28 times a day.

So, the Tivoli based solution was rejected and a new effort was introduced to find a solution to the log file problem.

IF IBM DIDN'T WANT LOG DATA TO BE USED, THEY WOULDN'T MAKE SO MUCH OF IT…

If there are two things that IBM understands, it's that enterprise software and subsystems will be used by large client populations and that large customers will make some very strange decisions when they 'design' their enterprise architecture.

Large client populations generate large quantities of log data, which must be accessible to the technical resources tasked with supporting enterprise applications. After a bit of searching and networking, a process was found that is built into WebSphere Application Server, which can limit the size of the application log files while automatically supporting log file archiving.

When a WebSphere Application Server instance runs, it has logging processes that are controlled by the *logging.properties* file located in the directory *C:\WebSphere\AppServer\properties* (obviously this assumes that you have installed WebSphere on your C drive. If not, then substitute the drive where you installed WAS for the C specification).

The *logging.properties* file, which is delivered by IBM as the default for a new WAS instance, is similar to the following:

```
#---------------------------------------------------------------------
# This file contains Ras properties that are used to enable or disable
# various services or to set service levels.
#
# The default state for each property is indicated, as well as the valid
# options for that property. The usual rules for java properties files
# apply.
# The expected format for an entry is key=value (with no separating
# white space).
#---------------------------------------------------------------------

#---------------------------------------------------------------------
# Activity Log Properties
#
# WARNING Before changing any of the Activity Log properties, all
# servers on the
# physical node, including AdminServers must be stopped.
#
# com.ibm.ws.ras.AcvitityLogEnabled : A property used to determine
# whether or not the
# servers on this node will write Ras events to the activity log or not.
# valid values
# are true and false, with true the default. Turning off this logging
# may have serious
# serviceability impacts.
#
# com.ibm.ws.ras.ActivityLogSize : Size of the activity log in
# kilobytes. The default
# value is 1Ø24 which yields a log size of 1 megabyte. See the Problem
# Determination
# guide for guidelines on setting the size of this log.
#---------------------------------------------------------------------

com.ibm.ws.ras.ActivityLogEnabled=true
com.ibm.ws.ras.ActivityLogSize=1Ø24

#---------------------------------------------------------------------
# Correlation Id property
#
# The following property is retrieved once for the life of a process at
# startup. It
# can be changed in this file at any time, but in order for the change
# to take effect
# on started processes, they must be restarted.
#
# com.ibm.ws.ras.UnitOfWork : A property that determines whether
```

```
# messages and
# diagnostic trace entries will be flagged with a correlator. The intent
# of the id is
# to allow correlation of events that occur in different processes as
# being related to
# or caused by a single client request. Valid values are true and false.
# The default
# value is true.
#-----------------------------------------------------------------------

com.ibm.ws.ras.UnitOfWork=true

#-----------------------------------------------------------------------
# Serious Event Forwarding property
#
# The following property is retrieved once for the life of a process at
# startup. It
# can be changed in this file at any time, but in order for the change
# to take effect
# on started processes, they must be restarted.
#
# com.ibm.ws.ras.SeriousEventEnable : A property that determines whether
# Ras messages
# are forwarded to the central repository (Admin Database). Valid values
# are true and
# false.
#
#    The default value is true.
#-----------------------------------------------------------------------

com.ibm.ws.ras.SeriousEventEnable=true

#-----------------------------------------------------------------------
# Ras Message Filter level property
#
# The following property is retrieved once for the life of a process at
# startup. It
# can be changed in this file at any time, but in order for the change
# to take effect
# on started processes, they must be restarted.
#
# com.ibm.ws.ras.MessageFilterLevel : A property that determines which
levels of Ras
#    messages are logged and which are not. Valid values are :
#     - error : only messages of severity error are logged.
#     - warning : only messages of severity error or warning are logged.
#     - audit : all messages are logged.
#    The default value is audit.
#-----------------------------------------------------------------------
```

```
com.ibm.ws.ras.MessageFilterLevel=audit

#-------------------------------------------------------------------
# Trace Format property
#
# The following property can be changed at any time. Whether or not the
# effect is
# immediate or delayed until the next process restart is dependent upon
# the
# facility.
#
# com.ibm.ws.ras.TraceFormat : A property that determines how trace
# output is
# formatted.
#
#    Valid values are :
#    - basic : generates the sparse legacy trace format.
#    - advanced : a more verbose style patterned after the basic format.
#
#    Recommended
#       - in most cases.
#       - loganalyzer : generate a trace output that can be parsed by
#         the Log Analyzer
#         tool. Generates larger trace files than the other options.
#         Recommended when
#         doing cross-process trace.
#
#    The default value is basic
#-------------------------------------------------------------------

com.ibm.ws.ras.TraceFormat=basic
```

This file can be used as a template to control logging at the application server level, with a suitable modification of the parameters. In particular, we want to provide dataset names and sizes that allow us to segment the standard out and standard error output streams into pieces that can be easily archived and handled by the Help Desk or application support teams.

