ALL WMB-MQ Commands

```
svmon -Pt15 | perl -e 'while(<>){print if($.==2||$&&&!$s++);$.=0 if(/^-+$/)}' //summary of the top 15 processes using memory on the system

svmon -G -i 1 2  //Amount of memory in use

svmon -Sut 10 // List of top memory usage of segments

nmon -> tcm
```

Minimum memory requirement calculation----------

Total memory pages (4 KB units) = T + (N * (PD + LD)) + F

where:

T   = Number of pages for text (shared by all users)

N   = Number of copies of this program running simultaneously

PD   = Number of working segment pages in process private segment

LD   = Number of shared library data pages used by the process

F   = Number of file pages (shared by all users)

Multiply the result by 4 to obtain the number of kilobytes required.

=======================

Memory requirements assessment with the rmss command

check if rmss is installed >> lslpp -ll bos.perf.tools

=================Java monitoring tools=======

```
--turningEGJVMGCOn.sh

mqsichangeproperties ${MQSI_BROKER_NAME} -e $1 -o ComIbmJVMManager -n jvmVerboseOption -v"gc"

param1=$1

mqsichangeproperties ${MQSI_BROKER_NAME} -e $1 -o ComIbmJVMManager -n jvmSystemProperty -v"-verbose:gc -Xverbosegclog:/tmp/${param1}.gc.trc"

--turningEGJVMGCOff.sh

mqsichangeproperties ${MQSI_BROKER_NAME} -e $1 -o ComIbmJVMManager -n jvmVerboseOption -v"none"

mqsichangeproperties ${MQSI_BROKER_NAME} -e $1 -o ComIbmJVMManager -n jvmSystemProperty -v" "
```

There are a few tools you can use to monitor and identify performance inhibitors in your Java™ applications.

vmstat

Provides information about various system resources. It reports statistics on kernel threads in the run queue as well as in the wait queue, memory usage, paging space, disk I/O, interrupts, system calls, context switches, and CPU activity.

iostat

Reports detailed disk I/O information.

topas

Reports CPU, network, disk I/O, Workload Manager and process activity.

tprof

Profiles the application to pinpoint any hot routines or methods, which can be considered performance problems.

ps -mo THREAD

Shows to which CPU a process or thread is bound.

Java profilers [-Xrunhprof, Xrunjpa64]

Determines which routines or methods are the most heavily used.

java -verbose:gc

Checks the impact of garbage collection on your application. It reports total time spent doing garbage collection, average time per garbage collection, average memory collected per garbage collection, and average objects collected per garbage collection.

==

Start Service Trace - startServiceTrace.sh >> mqsichangetrace ${MQSI_BROKER_NAME} -t -e $1 -l debug -c 1500000

Stop Service Trace -mqsichangetrace ${MQSI_BROKER_NAME} -t -e $1 -l none

retrieveServiceTrace.sh-- mqsireadlog ${MQSI_BROKER_NAME} -t -e $1 -o /tmp/servicetrace.xml

formatServiceTrace.sh--mqsiformatlog -i /tmp/servicetrace.xml -o /tmp/formattedservicetrace.txt

clearServiceTrace.sh - mqsichangetrace ${MQSI_BROKER_NAME} -t -e $1 -r

==================================Change EG JVM Size===

```
mqsichangeproperties ${MQSI_BROKER_NAME} -e MDAT -o ComIbmJVMManager -n jvmMaxHeapSize -v 2147483648
```

```
mqsichangeproperties ${MQSI_BROKER_NAME} -e SUPC -o ComIbmJVMManager -n jvmMaxHeapSize -v 1073741824
```

```
mqsichangeproperties ${MQSI_BROKER_NAME} -e SAPR3OUT -o ComIbmJVMManager -n jvmMaxHeapSize -v 1073741824
```

==

setJMSConfigurableProperties.sh: echo *** Set the JMS Bindings information

```
mqsichangeproperties ${MQSI_BROKER_NAME} -c JMSProviders -o WebSphere_MQ -n connectionFactoryName -v qcfSFX
```

```
mqsichangeproperties ${MQSI_BROKER_NAME} -c JMSProviders -o WebSphere_MQ -n jndiBindingsLocation -v file:/var/mqsi/jndi/
```

reportJMSConfigurableProperties.sh: echo *** Report the JMS Bindings information

```
mqsireportproperties ${MQSI_BROKER_NAME} -c JMSProviders -o WebSphere_MQ -r
```

resetJMSConfigurableProperties.sh: mqsichangeproperties ${MQSI_BROKER_NAME} -c JMSProviders -o WebSphere_MQ -n connectionFactoryName -v ''

```
mqsichangeproperties ${MQSI_BROKER_NAME} -c JMSProviders -o WebSphere_MQ -n jndiBindingsLocation -v ''
```

```
========================================

startSnapshotSpecificEG.sh: mqsichangeflowstats ${MQSI_BROKER_NAME} -s -e $1 -j -c active -t basic -n advanced -o xml

stopSnapshotSpecificEG.sh : mqsichangeflowstats ${MQSI_BROKER_NAME} -s -e $1 -j -c inactive -t none -n none

startSnapshotSpecificAppl.sh: mqsichangeflowstats ${MQSI_BROKER_NAME} -s -e $1 -j -k $2 -c active -t basic -n advanced -o xml

stopSnapshotSpecificAppl.sh: mqsichangeflowstats ${MQSI_BROKER_NAME} -s -e $1 -j -k $2 -c inactive -t none -n none

startSnapshotAllFlowStats.sh: mqsichangeflowstats ${MQSI_BROKER_NAME} -s -g -j -c active -t basic -n advanced -o xml

stopSnapshotAllFlowStats.sh: mqsichangeflowstats ${MQSI_BROKER_NAME} -s -g -j -c inactive -t none -n none

========================================

reportSAPConfigurableServices.sh:mqsireportproperties ${MQSI_BROKER_NAME} -c SAPConnection -o AllReportableEntityNames -r

reportEISProperties.sh: mqsireportproperties ${MQSI_BROKER_NAME} -c EISProviders -o AllReportableEntityNames  -r

reportBrokerwideListener.sh:

mqsireportproperties ${MQSI_BROKER_NAME} -b httplistener -o HTTPListener -a

mqsireportproperties ${MQSI_BROKER_NAME} -b httplistener -o HTTPConnector -a

reportMonitoringAll.sh: mqsireportflowmonitoring ${MQSI_BROKER_NAME} -g -j

reportWebadminAll.sh: mqsireportproperties ${MQSI_BROKER_NAME} -b webadmin -o AllReportableEntityNames -r

reportDataCapture.sh: mqsireportproperties ${MQSI_BROKER_NAME} -e RECRPLY  -o ComIbmDataCaptureManager -a

reportBrokerAll.sh: mqsireportproperties ${MQSI_BROKER_NAME} -c AllTypes -o AllReportableEntityNames -r

reportBrokerStores.sh:

mqsireportproperties ${MQSI_BROKER_NAME} -o BrokerRegistry -n brokerKeystoreFile

mqsireportproperties ${MQSI_BROKER_NAME} -o BrokerRegistry -n brokerTruststoreFile

mqsireportproperties ${MQSI_BROKER_NAME} -o ComIbmJVMManager -a -e MCHA

reportBrokerKeys.sh:

keytool -list -v -keystore /var/mqsi/PKI/${MQSI_BROKER_NAME}_KeyStore.jks -storepass wmbadm1n

keytool -list -v -keystore /opt/IBM/mqsi/8.0.0.1/jre16/lib/security/cacerts -storepass changeit

mqsireportproperties ${MQSI_BROKER_NAME} -e $1 -o ComIbmJVMManager -a

=====================================================================

setupWebUser.sh: mqsiwebuseradmin ${MQSI_BROKER_NAME} -c -u wmbadmin -a wmbadm1n -r wmbadmin

setupRecordReplay.sh:

mqsideleteconfigurableservice ${MQSI_BROKER_NAME} -c DataCaptureStore -o RecordAndReplay

mqsicreateconfigurableservice ${MQSI_BROKER_NAME} -c DataCaptureStore -o RecordAndReplay -n dataSourceName,egForRecord,egForView -v MBRECORD,RECRPLY,RECRPLY

mqsideleteconfigurableservice ${MQSI_BROKER_NAME} -c DataCaptureSource -o RecordAndReplay
```

```
mqsicreateconfigurableservice ${MQSI_BROKER_NAME} -c DataCaptureSource -o RecordAndReplay -n dataCaptureStore,topic -v RecordAndReplay,'$SYS/Broker/'${MQSI_BROKER_NAME}'/Monitoring/#'

mqsideleteconfigurableservice ${MQSI_BROKER_NAME} -c DataDestination -o RecordAndReplay

QMFRONT=`echo ${MQSI_BROKER_NAME} | cut -c1-11`

mqsicreateconfigurableservice ${MQSI_BROKER_NAME} -c DataDestination -o RecordAndReplay -n egForReplay,endpoint,endpointType -v RECRPLY,wmq:/msg/queue/REPLAY.FLOW.WMB@${QMFRONT}.QM1,WMQDestination

mqsichangeproperties ${MQSI_BROKER_NAME} -o ComIbmDataCaptureManager -n enabled -v true
```

setupWebAdmin.sh:

```
mqsichangeproperties ${MQSI_BROKER_NAME} -b webadmin -o server -n enabled -v true

mqsichangeproperties ${MQSI_BROKER_NAME} -b webadmin -o HTTPConnector -n port -v 8080
```

setupWebAdminSecurity.sh:

```
mqsistop ${MQSI_BROKER_NAME}

mqsichangebroker ${MQSI_BROKER_NAME} -s active

mqsistart ${MQSI_BROKER_NAME}
```

setupMoreWebUsers.sh:

```
mqsiwebuseradmin ${MQSI_BROKER_NAME} -c -u wmbrr1 -a wmbrr1 -r wmbrr1

mqsiwebuseradmin ${MQSI_BROKER_NAME} -c -u wmbrr2 -a wmbrr2 -r wmbrr2
```

setupBrokerTruststore.sh:

```
mqsichangeproperties ${MQSI_BROKER_NAME} -o BrokerRegistry -n brokerTruststoreFile -v /opt/IBM/mqsi/8.0.0.1/jre16/lib/security/cacerts

mqsisetdbparms ${MQSI_BROKER_NAME} -n brokerTruststore::password -u ignore -p changeit
```

setupBrokerKeystore.sh:

```
mqsichangeproperties ${MQSI_BROKER_NAME} -o BrokerRegistry -n brokerKeystoreFile -v /var/mqsi/PKI/${MQSI_BROKER_NAME}_KeyStore.jks

mqsisetdbparms ${MQSI_BROKER_NAME} -n brokerKeystore::password -u ignore -p wmbadm1n
```

==============================

setBrokerwideListener.sh :

```
echo \*\*\* Disable all existing Execution Group HTTP and SOAP listeners

mqsichangeproperties ${MQSI_BROKER_NAME} -o ExecutionGroup -n soapNodesUseEmbeddedListener -v false

mqsichangeproperties ${MQSI_BROKER_NAME} -o ExecutionGroup -n httpNodesUseEmbeddedListener -v false

echo \*\*\* Enable the Broker wide HTTP listener

mqsichangeproperties ${MQSI_BROKER_NAME} -b httplistener -o HTTPConnector -n port -v 8070
```

setSAPLibs.sh:

```
mqsichangeproperties ${MQSI_BROKER_NAME} -c EISProviders -o SAP  -n jarsURL -v /var/mqsi/SAP
mqsichangeproperties ${MQSI_BROKER_NAME} -c EISProviders -o SAP  -n nativeLibs -v /var/mqsi/SAP
```

set_webusers_MQauth.sh:

```
# MQ authorities for wmbrr1  (web admin user with read only access and readonly access to record/replay, ie. view only)
setmqaut -m UNXS0383.MB.QM1 -t qmgr -p wmbrr1 -all +connect +inq
setmqaut -m UNXS0383.MB.QM1 -t queue -p wmbrr1 -n SYSTEM.BROKER.AUTH -all +inq
setmqaut -m UNXS0383.MB.QM1 -t queue -p wmbrr1 -n SYSTEM.BROKER.AUTH.default -all
setmqaut -m UNXS0383.MB.QM1 -t queue -p wmbrr1 -n SYSTEM.BROKER.DEPLOY.QUEUE -all +put
setmqaut -m UNXS0383.MB.QM1 -t queue -p wmbrr1 -n SYSTEM.BROKER.DEPLOY.REPLY -all +get +put
setmqaut -m UNXS0383.MB.QM1 -t queue -p wmbrr1 -n SYSTEM.BROKER.DC.AUTH -all +inq
====================
```

backupBroker.sh:

```
mqsistop ${MQSI_BROKER_NAME}
mqsibackupbroker ${MQSI_BROKER_NAME} -d /tmp/backups
mqsistart ${MQSI_BROKER_NAME}
```

restoreBroker.sh:

```
mqsistop ${MQSI_BROKER_NAME}
mqsirestorebroker ${MQSI_BROKER_NAME} -d /tmp/backups -a $1
mqsistart ${MQSI_BROKER_NAME}
=========
```

```
checkRRDatabase.sh:mqsicvp -n MBRECORD -u MBRECORD -p MBR3C0RD
enableMonitoringAll.sh: mqsichangeflowmonitoring ${MQSI_BROKER_NAME} -g -j -c active
disableMonitoringAll.sh: mqsichangeflowmonitoring ${MQSI_BROKER_NAME} -g -j -c inactive
==============
```

WMBControl.sh :

```
#! /bin/bash
#
# Start / Stop message Broker and MQ
#. /etc/rc.d/init.d/functions
#RETVAL=0
start()
      {
   QMFRONT=`echo ${MQSI_BROKER_NAME_2} | cut -c1-8`
      echo \*\*\* Starting WebSphere MQ Message Broker Queue Manager
      strmqm ${QMFRONT}.MB.QM2
   echo \*\*\* Starting WebSphere MQ MFT Coordination Queue Manager
      strmqm ${QMFRONT}.CDN.QM2
   echo \*\*\* Starting WebSphere MQ MFT Command Queue Manager
      strmqm ${QMFRONT}.CMD.QM2
#   echo \*\*\* Starting WebSphere MQ MFT Client Agent Queue Manager
#      strmqm ${QMFRONT}.AGT.QM2

      while true
      do
      RUNNING=`dspmq|grep -i RUNNING |wc -l`
      if [ ${RUNNING} = 3 ]
      then
      break
      fi
      done

      echo \*\*\* Starting WebSphere Message Broker
      mqsistart ${MQSI_BROKER_NAME}

#      echo \*\*\* Starting MFT Database Logger
#      . /var/mqm/scripts/${QMFRONT}/startMFTDatabaseLogger.sh

      }
stop()
      {
   QMFRONT=`echo ${MQSI_BROKER_NAME_2} | cut -c1-8`
   echo \*\*\* Stopping WebSphere Message Broker
      mqsistop ${MQSI_BROKER_NAME}

#      echo \*\*\* Stopping WebSphere MQ MFT Client Agent Queue Manager
#      endmqm ${QMFRONT}.AGT.QM2
   echo \*\*\* Stopping WebSphere MQ MFT Command Queue Manager
```

```
      endmqm ${QMFRONT}.CMD.QM2
 echo \*\*\* Stopping WebSphere MQ MFT Coordination Queue Manager
      endmqm ${QMFRONT}.CDN.QM2
      echo \*\*\* Stopping WebSphere MQ Message Broker Queue Manager
      endmqm ${QMFRONT}.MB.QM2


      while true
      do
      ENDED=`dspmq|grep -i "Ended" |wc -l`
      if [ ${ENDED} = 3 ]
      then
      break
      fi
      done

#             echo \*\*\* Stopping MFT Database Logger
#      . /var/mqm/scripts/${QMFRONT}/stopMFTDatabaseLogger.sh

      }
force()
   {
      QMFRONT=`echo ${MQSI_BROKER_NAME_2} | cut -c1-8`
   echo \*\*\* Stopping WebSphere Message Broker Immediately
      mqsistop -i ${MQSI_BROKER_NAME}

#      echo \*\*\* Stopping WebSphere MQ MFT Client Agent Queue Manager Immediately
#      endmqm -i ${QMFRONT}.AGT.QM2
   echo \*\*\* Stopping WebSphere MQ MFT Command Queue Manager Immediately
      endmqm -i ${QMFRONT}.CMD.QM2
   echo \*\*\* Stopping WebSphere MQ MFT Coordination Queue Manager Immediately
      endmqm -i ${QMFRONT}.CDN.QM2
      echo \*\*\* Stopping WebSphere MQ Message Broker Queue Manager Immediately
```

```
    endmqm -i ${QMFRONT}.MB.QM2

#   echo \*\*\* Stopping MFT Database Logger
#      . /var/mqm/scripts/${QMFRONT}/stopMFTDatabaseLogger.sh

  }
status()
  {
  echo \*\*\* Queue Manager Status
  dspmq
  echo \*\*\* Message Broker Status
    mqsilist
  }
case "$1" in
  start)
    start
    ;;
  stop)
    stop
    ;;
  force)
    force
    ;;
  status)
    status
    ;;
  *)
    echo "Usage: $0 {start|stop|force|status}"
    exit 1
esac
exit
```