Consider the following changes to the *logging.properties* file, listed above. Where sections remain the same it has been indicated within the following listing. Where changes or additions have been made, the modified sections have been listed in their entirety:

```
#-------------------------------------------------------------------
# This file contains Ras properties that are used to enable or disable
```

```
# various services or to set service levels.

ooooo
ooooo
ooooo

#------------------------------------------------------------------------
# Activity Log Properties
#
ooooo
ooooo
ooooo
#------------------------------------------------------------------------
# Correlation Id property
#
ooooo
ooooo
ooooo

#------------------------------------------------------------------------
# Serious Event Forwarding property
ooooo
ooooo
ooooo

#------------------------------------------------------------------------
# Ras Message Filter level property
#
# The following property is retrieved once for the life of a process at
# startup. It
# can be changed in this file at any time, but in order for the change
# to take effect
# on started processes, they must be restarted.
#
# com.ibm.ws.ras.MessageFilterLevel : A property that determines which
# levels of Ras
#    messages are logged and which are not. Valid values are :
#     - error : only messages of severity error are logged.
#     - warning : only messages of severity error or warning are logged.
#     - audit : all messages are logged.
#    The default value is audit.
#------------------------------------------------------------------------

com.ibm.ws.ras.MessageFilterLevel=warning

#------------------------------------------------------------------------
# Trace Format property
#
ooooo
ooooo
```

```
ooooo

#------------------------------------------------------------------------
# Standard Output Application Log File
#
#   The four properties for the System.out log are listed below:
#
#   com.ibm.ws.ras.SystemOutLogEnable
#   Set this property to true, to enable this log. The default value is
#   false.
#
#   com.ibm.ws.ras.SystemOutLogName
#   Set this property to a unique, fully qualified file name. The
#   recommended value
#   is <WASHOME>/logs/<ServerName>_SystemOut.log. Ensure the specified
#   file name is
#   in a valid format for the platform. If a non-default directory is
#   specified,
#   create the directory before you start the server process. In
#   addition,
#   WebSphere Application Server must have write access to this
#   directory.
#
#   com.ibm.ws.ras.SystemOutLogRollover
#   Specify the maximum size of the log file, in megabytes. The default
#   is 1 (one
#   megabyte). This number must be positive.
#
#   com.ibm.ws.ras.SystemOutLogBackups
#   Specify the maximum number of archive files to retain. The default
#   is 1. This
#   number must be positive.
#
#------------------------------------------------------------------------

com.ibm.ws.ras.SystemOutLogEnable=true
com.ibm.ws.ras.SystemOutLogName=C:\WebSphere\AppServer\logs\<appsrvrnm>_Stdout.log
com.ibm.ws.ras.SystemOutLogRollover=4
com.ibm.ws.ras.SystemOutLogBackups=30

#------------------------------------------------------------------------
# Standard Error Application Log File
#
#   The four properties for the System.out log are listed below:
#
#   com.ibm.ws.ras.SystemErrLogEnable
#    Set this property to true, to enable this log. The default value is
#    false.
#
#    com.ibm.ws.ras.SystemErrLogName
```

```
#    Set this property to a unique, fully qualified file name. The
#    recommended value
#    is <WASHOME>/logs/<ServerName>_SystemErr.log. Ensure the specified
#    file name is
#    in a valid format for the platform. If a non-default directory is
#    specified,
#    create the directory before you start the server process. In
#    addition,
#    WebSphere Application Server must have write access to this
#    directory.
#
#    com.ibm.ws.ras.SystemErrLogRollover
#    Specify the maximum size of the log file, in megabytes. The default
#    is 1 (one
#    megabyte). This number must be positive.
#
#    com.ibm.ws.ras.SystemErrLogBackups
#    Specify the maximum number of archive files to retain. The default
#    is 1. This
#    number must be positive.
#
#------------------------------------------------------------------------

com.ibm.ws.ras.SystemErrLogEnable=true
com.ibm.ws.ras.SystemErrLogName=C:\WebSphere\AppServer\logs\<appsrvrnm>_stdout.log
com.ibm.ws.ras.SystemErrLogRollover=1
com.ibm.ws.ras.SystemErrLogBackups=30
```

If you compare the listings of the two files closely, you will note that the value of the RAS Message Filter Level parameter (*com.ibm.ws.ras.MessageFilterLevel*) in the second list has been changed from the original default value of ‘audit’ to a more production compatible value of ‘warning’.