===========================SET-II===

DBA Team: @62547

Project Code: 17281 CR-1

Topas - memory utilization graphical

```
cat wmbevent.log |grep "Aug  4" | grep "user:err|error" | more

cat wmbevent.log |grep "May  2" | grep " FulfilmentSourceService" | more

cat wmbevent.log |grep "2013-01-19T09” |more

C:\>mqsireportproperties BKR3 -o ComIbmJVMManager  -e TEST4 -a

C:\>mqsireportproperties BKR3 -o ComIbmXmlParserFactory  -e TEST4 -a

C:\>mqsireportproperties UNXS0389.MB1 -o AllReportableEntityNames -a

C:\>mqsireportproperties BKR3 -o ComIbmJVMManager  -e TEST4 –r

C:\>mqsireportresourcestats UNXS0389.MB1 -e MDAT

cat wmbevent.log |grep "Sep 17" >> /support/home/wmbadmin/WMB.log

mqsibrowse UNXS0460.MB1 -t BROKERAAEG

echo `netstat -i | grep unxs | grep -v "bk" | awk '{ print $4 }' | tr '[:lower:]' '[:upper:]'`.MB.QM1
```

----Display specific status of queue attributes

```
DISPLAY QSTATUS(ERRORS.WMB.ERRORLOG.INPROGRESS) CURDEPTH UNCOM IPPROCS LGETDATE LGETTIME LPUTDATE LPUTTIME MONQ OPPROCS

DISPLAY CONN(*) TYPE(CONN) ALL

echo 'DISPLAY CONN(*) WHERE(UOWSTATE EQ ACTIVE)' |runmqsc UNXS0383.MB.QM1 |more

echo 'DISPLAY CONN(*) TYPE(CONN) ALL' | runmqsc UNXS0115.QM1 |grep 'AMQ8276'|grep 'UOWSTATE'|wc –l

echo 'DISPLAY CONN(*) TYPE(CONN) ALL' | runmqsc UNXS0115.QM1  >> CONNECTSTAT.txt
```

```
./qload -m UNXS03911.MB.QM1 -I ERRORS.WMB.ARCHIVE -f message1 -r1

mqsireportbroker UNXS0460.MB1

fteListAgents

fteListAgents -v

ftePingAgent MFTHRSAXPPRD

fteShowAgentDetails MFTHRSXAPPRD

fteShowAgentDetails -v MFTHRSAPPRD

fteStartAgent MFTETLUXATZ
```

```
fteStopAgent MFTATGBCCPRXF

fteStopAgent -i MFTHRSAPPRX  //force fully stop

fteCreateAgent -agentName MFTHRSX -agentQMgr UNXS0393.AGT.QM1 –f

fteCreateMonitor -ma MFTHRSAPPRD -mm UNXS039X.AGT.QM1 -mn ServiceNowPRDMonitor -md
/var/mqsi/tempdata/saphrmqmft

/HRMQFT/ServiceNow -mt /var/mqm/scripts/MFTHRSAPPRDScripts/ServiceNowPRDTemplate.xml -pi 5 -pu minutes -tr
match,"*.csv"


fteDeleteAgent MFTHRSAPPXD

fteDeleteMonitor -ma MFTHRSAPPRXD -mm UNXS0393.AGT.QM1 -mn ServiceNowPRDMonitor


./showMFTPRDAgents.sh

./startMFTHRSAPPRDAgent.sh

stopMFTHRSAPPRDAgent.sh

stopMFTHRSAPPRDAgent.sh

./startMFTHRSAPPRDA>


unxa0393:/var/mqm/scripts/MFTHRSAPPRDScripts>ll

total 168

drwxrws---    2 wmbadmin mqbrkrs       4096 12 Apr 11:15 .

drwxrwsrwx   11 wmbadmin mqbrkrs       4096 13 Jun 12:18 ..

-rwxrwxrwx    1 wmbadmin mqbrkrs        140 21 Feb 12:29 agent.properties

-rwxrwxrwx    1 wmbadmin mqbrkrs      12176 11 Apr 14:23 archive_53.pl

-rw-rw----    1 wmbadmin mqbrkrs       1320 24 Jun 14:02 archive_snow.log

-rwxrwxrwx    1 wmbadmin mqbrkrs        266 21 Feb 12:31 createServiceNowPRDMonitor.sh

-rwxrwxrwx    1 wmbadmin mqbrkrs         54 21 Feb 12:31 deleteMFTHRSAPPRDAgent.sh

-rwxrwxrwx    1 wmbadmin mqbrkrs        113 21 Feb 12:32 deleteServiceNowPRDMonitor.sh

-rwxrwxrwx    1 wmbadmin mqbrkrs         56 21 Feb 12:32 forceMFTHRSAPPRDAgent.sh

-rwxrwxrwx    1 wmbadmin mqbrkrs        852 21 Feb 12:32 MFTPRDControl.sh

-rwxrwxrwx    1 wmbadmin mqbrkrs         55 21 Feb 12:34 pingMFTHRSAPPRDAgent.sh

-rwxrwxrwx    1 wmbadmin mqbrkrs         74 21 Feb 12:34 pingMFTSNOWPRDAgent.sh

-rwxrwxrwx    1 wmbadmin mqbrkrs        965 21 Feb 12:38 ServiceNowPRDTemplate.xml

-rwxrwxrwx    1 wmbadmin mqbrkrs         97 21 Feb 12:38 setupMFTHRSAPPRDAgent.sh
```

```
-rwxrwxrwx    1 wmbadmin mqbrkrs         69 21 Feb 12:39 showMFTHRSAPPRDAgent.sh
-rwxrwxrwx    1 wmbadmin mqbrkrs         47 06 Sep 2012  showMFTPRDAgents.sh
-rwxrwxrwx    1 wmbadmin mqbrkrs       3910 12 Apr 11:16 snow_archive_control.ksh
-rwxrwxrwx    1 wmbadmin mqbrkrs         95 21 Feb 12:40 startMFTHRSAPPRDAgent.sh
-rwxrwxrwx    1 wmbadmin mqbrkrs         53 21 Feb 12:41 stopMFTHRSAPPRDAgent.sh
```

Issue with Agent when no reachable – Restart Agent & update the file to be processed – using touch command

```
/var/mqsi/tempdata/saphrmqmft/HRMQFT/ServiceNow>
```

```
grep -r -n "1891641427" .
```

jbossadm

test1234

```
cat wmbevent.log |grep "Jun 23" | grep "A schema" |more
```

```
D:\Program Files\IBM\MQSI\8.0.0.0>runmqsc QM1
5724-H72 (C) Copyright IBM Corp. 1994, 2009.  ALL RIGHTS RESERVED.
Starting MQSC for queue manager QM1.
```

```
cat KITS_ErrorLog_20112012.txt | grep "2012-12-20T17:19" | more
find . -type f -print |xargs grep -li 'B28013'
```

```
du -ak * | sort -nr >> /support/home/wmbadmin/diskUsage.txt
awk '{c+=gsub(s,s)}END{print c}' s='/HazardousMaterial' KITS_ErrorLog_08032013.txt
awk '{c+=gsub(s,s)}END{print c}' s='/HazardousMaterial' *.txt
2012-11-20T
```

```
alter ql(TEST_IN) PUT(DISABLED)
     1 : alter ql(TEST_IN) PUT(DISABLED)
```

```
alter qa(FULFILMENTSOURCE.STEP.INBOUND.WMB) GET(ENABLED)


mqsireportproperties UNXS0462.MB1   -e TEST_INSTANCE -o ComIbmJVMManager –a

mqsireportproperties UNXS0378.MB1   -e SWEB  -o ComIbmJVMManager -r

mqsireportproperties UNXS0391.MB1   -e MDAT  -o ComIbmJVMManager -r



mqsichangeproperties  UNXS0378.MB1   -e  SWEB  -o ComIbmJVMManager -n jvmMaxHeapSize -v 2147483648

mqsistopmsgflow UNXS0231.MB1  -e DarwinTest

mqsistartmsgflow UNXS0231.MB1 -e DarwinTest


mqsireportflowmonitoring UNXS0391.MB1   -g –j

mqsireportflowmonitoring UNXS0391.MB1 -e SUPC -k Shipment_synchroniseShipment-ASN -j

mqsistopmsgflow UNXS0389.MB1  -e MDAT -k ProductRangingService -m com.kingfisher.ukbq.ProductRangingService.ProductRanging_STEP_Pub

mqsistartmsgflow UNXS0389.MB1  -e MDAT -k ProductRangingService -m com.kingfisher.ukbq.ProductRangingService.ProductRanging_STEP_Pub


mqsistartmsgflow UNXS0378.MB1  -e SAPR3OUT -k SAPR3GenericOutbound


mqsisetdbparms UNXS0383.MB1 -n MBREPOS -d

mqsisetdbparms UNXS0383.MB1 -n MBODS -d


mqsisetdbparms BKR1 -n MBREPOS -u MBREPOS -p MBR3P0S

mqsisetdbparms UNXS0383.MB1 -n MBODS -u MBODS -p MB0D5


PATH=/usr/bin:/etc:/usr/sbin:/usr/ucb:$HOME/bin:/usr/bin/X11:/sbin:.


export PATH


if [ -s "$MAIL" ]          # This is at Shell startup.  In normal

then echo "$MAILMSG"       # operation, the Shell checks
```

```
fi                      # periodically.
TMOUT=43200 ; TIMEOUT=43200
export readonly TMOUT TIMEOUT

#Set the Prompt
PS1=`hostname`':$PWD'">"
export PS1

alias ll='ls -al'
set -o vi

export ODBCINI=/var/mqsi/odbc/odbc.ini
export ODBCSYSINI=/var/mqsi/odbc/odbcinst.ini
export MQSI_BROKER_NAME=`netstat -i | grep unxs | grep -v "bk" | awk '{ print $4 }' | tr '[:lower:]' '[:upper:]'`.MB1
export CLASSPATH=/var/mqsi/shared-classes/ojdbc6.jar:$CLASSPATH

alias taillog='tail -f /var/mqsi/log/${MQSI_BROKER_NAME}/wmbevent.log'
ls -1 S000006[89].*|xargs rm –fr
awk '{c+=gsub(s,s)}END{print c}' s='UNXS0391.MB1' KITS_ErrorLog_08032013.txt

/usr/mqm/bin/runmqlsr -r -m UNXS0391.MB.QM1 -t TCP -p 1430
/usr/mqm/bin/runmqchi -m UNXS0391.MB.QM1 -q SYSTEM.CHANNEL.INITQ -rUNXS0391.MB.QM1
ps -eaf|grep mqm |grep UNXS0391.MB.QM1
/usr/mqm/bin/runmqchl -c UNXS0391.TO.LNXS0236 -m UNXS0391.MB.QM1

find . -exec grep -l B290139073 {} \;
grep -l FulfilmentSource_STEP_Pub *

B280138116




mqsichangetrace UNXS0240.MB1 -u -e RETL  -l debug -r -c 50000
mqsichangetrace UNXS0240.MB1 -u -e RETL  -l none

mqsireadlog UNXS0240.MB1 -u -e RETL  -f -o MB1_HT1.xml

mqsiformatlog -i MB1_HT1.xml -o MB1_HT2.txt
```