When applications and the WAS instance are first being tested, the overhead of having messages generated at the audit level can be justified. However, once clients begin using delivered applications in earnest, the amount of resource consumed and the volume of data generated by audit level message capture are hard to justify.

In addition to the parameter change, two new sections of the properties file have been added to format the application standard output messages and the application standard error messages. As indicated in the in-line documentation, there are four customizable parameters for each log file:

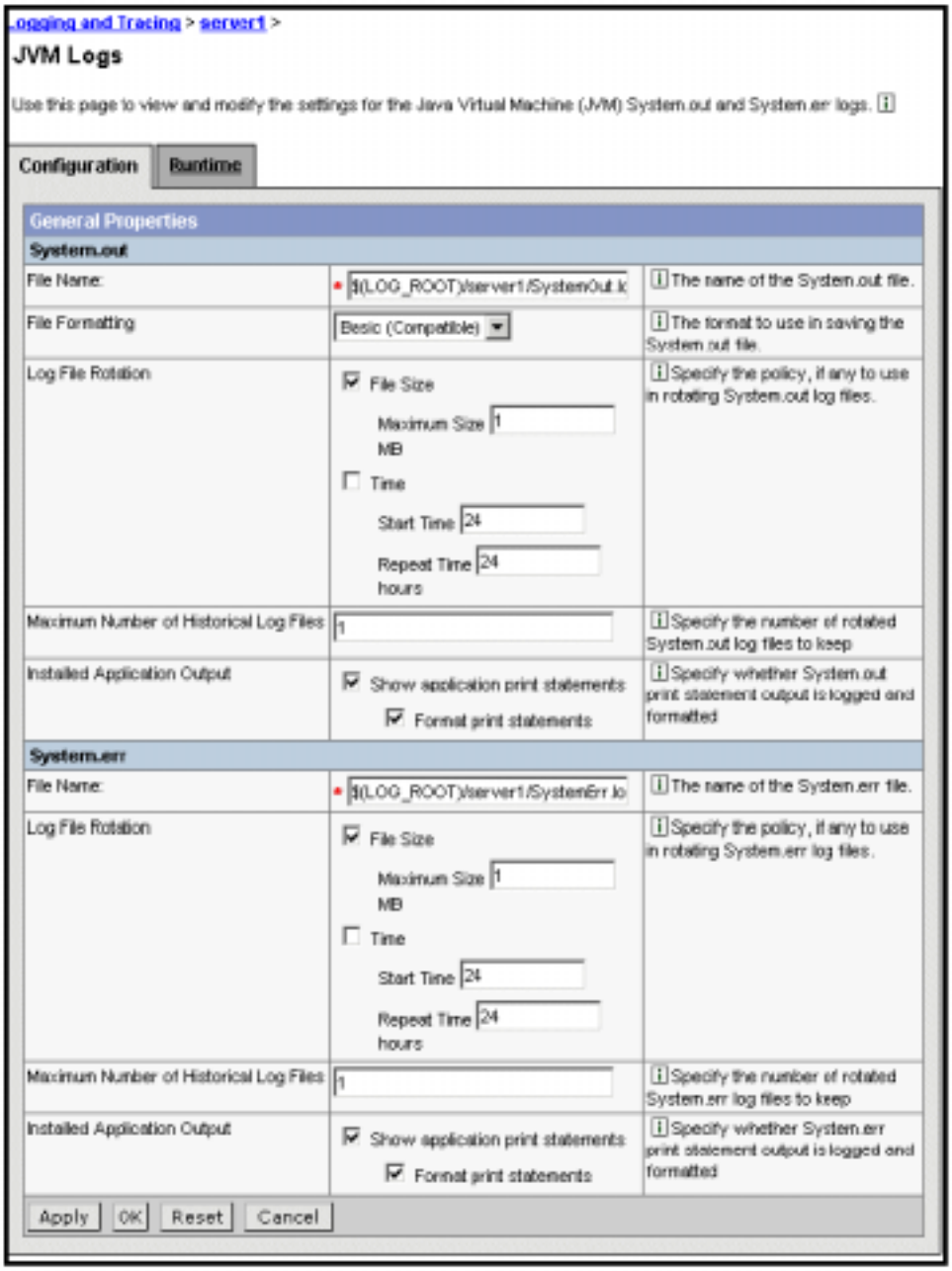

Figure 3: Log file rotation panel

- com.ibm.ws.ras.System%%%LogEnable
- com.ibm.ws.ras.System%%%LogName

- com.ibm.ws.ras.System%%%LogRollover
- com.ibm.ws.ras.System%%%LogBackups.

Note: the value of %%% is either **out** for standard output or **err** for standard error.

When the value of *com.ibm.ws.ras.System%%%LogEnable* is set to true, then logging of the standard output or error messages is enabled, and the application server instance within WAS will write diagnostic information to the files identified in the application server file properties page which was shown in Figure 2.