Service Trace:

```
mqsichangetrace BKR2 -t -b -l debug

mqsichangetrace BKR2 -t -b -l none

mqsireadlog BKR2 -t -e EG3  -f -o MB1_HT3.xml

mqsiformatlog -i MB1_HT3.xml -o MB1_HT3.txt
```

:FTE Alert:

This is a Known issue on the system.

Defect already raised with Mike Park

Reference Incident: 1938468,1993207

Host Name : unxs0007.uk.b-and-q.com

Host IP   : 172.19.174.16 / 28.4.174.16

System Number : 01

Client : 901

Language Code : EN

Code Page: 1100

UserName: MBROKER

Password: Wmb@dm1n

```
# ipcs -l //show limit

# ipcs -s /show semafore

# impcs -m / show Shared Memory

# ipcs -s 4194315 3145740 13

# ipcrm -s 818937948
```

=================================SET-III==========================

ATG command :

```
runAssembler sample.ear -m PioneerCyclingJSP DafEar.Admin
```

java weblogic.DDConverter -d . sample.ear --It will read the EAR file and outside of the EAR application it will generate the fresh XML files.

----

netstat -a | find "LISTENING"

mqsisetdbparms MB7BROKER -n TESTBKDB -u db2admin -p db2admin200

mqsisetdbparms MB7BROKER -n INF_ECOMM_INT -u informix -p wipro@123

mqsisetdbparms L_BROKER1_MAXISADM -n TESTDB -u db2admin -p db2admin200

mqsisetdbparms D_SOA_BRK_01 -n INTDEV -u wmbrks -p wmbrks -a // '-a' option: incase of MB 6.1.0.8 will work when broker is running

mqsisetdbparms L_BROKER1_MAXISADM -n L_BRKDB1 -u maxisadm -p maxisadm11 // not possible to change uid and pass for Broker DB.

mqsicvp MB7BROKER -n INF_ECOMM_INT -v // Test Database connectivity from The broker , shows all details of the DSN to use by the broker

mqsicvp DEV_BRKR1 -n MBREPOS -v

mqsicvp -n MBREPOS -u MBREPOS -p MBREPOS

mqsisetdbparms UNXS0231.MB1 -n MBREPOS -d --deleted all previously created identifiers

mqsicvp UNXS0231.MB1 -n MBREPOS -v

mqsicvp -n MBREPOS -u MBREPOS -p MBREPOS

mqsichangebroker L_BROKER1_MAXISADM -i maxisadm -a maxisadm11 -p maxisadm11 // in case service password been changed, use it

mqsireload D_SOA_BRK_01 -e EG_InternetHotticket

mqsireportproperties MB7BROKER -e EG_CommonServices -o HTTPSConnector -n port // Show the port number for the execution group

mqsireportproperties MB7BROKER -c JDBCProviders -o INF_ECOMM_INT -r //jdbcProviders details

mqsireportproperties MB7BROKER -c JDBCProviders -a -o AllReportableEntityNames //to view the list of available JDBCProvider services

mqsireportproperties DEV_BRKR1 -c JMSProviders -a -o AllReportableEntityNames

mqsireportproperties DEV_BRKR1 -c AllTypes -o AllReportableEntityNames -r

mqsiservice -v // product version

mqsiservice DEV_BRK5 -t /time details

```
mqsicreatebroker MB7BROKER -i abhijith -a wipro@123 -q MB7QMGR

sc delete MQSeriesBrokerMB7BROKER

mqsideletebroker MB7BROKER -q

dltmqm MB7QMGR
```

===========mqsiapplybaroverride ================================

Create a configuration file called mbconfig.properties at some path with the following lines (with the proper values inserted):

```
STEPURL=path to the STEP web service for the tier

com.kingfisher.ukbq.ProductSetService.ProductSet_STEP_Sub#USERNAME=STEP User ID

com.kingfisher.ukbq.ProductSetService.ProductSet_STEP_Sub#PASSWORD=STEP Password
```

E.g.

```
STEPURL=http://localhost:7800/step/ws

com.kingfisher.ukbq.ProductSetService.ProductSet_STEP_Sub#USERNAME=mbsuperuser

com.kingfisher.ukbq.ProductSetService.ProductSet_STEP_Sub#PASSWORD=mbsuperuser
```

4.1.2.1 Deployment of the .Bar file

There are some properties in the .Bar file that need to be overridden with tier-specific values before deployment.

Perform the following steps whenever you build the .Bar file:

- Extract the bar file (with a zip file extractor program like WinZip) contents into a directory

- Run the following command from MQSI command console with the

o mqsiapplybaroverride -b “<appzip file path>/ProductSetService.appzip” -p "<config file name with complete path>/mbconfig.properties"

o Here <appzip file path> should be replaced by the path where the contents of the bar file are extracted (including the file ProductSetService.appzip); and "<config file name with complete path> should be replaced with the path where mbconfig.properties has been created

- Now add the ProductService.appzip file back to the .Bar file

The .Bar file can now be deployed.

OR- 2nd Method(easy)

```
mqsiapplybaroverride -b TestBusinessApp.bar -k TestBusinessApp -r -m
"com.kingfisher.ukbq.Order.TestBusinessAppMF#additionalInstances=35,com.kingfisher.ukbq.Order.TestBusinessAppMF#MQ
Input.additionalInstances=40"
```

===========

```
tracert/traceroute  <IP> <Port>
```

RFUTILS.SVRCONN/TCP/172.19.174.41(1430) --to connect to remote QM from local rfhutilc

Broker's home from where mqsc command can be run: /opt/IBM/mqsi/8.0.0.0/bin

Odbc path: /mbud1/mqsi/odbc

=====

```
dspmqver //know MQ version

mqsilist BRK_01 -e EG // list of deployed flows

mqsichangetrace L_BROKER1_MAXISADM -n off -e Test  -r //changes Trace node's setting to OFF

mqsichangeproperties DEV_BROKER1 -c EISProviders -o SAP -n jarsURL,nativeLibs -v C:\SAP_JCO

------------------runmqsc----------------------------

display q(*) where (usage eq XMITQ)

DIS QMGR              // display qm setting

DIS Q(*) CURDEPTH WHERE(CURDEPTH GT 0)  // display non empty queues

DIS CHL(*) ALL  //Display channels,name & type

DIS CHL(*) CONNAME //Display channels with specific atributes

DIS SERVICE(*) ALL  // Display service name & types all attributes

DIS LSTR(*) ALL  // Display Listners name & types all attributes

-----------------------------------------

amqmdain reg QM.MAXISADM -c  display -s Log -v * // display log setting

dspmqfls -m QM.MAXISADM -t qlocal *  //display queue filenames for queues
```

================

```
crtmqm -c "local queue" -ll -q QM.MAXISADM

mqsicreateconfigmgr CONFIG_MANAGER_MAXISADM -i maxisadm -a maxisadm -q QM.MAXISADM
```

```
mqsicreatedb L_BRKDB1 -i maxisadm -a maxisadm -e DB2  //ODBC Data Source name 'L_BRKDB1' created succesfully

mqsicreatebroker L_BROKER1_MAXISADM -i maxisadm  -a maxisadm -q QM.MAXISADM -n L_BRKDB1

mqsideleteconfigmgr CONFIG_MANAGER_MAXISADM

mqsideletedb L_BRKDB1

mqsideletebroker L_BROKER1_MAXISADM

strmqm QM.MAXISADM

==================Creating Configurable service/Adpater for SAP connection============

mqsicreateconfigurableservice UNXS0459.MB1 -c SAPConnection -o SAPR3IDOCOUTBOUND.outadapter

mqsichangeproperties UNXS0459.MB1 -c SAPConnection -o SAPR3IDOCOUTBOUND.outadapter -n applicationServerHost -v SAPZ97WMB

mqsichangeproperties UNXS0459.MB1 -c SAPConnection -o SAPR3IDOCOUTBOUND.outadapter -n client -v 900

mqsichangeproperties UNXS0459.MB1 -c SAPConnection -o SAPR3IDOCOUTBOUND.outadapter -n gatewayHost -v SAPZ97WMB

mqsichangeproperties UNXS0459.MB1 -c SAPConnection -o SAPR3IDOCOUTBOUND.outadapter -n gatewayService -v sapgw01

mqsichangeproperties UNXS0459.MB1 -c SAPConnection -o SAPR3IDOCOUTBOUND.outadapter -n systemNumber -v 01

mqsichangeproperties UNXS0459.MB1 -c SAPConnection -o SAPR3IDOCOUTBOUND.outadapter -n connectionIdleTimeout -v 0

mqsireportproperties UNXS0459.MB1 -c SAPConnection -o AllReportableEntityNames -r

mqsideleteconfigurableservice MB8BROKER -c SAPConnection -o mySAPAdapter.outadapter


=============================================================

mqsilist

mqsistart CONFIG_MANAGER_MAXISADM

mqsistart L_BROKER1_MAXISADM

===================================

mqsichangetrace L_BROKER1_MAXISADM -u -e Test  -l debug -r -c 50000

mqsichangetrace L_BROKER1_MAXISADM -u -e Test  -l none

mqsireadlog L_BROKER1_MAXISADM -u -e Test -f -o UMB_HT1.xml

mqsiformatlog -i UMB_HT1.xml -o UMB_HT2.txt

notepad UMB_HT2.txt

========================================================

mqsichangetrace UNXS0231.MB1 -u -e DarwinTest  -k Location -f com.kingfisher.ukbq.LocationService.LocationService -l debug -c 5000 -r


mqsichangetrace UNXS0231.MB1 -u -e DarwinTest -k Location -f com.kingfisher.ukbq.LocationService.LocationService -l none
```

```
mqsireadlog UNXS0231.MB1  -u -e DarwinTest -o trace2.xml

mqsiformatlog -i trace2.xml -o formattraceSync.log
```

=================

```
mqsireportproperties D_SOA_BRK_01 –e EG_InternetHotticket -o HTTPConnector -n explicitlySetPortNumber  -v 9003
```

```
mqsichangebroker L_BROKER1_MAXISADM -l "C:\lilfolder"

mqsireportbroker  L_BROKER1_MAXISADM //shows Broker's Install path,Work path,Broker UUID,Process id,Queue Manager,User Name Server Queue Manager,Broker database name,Broker database userId,Broker database password,User lil path,User exit path,Active user exits,LDAP principal,LDAP credentials,ICU converter path,HTTP listener port,Pubsub migration,Pubsub access control,Trusted (fastpath) Queue Manager application,Configuration change timeout,Internal configuration timeout,Statistics major interval,Operation mode,Fixpack capability level,Broker registry format

mqsichangeflowmonitoring DEV_BROKER2 -e DEV_EG_DEFAULT -f testevtFlow -s "MQ Input.TransactionStart" -i enabled

mqsichangeflowmonitoring DEV_BROKER2 -e DEV_EG_DEFAULT -k TestEvent -j  //allmessage Flow

mqsichangeflowmonitoring DEV_BRK5 -g -j -c active  //all eg and all msgflow

mqsireportflowmonitoring UNXS0376.MB1 -e SAPR3IN -k SAPR3GenericInbound -j //Display monitoring options for all message flows in application application1 in execution group default
```

Full Name		SAPR3GenericInbound.appzip

===

```
mqsicreateexecutiongroup -i 172.21.33.65 -q DSOABRK1 -b D_SOA_BRK_01 -e EG_InternetHotticket -l -v Test_EG_InternetHotticket.log
```

START/STOP - wmb configuration manager : mqsistart D_SOA_CFG_01

```
mqsistop D_SOA_CFG_01
```

```
mqsicreateexecutiongroup -i 127.0.0.1 -q DSOACFG1 -b D_SOA_BRK_01 -e EG_InternetHotticket -l -v Test_EG_InternetHotticket.log

mqsicreateexecutiongroup -i 127.0.0.1 -p 1414 -q QM.MAXISADM -b L_BROKER1_MAXISADM -e EG_InternetHotticket -l

mqsideleteexecutiongroup -i 127.0.0.1 -p 1414 -q QM.MAXISADM -b L_BROKER1_MAXISADM -e EG_InternetHotticket

DEFINE ql(CIBCHARGING.MSG) maxdepth(999999)

DEFINE ql(PROVISIONING.INCOMPLETE) maxdepth(999999)
```

```
mqsideploy -i 127.0.0.1 -q DSOACFG1 -p 2415 -b D_SOA_BRK_01 -e EG_InternetHotticket -a Test_ProvisionConsumerService.bar  -v Test_ProvisionConsumerService.log -w 720

mqsideploy -i 127.0.0.1 -q QM.MAXISADM -p 1414 -b L_BROKER1_MAXISADM -e Test -d MF_AsynRequest.cmf

ps -ef | grep runmqlsr // know the quuemaneger's listners details
```

```
DISPLAY LISTENER(*) ALL //shows all listner of MQ
```

==

SSL over HTTP:

=================

Create a keystore file to store the broker's certificates

Configuring the broker to use SSL on a particular port

Creating a message flow to process HTTPS requests

Testing your example

=================

```
"C:\Program Files\IBM\MQSI\7.0\jre16\bin\keytool" -genkey -keystore myTestKeyStoreFile -storepass db2admin200 -alias myTestKey
```

Choose the keystore file to be used, by setting a value for keystoreFile

```
mqsichangeproperties BROKER_TEST1 -b httplistener -o HTTPSConnector -n keystoreFile -v "C:\Program Files\IBM\MQSI\7.0\jre16\bin\myTestKeyStoreFile"
```

Specify the password for the keystore file, by setting a value for keystorePass

```
mqsichangeproperties BROKER_TEST1 -b httplistener -o HTTPSConnector -n keystorePass -v db2admin200
```

Specify the port on which WebSphere Message Broker will listen for HTTPS requests

```
mqsichangeproperties BROKER_TEST1 -b httplistener -o HTTPSConnector -n port -v 1418
```

Turn on SSL support in message broker, by setting a value for enableSSLConnector

```
mqsichangeproperties BROKER_TEST1 -b httplistener -o HTTPListener  -n enableSSLConnector -v true
```

create a flow only having "HTTPInput" and "HTTPReply" node

Test in IE,Mozila:

https://localhost:1418/testHTTPS

===

Extracting a certificate from another keystore:

```
"C:\Program Files\IBM\MQSI\7.0\jre16\bin\keytool" -export -alias myTestKey -file myTestCert -keystore myTestKeyStoreFile -storepass db2admin200
```

Importing a certificate into the cacerts file:

```
"C:\Program Files\IBM\MQSI\7.0\jre16\bin\keytool" -import -alias myTestKey -file myTestCert -keystore cacerts -storepass changeit
```

==

If you need DTD support, the answer (for now) is to use XMLNS not the XMLNSC.DTD support in XMLNSC parser is limited.

==========================

```
mqsichangetrace L_BROKER1_MAXISADM -u -e Test -f MF_TemService -l debug -r -c 50000

mqsichangetrace L_BROKER1_MAXISADM -u -e Test -f MF_TemService -l none

mqsireadlog L_BROKER1_MAXISADM -u -e Test -f -o INE.xml

mqsiformatlog -i INE.xml -o koyel.txt

notepad INE.xml
```

=================================Configuring HTTP Proxy servlet=============================

```
$mqsichangeproperties L_BROKER1_MAXISADM -e Test -o HTTPConnector -n enableMQListener -v true

$mqsireportproperties L_BROKER1_MAXISADM -e Test -o HTTPConnector -n enableMQListener
```

============================

Call different(from which is specified at node) DSN from ESQL:

```
SET admqry = 'SELECT sysdate from dual;';

SET sDate.Rows[] = PASSTHRU(admqry TO Database.{ADMDB});

---- Insert 0 rows into database - to check insertion/update authority exist or not

PASSTHRU ('INSERT INTO MBCONFIG Select * from MBCONFIG where 1=2');

PASSTHRU ('UPDATE MBCONFIG SET CANVAL = 123 WHERE 1=2');
```

===

stop broker before mqsisetdbparms incase of MB 6.1.0.4 else(in case of 6.1.0.8 use -a option, no restart)

=============DB2 SQL command============

```
select * from WMBRKS.EAI_ERRORLOG order by MSG_OUT_DATE desc fetch first 10 row only;

select * from WMBRKS.EAI_AUDIT_TRAIL order by MSG_OUT_DATE desc fetch first 10 row only;
```

==========================

```
CREATE FIELD OutputRoot.XMLNSC.payload.Body.Order TYPE Name;
CREATE FIRSTCHILD OF OutputRoot.XMLNSC.payload.Body.Order Domain('XMLNSC') NAME 'QueueName' VALUE 'out2';
```

==

```
$ ps -ef | grep runmqchl/runmqlsr
```

dis chs(*) all -this will show you the process ID of the receiver channel - it's in JOBNAME (this is usually process-id/thread id, in hex).