The dataset name indicated in the parameter *com.ibm.ws.ras.System%%%LogName* must match the value set in the application server properties page shown in Figure 2. It is recommended (as well as being common sense) that you include the name of the application or another identifier in the log file name in order to be able to differentiate log files by the application(s) that write to them.

The value of *com.ibm.ws.ras.System%%%LogRollover* is the target size of the log file in megabytes. When logging is active, and the size of the file reaches the limit specified, the WAS instance will close the application log, rename the current file, including a timestamp in the file name, and open a new log for output. In most systems the amount of error output is significantly less that the amount of standard output, so you may want to vary the size of the *com.ibm.ws.ras.SystemOutLogRollover* versus the *com.ibm.ws.ras.SystemErrLogRollover* parameters so that there is a closer synchronization of the switching and archiving between the two files.

DOES THE PROCESS SEEM COMPLICATED?

Well, of course it does seem complicated! If it didn't, then we wouldn't have jobs supporting WebSphere Application Server, and all of the applications that depend on the server for Web enablement. However, even IBM seems to have realized that

the method of implementing log file rotation outlined above is a bit on the hairy side.

So, in WebSphere Application Server V5.1, the process for specifying log file rotation has been simplified by creating a log file rotation panel, which allows users to specify their requirements in a clear and concise format – see Figure 3.

As you can see, in addition to simplifying the specification and start of log file rotation, an option is also presented that allows users to rotate files by time instead of size of the log file. With this improvement, it is now possible to keep the SystemOUT and SystemERR files in synchronization by time stamp.

CONCLUSION

Within a WebSphere Application Server installation, it is necessary to limit the size of application server log files. The need stems from both infrastructure requirements and the ergonomic impact of trying to analyse large quantities of log data in order to ascertain the conditions that existed within the instance at the time a problem occurred.

In order to create usable log files for applications that are installed in a WebSphere Application Server instance, it is necessary periodically to close the existing log files, archive those files for the support staff as well as for the performance analysts, and open new files. If you are using WAS V4.1, then by following the procedures outlined in this article, you can automatically create log files of a consistent size. This means that application support personnel can be more effective in resolving problems, since they will have less data to go through looking for incident documentation.

Of course, if you have the luxury of either upgrading your instance to WAS V5.1 or you have started with V5.1, the specification of log file rotation parameters is simplified. However, just because it is simpler doesn't mean that it is any less important to the process of resolving problems.

Aaron Cain
Independent Consultant
The Performance Edge Limited (UK)

Customer Evolutions has announced Release 2.0 of Enterprise Customer Profile (ECP) for the Java platform, which now supports additional platforms including J2EE 1.4-compliant application servers, Linux, and DB2.

The product serves as a single, unified, and integrated source of customer information across an organization. ECP comprises common business services connected to a database platform that provides data access and data synchronization capabilities.

ECP includes a Web services architecture that supports XML access over HTTP and WebSphere MQ. It has a flexible transport layer that supports legacy and third-party protocols and formats including WebSphere MQ.

For further information contact:
Customer Evolutions, 2009 N Lincoln Avenue, Chicago, IL 60614, USA.
Tel: (773) 509 6343.
URL: www.customerevolutions.com/solutions/solutionsCdi.php.

* * *

MQSoftware has announced Version 3.2 of Q Pasa!, its real-time middleware monitoring solution. The new version contains new capabilities designed to facilitate better information flow to fix problems faster, and manage user's enterprise infrastructure more efficiently and cost-effectively.

The company claims that the product provides a comprehensive solution for WebSphere MQ monitoring. New features include tabular views of connected WebSphere MQ applications providing insight into application resource usage, and extended cluster support to allow administration of WebSphere MQ clusters.

For further information contact:
MQSoftware, 1660 South Highway 100, Suite 400, Minneapolis, MN 55416, USA.
Tel: (952) 345 8720.
URL: www.mqsoftware.com/product/qpasa.jsp.

* * *

Netegrity has announced Version 6.0 of TransactionMinder

This version adds support for more Web services architectures. It can run as an agent natively on .NET, or in proxy mode on application servers like WebSphere and WebLogic.

Version 6.0 also supports the recently-ratified WS-Security 1.0 specification and now is capable of producing and consuming SOAP messages using three different security tokens: Username/Password digest, X.509 certificates and SAML tokens. It also supports WS-Security Encryption to encrypt and decrypt tokens and message elements. It can also protect Web services hosted on IBM WebSphere and BEA WebLogic platforms.

For further information contact:
Netegrity, 201 Jones Rd, Waltham, MA 02451, USA.
Tel: (781) 890 1700.
URL: www.netegrity.com/products/products.cfm?page=TMoverview.

* * *