Try a STOP CHL(rcvr chl name) MODE(FORCE) on the remote end.

Wait until all the channels say STOPPED (not STOPPING), then do a START CHL(chl name) on both the remote and local side (i.e. start the channel at both ends).

On Queue Manager: SSOABRK1

```
 DEFINE QL(QL.GEN.INSL.BRK.RES.02) maxdepth(999999)

 DEFINE QL(QL.GEN.INSL.BRK.RES.01) maxdepth(999999)


 DEFINE QREMOTE(QR.CIBCHR.BRK.INSL.REQ.01) DESCR('Queue for CIB charging reqest') RNAME(QL.CIBCHR.BRK.INSL.REQ.01) RQMNAME(SSOAIN1) XMITQ(SSOAIN1)
```

```
1. STOP CHL(SSOAIN1.SSOABRK1.01) MODE(FORCE) -- receiver

4. STOP CHL(SSOABRK1.SSOAIN1.01) MODE(FORCE) -- sender


5. START CHL(SSOABRK1.SSOAIN1.01) --sender

8. START CHL(SSOAIN1.SSOABRK1.01) --receiver
```

On Queue Manager: SSOAIN1

```
DEFINE QL(QL.CIBCHR.BRK.INSL.REQ.01) maxdepth(999999)

DEFINE QREMOTE(QR.GEN.INSL.BRK.RES.02) DESCR('Queue for CIB charging response') RNAME(QL.GEN.INSL.BRK.RES.02) RQMNAME(SSOABRK1) XMITQ(SSOABRK1)
```

```
2. STOP CHL(SSOABRK1.SSOAIN1.01) MODE(FORCE) -- receiver

3. STOP CHL(SSOAIN1.SSOABRK1.01) MODE(FORCE) -- sender


7. START CHL(SSOAIN1.SSOABRK1.01) -- sender
```

6. START CHL(SSOABRK1.SSOAIN1.01) -- receiver

===

```
crtmqm -c "SOA Adapter QM" -lc PSOAADP1 //To create queue manager

strmqm PSOAADP1                                   //To start queue manager

stpmqm PSOAADP1                                             //To stop queue manager

runmqsc PSOAADP1 < WMQ_Script_PRD_PSOAADP1_SMS.txt >> WMQ_Script_PRD_PSOAADP1_SMS.log //To run mq scripts in batch

start chinit initq(SYSTEM.CHANNEL.INITQ) OR start chinit //start the channel initiator from RUNMQSC (Windows NT, UNIX and OS/2). The

start runmqlsr -t tcp -m QMA -p 1414

start runmqchi                                    //use the channel initiator to start channels


runmqlsr -t tcp -m MB7QMGR -p 1414 /start TCP MQ listner
```

The following are to be run in runmqsc.

```
DEFINE LISTENER(PSOAADP1) TRPTYPE(TCP) CONTROL(QMGR) PORT(1416) //To define listener

START LISTENER(PSOAADP1)                                          //To start listener

DEFINE CHL(PSOAADP1.SVRCONN.01) CHLTYPE(SVRCONN) MCAUSER ('mqm') //To define server connection channel

DEFINE CHL(PSOABRK1.PSOAADP1.01) CHLTYPE(RCVR)                    //To define receiver channel

DEFINE QL(PSOABRK1) USAGE(XMITQ) TRIGDATA(PSOAADP1.PSOABRK1.01) INITQ(SYSTEM.CHANNEL.INITQ) //To define transmission queue

DEFINE CHL(PSOAADP1.PSOABRK1.01) CHLTYPE(SDR) TRPTYPE(TCP) CONNAME('172.16.136.27(1414)') XMITQ(PSOABRK1) //To define sender channel

START CHL(PSOAADP1.PSOABRK1.01)                                       //To start channel

DEFINE QL(QL.SMS.SOA.RES.00) DEFPSIST(YES) MAXDEPTH(9999) DESCR(Rpl) //To define local queue

DEFINE QR(QR.SMS.SOA.REQ.01) RNAME(QL.SMS.SOA.REQ.01) RQMNAME(PSOAADP1) XMITQ(PSOAADP1) //To define remote queue
```

==

----Process Triggering--------

1. Create an performance event when queue reaches 80%

```
ALTER QMGR PERFMEV(ENABLED)

ALTER QLOCAL('MYQUEUE') QDEPTHHI(80) QDPHIEV(ENABLED)
```

2. Define a trigger when a msg put into the queue and it calls for a process definition

```
DEFINE QLOCAL(TRIG.EXAMPLE.QLOCAL) +
DESCR('Example Queue for Triggering') +
DEFPRTY(0) +
DEFSOPT(SHARED) +
GET(ENABLED) +
MAXDEPTH(5000) +
MAXMSGL(4194304) +
MSGDLVSQ(PRIORITY) +
PUT(ENABLED) +
RETINTVL(999999999) +
TRIGTYPE(EVERY) +
PROCESS(TRIG.EXAMPLE.PROCESS) +
INITQ(TRIG.EXAMPLE.INITQ) +
USAGE(NORMAL) +
REPLACE
```

3. Process definition calls an application script written in .bat,.sh,etc

```
DEFINE PROCESS (TRIG.EXAMPLE.PROCESS) +
DESCR('Example Process for Triggering') +
APPLTYPE(UNIX) +
APPLICID(/MQ/emailMessage.sh) +
REPLACE
```

5. runmqtrm -m QMNAME -q TRIG.EXAMPLE.INITQ &

4. Script sends an email alert.

```
DEFINE QLOCAL('QL.TRIGGER_MSGS') REPLACE
DEFINE QLOCAL('TRIG.EXAMPLE.QLOCAL') +
    TRIGGER +
    TRIGTYPE (EVERY) +
    INITQ('QL.TRIGGER_MSGS') +
    PROCESS('TRIG.EXAMPLE.PROCESS') +
    REPLACE



win:
DEFINE PROCESS('TRIG.EXAMPLE.PROCESS') +
    USERDATA('PARAM1 PARAM2') +
    APPLICID('D:\var\TriggerAppByMQ.bat') REPLACE

win:
DEFINE PROCESS('TRIG.EXAMPLE.PROCESS') +
    USERDATA('') +
    APPLICID('D:\var\test.bat') REPLACE
----------------------------------------------------
unix:
DEFINE PROCESS('TRIG.EXAMPLE.PROCESS') +
    USERDATA('PARAM1 PARAM2') +
    APPLICID('touch /support/home/wmbtcsdv/Trace/testii.txt') REPLACE


//start runmqtrm  -m UNXS0231.MB.QM1  -q QL.TRIGGER_MSGS
//runmqtrm -m DEV_QM5 -q QL.TRIGGER_MSGS
//runmqtrm -m UNXS0231.MB.QM1 -q QL.TRIGGER_MSGS

///usr/bin/runmqtrm
win:
DEFINE SERVICE('TRIG_MON_START') +
    CONTROL(QMGR) +
    SERVTYPE(SERVER) +
    STARTCMD('C:\Program Files\IBM\WebSphere MQ\bin\runmqtrm.exe') +
    STARTARG('-m +QMNAME+ -q "QL.TRIGGER_MSGS"') +
    STOPCMD('C:\Program Files\IBM\WebSphere MQ\bin\amqsstop.exe') +
    STOPARG('-m +QMNAME+ -p +MQ_SERVER_PID+') +
    STDERR('C:\Program Files\IBM\WebSphere MQ\errors\outerror.log') +
    STDOUT('C:\Program Files\IBM\WebSphere MQ\errors\output.log') +
    REPLACE
```

unix:

```
DEFINE SERVICE('TRIG_MON_START') +
    CONTROL(QMGR) +
    SERVTYPE(SERVER) +
    STARTCMD('/usr/bin/runmqtrm') +
    STARTARG('-m +QMNAME+ -q QL.TRIGGER_MSGS') +
    STOPCMD('/usr/bin/amqsstop') +
    STOPARG('-m +QMNAME+ -p +MQ_SERVER_PID+') +
    REPLACE
```

==================

Queries for Audit Logging:

==============

```
   INSERT INTO WMBRKS.MBRK_AUDIT_CONTROL
   (PROJECT_NAME, FLOW_NAME, AUDIT_IN, AUDIT_OUT, AUDIT_ERR, AUDIT_ERR_LOG, AUDIT_IN_QNAME, AUDIT_OUT_QNAME, AUDIT_ERR_QNAME, AUDIT_ERR_LOG_QNAME, PAYLOADREQ, LAST_UPDATED, REMARKS)
 VALUES
   ('INTERNET_HOTTICKET', 'MF_HT_BroadBand_ProvisioningConsumer', 'Y', 'Y', 'Y', 'Y', 'QL.MBRK.AUDIT.INMSG.01', 'QL.MBRK.AUDIT.OUTMSG.01', 'QL.MBRK.ERR.01', 'QL.MBRK.ERR.LOG.01', 'Y', CURRENT_DATE, 'AbhijitK');
```

==============

```
UPDATE MBRK_AUDIT_CONTROL SET AUDIT_IN = 'N', AUDIT_OUT = 'N' WHERE PROJECT_NAME = 'MF_HT_BroadBand_ProvisioningConsumer';

COMMIT;

DELETE FROM WMBRKS.MBRK_AUDIT_CONTROL WHERE FLOW_NAME = 'MF_HT_BroadBand_ProvisioningConsumer';

SELECT * from WMBRKS.MBRK_AUDIT_CONTROL where FLOW_NAME='MF_HT_CustomerFlow';

SELECT * FROM WMBRKS.EAI_AUDIT_TRAIL order by MSG_OUT_DATE desc fetch first 10 row only;

SELECT * FROM WMBRKS.EAI_AUDIT_TRAIL where SRCE_APP_NAME='MF_HT_CustomerFlow' order by MSG_OUT_DATE desc fetch first 10 row only;
SELECT * FROM WMBRKS.EAI_AUDIT_TRAIL where SRCE_APP_NAME='MF_HT_BroadBand_ProvisioningConsumer' order by MSG_OUT_DATE desc fetch first 100 row only;

SELECT * FROM WMBRKS.EAI_AUDIT_TRAIL where SRCE_APP_NAME='MF_HT_BroadBand_New' order by MSG_OUT_DATE desc fetch first 100 row only;

SELECT * FROM WMBRKS.EAI_AUDIT_TRAIL where SRCE_APP_NAME='MF_HT_BroadBand_ProvisioningConsumer' and AUDIT_IDENTIFIER_VALUE='800115683194' order by MSG_OUT_DATE desc fetch first 100 row only;
```

==

JVM not found issue:- installer cannot find the correct version of Java in the standard directories of your system

SET Java home to the JDK or JRE as appropriate

===========Creating JDBC4typeconnection for Oracle=================

```
mqsideleteconfigurableservice MB7BROKER -c JDBCProviders -o SIMPLERROUTEDB


mqsicreateconfigurableservice MB7BROKER -c JDBCProviders -o SIMPLERROUTEDB

-n connectionUrlFormat,connectionUrlFormatAttr1,description,jarsURL,portNumber,

serverName,type4DatasourceClassName,type4DriverClassName

-v "jdbc:oracle:thin:[user]/[password]@[serverName]:[portNumber]:[connectionUrlFormatAttr1],

<SID>,Simplified Database Routing Sample Database,<JARS URL>,<PORT NUMBER>,

<SERVER NAME>,oracle.jdbc.xa.client.OracleXADataSource,oracle.jdbc.OracleDriver"
```

to check that the default broker registry was correctly updated with the sample SIMPLERROUTEDB JDBCProvider entry:

```
mqsireportproperties MB7BROKER -o SIMPLERROUTEDB -c JDBCProviders -r
```

If the update was successful, the reported property output from the command matches the following example:

```
JDBCProviders
  SIMPLERROUTEDB
    connectionUrlFormat='jdbc:oracle:thin:[user]/[password]@[serverName]:[portNumber]:[connectionUrlFormatAttr1]'
    connectionUrlFormatAttr1='orc1'
    connectionUrlFormatAttr2=''
    connectionUrlFormatAttr3=''
    connectionUrlFormatAttr4=''
    connectionUrlFormatAttr5=''
```

```
   databaseName='default_Database_Name'
   databaseType='default_Database_Type'
   databaseVersion='default_Database_Version'
   description='Simplified Database Routing Sample Database'
   environmentParms='default_none'
   jarsURL='C:\oracle\oraxx\jdbc\lib'
   portNumber='1521'
   securityIdentity='default_User@default_Server'
   serverName='localhost'
   type4DatasourceClassName='oracle.jdbc.xa.client.OracleXADataSource'
   type4DriverClassName='oracle.jdbc.OracleDriver'
```

Specify the user identifier and password to associate with the JDBC provider SIMPLERROUTEDB:

Use the mqsisetdbparms and mqsichangeproperties commands to specify a user identifier and password for the broker to use with JDBC provider SIMPLERROUTEDB. This user identifier must be the same user identifier that you used when you created the database.

a.Enter the following command to associate the user identifier and password with a security identity:

```
mqsisetdbparms MB7BROKER -n jdbc::mySecurityIdentity -u <user ID> -p <password>
```

b.Enter the following command to associate the security identity, which you defined in the preceding step, with the securityIdentity property of the JDBC provider SIMPLERROUTEDB:

mqsichangeproperties MB7BROKER -c JDBCProviders -o SIMPLERROUTEDB -n securityIdentity -v mySecurityIdentityc.You must stop and restart the broker for the changes to the JDBC provider to become available to a message flow that is already deployed. If you have already deployed the sample, stop and restart the broker.

========================

```
mqsireportproperties BKR2 -c JDBCProviders -o MBREPOS_D -r

mqsicreateconfigurableservice UNXS0458.MB1 -c JDBCProviders -o MBODS

mqsicreateconfigurableservice UNXS0458.MB1 -c JDBCProviders -o MBREPOS
```

```
mqsichangeproperties UNXS0458.MB1 -c JDBCProviders -o MBODS -n connectionUrlFormat,connectionUrlFormatAttr1,connectionUrlFormatAttr2,connectionUrlFormatAttr3,connectionUrlFormatAttr4,connectionUrlFormatAttr5,databaseName,databaseSchemaNames,databaseType,databaseVersion,description,environmentParms,jarsURL,jdbcProviderXASupport,maxConnectionPoolSize,portNumber,securityIdentity,serverName,type4DatasourceClassName,type4DriverClassName -v "jdbc:oracle:thin:[user]/[password]@[serverName]:[portNumber]:[connectionUrlFormatAttr1]","IBRS1","","","","","default_Dat
```

```
abase_Name","","Oracle","10.2.0.1.0","default_Description","default_none","/dmbu3/mqsi/config/UNXS0458.MB1/shared-classes/","true","10","16444","MBODSSecurityIdentity","unxs0190","oracle.jdbc.xa.client.OracleXADataSource","oracle.jdbc.OracleDriver"
```

```
mqsichangeproperties UNXS0458.MB1 -c JDBCProviders -o MBREPOS -n connectionUrlFormat,connectionUrlFormatAttr1,connectionUrlFormatAttr2,connectionUrlFormatAttr3,connectionUrlFormatAttr4,connectionUrlFormatAttr5,databaseName,databaseSchemaNames,databaseType,databaseVersion,description,environmentParms,jarsURL,jdbcProviderXASupport,maxConnectionPoolSize,portNumber,securityIdentity,serverName,type4DatasourceClassName,type4DriverClassName -v "jdbc:oracle:thin:[user]/[password]@[serverName]:[portNumber]:[connectionUrlFormatAttr1]","TMBU1","","","","","default_Database_Name","","Oracle","10.2.0.1.0","default_Description","default_none","/dmbu3/mqsi/config/UNXS0458.MB1/shared-classes/","true","10","1527","MBREPOSSecurityIdentity","unxs0376","oracle.jdbc.xa.client.OracleXADataSource","oracle.jdbc.OracleDriver"
```

```
mqsisetdbparms UNXS0458.MB1 -n jdbc::MBREPOSSecurityIdentity -u MBREPOS -p MBR3P0S

mqsisetdbparms UNXS0458.MB1 -n jdbc::MBODSSecurityIdentity -u CWODS -p cradmin2k11
```

==

---Configurable Item for SAPBWZABAPSSRFCINTERFACERFC.outadapter

(SAP BW RFC)

```
mqsicreateconfigurableservice UNXS0460.MB1  -c SAPConnection -o SAPBWZABAPSSRFCINTERFACERFC.outadapter
```

```
mqsichangeproperties UNXS0460.MB1  -c SAPConnection -o SAPBWZABAPSSRFCINTERFACERFC.outadapter -n applicationServerHost -v BQAIX3C

mqsichangeproperties UNXS0460.MB1  -c SAPConnection -o SAPBWZABAPSSRFCINTERFACERFC.outadapter -n client -v 102

mqsichangeproperties UNXS0460.MB1  -c SAPConnection -o SAPBWZABAPSSRFCINTERFACERFC.outadapter -n systemNumber -v 00

mqsichangeproperties UNXS0460.MB1  -c SAPConnection -o SAPBWZABAPSSRFCINTERFACERFC.outadapter -n connectionIdleTimeout -v 0

mqsisetdbparms UNXS0460.MB1  -n SAPBWZABAPSSRFCINTERFACERFC.outadapter -u MBROKER -p  kits1234
```

---Configurable Item for SAPR3IDOCINBOUND.inadapter

```
mqsicreateconfigurableservice UNXS0460.MB1  -c SAPConnection -o SAPR3IDOCINBOUND.inadapter

mqsichangeproperties UNXS0460.MB1  -c SAPConnection -o SAPR3IDOCINBOUND.inadapter -n gatewayService -v sapgw01

mqsichangeproperties UNXS0460.MB1  -c SAPConnection -o SAPR3IDOCINBOUND.inadapter -n applicationServerHost -v unxs0007

mqsichangeproperties UNXS0460.MB1  -c SAPConnection -o SAPR3IDOCINBOUND.inadapter -n client -v 900

mqsichangeproperties UNXS0460.MB1  -c SAPConnection -o SAPR3IDOCINBOUND.inadapter -n gatewayHost -v unxs0007
mqsichangeproperties UNXS0460.MB1  -c SAPConnection -o SAPR3IDOCINBOUND.inadapter -n systemNumber -v 01

mqsichangeproperties UNXS0460.MB1  -c SAPConnection -o SAPR3IDOCINBOUND.inadapter -n connectionIdleTimeout -v 0

mqsichangeproperties UNXS0460.MB1  -c SAPConnection -o SAPR3IDOCINBOUND.inadapter -n rfcProgramID -v DV7_BROKER_RFC

mqsisetdbparms UNXS0460.MB1  -n SAPR3IDOCINBOUND.inadapter -u MBROKER -p  Wmb@dm1n
```

------Other way--------------

```
mqsicreateconfigurableservice UNXS0460.MB1 -c SAPConnection -o SAPR3IDOCOUTBOUND.outadapter -n RFCTraceLevel,RFCTraceOn,RFCTracePath,SAPSystemID,applicationServerHost,assuredOnceDelivery,client,connectionIdleTimeout,gatewayHost,gatewayService,loadBalancing,logonGroup,messageServerHost,numberOfListeners,retryConnectionOnStartup,retryInterval,retryLimit,rfcProgramID,sharedTidStoreClientDefinitionFile,sharedTidStoreQmgr,systemNumber  -v "","","","","unxs0007","","902","0","unxs0007","sapgw01","","","","","","","","DV7_BROKER_RFC","","","01"


mqsisetdbparms UNXS0460.MB1  -n SAPR3IDOCOUTBOUND.outadapter -u MBROKER -p  Wmb@dm1n
```

===

=====Setting MQMD & MQRFH2 Header==============

```
SET OutputRoot.MQMD.Encoding = 273;

SET OutputRoot.MQMD.CodedCharSetId = 819;

SET OutputRoot.MQMD.Format = MQFMT_RF_HEADER_2;


SET OutputRoot.MQMD.MsgType = MQMT_DATAGRAM;

SET OutputRoot.MQMD.Persistence = MQPER_PERSISTENT;

SET OutputRoot.MQMD.Expiry = MQEI_UNLIMITED;


CREATE NEXTSIBLING OF OutputRoot.MQMD DOMAIN 'MQRFH2';

SET OutputRoot.MQRFH2.(MQRFH2.Field)Version = 2;

SET OutputRoot.MQRFH2.(MQRFH2.Field)Format = 'MQSTR';

SET OutputRoot.MQRFH2.(MQRFH2.Field)Encoding = 273;

SET OutputRoot.MQRFH2.(MQRFH2.Field)CodedCharSetId = 819;


SET OutputRoot.MQRFH2.usr.test='scm';

SET OutputRoot.MQRFH2.usr.test2=CURRENT_GMTTIMESTAMP;

SET OutputRoot.MQRFH2.psc.Command              = 'Publish';
SET OutputRoot.MQRFH2.psc.Topic	= '/TEST/MQRFSSS/TESTI';
```

=====SOA Principles implemented==========

Principles, in order of preference are;

Abstraction -expose only the required functionality, control access to service documentation and utilise our integration architecture to encapsulate legacy applications

Contract Standardisation - maintain data (XSD), policy and contract (WSDL) separately, define contracts prior to implementation

Loose Coupling -enforce contract centralisation, maximise logic-to-contract coupling

Reuse - Use a service registry to enable re-use through discoverability

Service Statelessness -build short-lived, stateless, services wherever possible. Allow for the storage of configuration information only

Discoverability -use design time discoverability as our primary means of discovery. Use standardised service contracts as an aid to discovery. Create a repository of services

Autonomy – acknowledge that our services can not be entirely autonomous due to their reliance of Enterprise systems such as SAP

===

1) Edit odbc64.ini(brokers_install_dir/ODBC64.ini)

2) mesisetdbparam command

3)

odbc64.ini sample file (ah14452_)

The correct content for HP-UX (PA-RISC platform) is:

```
############################################################

## 64 bit ODBC database driver manager initialisation file. ##

############################################################

# It is recommended that you take a copy of this file and #

# then edit the copy. #

# #

# 1. Complete the 'Mandatory information stanza' section #

# at the end of the file. #
# #

# 2. For each data source, add the name of the data source #

# into the 'List of data sources stanza' section. #

# #

# 3. For each data source, create a stanza in the #

# 'Individual data source stanzas' section. #

##############################################################


#######################################

#### List of data sources stanza ######

#######################################

[ODBC Data Sources]

DB2DB=IBM DB2 ODBC Driver

ORACLEDB=DataDirect 5.0 64-bit Oracle Wire Protocol

ORACLERACDB=DataDirect 5.0 64-bit Oracle Wire Protocol (Real Application Clusters)

SYBASEDB=DataDirect 5.0 64-bit Sybase Wire Protocol

SYBASEDBUTF8=DataDirect 5.0 64-bit Sybase UTF8 Driver

INFORMIXDB=IBM Informix ODBC Driver


##########################################################

########## Individual data source stanzas ################

##########################################################

# DB2 stanza

[DB2DB]

Driver=libdb2Wrapper64.sl

Description=DB2DB DB2 ODBC Database

Database=DB2DB

# Oracle stanza

[ORACLEDB]

Driver=<Your install directory>/ODBC64/V5.0/lib/UKora20.sl

Description=DataDirect 5.0 Oracle Wire Protocol

HostName=<Your Oracle Server Machine Name>

PortNumber=<Port on which Oracle is listening on HostName>
```

```
SID=<Your Oracle SID>
CatalogOptions=0
ProcedureRetResults=1
EnableStaticCursorsForLongData=0
ApplicationUsingThreads=1
EnableDescribeParam=1
OptimizePrepare=1
WorkArounds=536870912
ColumnSizeAsCharacter=1
# Oracle Real Application Clusters stanza
[ORACLERACDB]
Driver=<Your Broker install directory>/ODBC64/V5.0/lib/UKora20.sl
Description=DataDirect 5.0 64bit Oracle Wire Protocol
HostName=<Your Oracle Server Machine Name>
PortNumber=<Port on which Oracle is listening on HostName>
ServiceName=<Your Oracle RAC Net Service Name>
CatalogOptions=0
EnableStaticCursorsForLongData=0
ApplicationUsingThreads=1
EnableDescribeParam=1
OptimizePrepare=1
WorkArounds=536870912
ProcedureRetResults=1
ColumnSizeAsCharacter=1
# Sybase stanza
[SYBASEDB]
Driver=<Your Broker install directory>/ODBC64/V5.0/lib/UKase20.sl
Description=DataDirect 5.0 Sybase Wire Protocol
Database=<Your Database Name>
ServerName=<YourServerName>
EnableDescribeParam=1
OptimizePrepare=1
SelectMethod=0
```

```
NetworkAddress=<YourServerName>,<YourPortNumber>
SelectUserName=1
# Sybase Stanza for a UTF8 datasource
[SYBASEDBUTF8]
Driver=<Your Broker install directory>/ODBC64/V5.0/lib/UKase20.sl
Description=DataDirect 5.0 64bit Sybase Wire Protocol
Database=<Your Database Name>
ApplicationUsingThreads=1
EnableDescribeParam=1
OptimizePrepare=1
SelectMethod=0
NetworkAddress=<Your Sybase Server Name>,<Your Sybase Port Number>
SelectUserName=1
ColumnSizeAsCharacter=1
Charset=UTF8
# Informix stanza
[INFORMIXDB]
Driver=libinfWrapper64.sl
Description=IBM Informix ODBC Driver
ServerName=<Your Informix Server Name>
Database=<Your Database Name>

##########################################
###### Mandatory information stanza ######
##########################################
[ODBC]
Trace=0 # To turn on ODBC trace set Trace=1
TraceFile=<A Directory with plenty of free space>/odbctrace.out
TraceDll=<Your Broker install directory>/ODBC64/V5.0/lib/odbctrac.sl
InstallDir=<Your Broker install directory>/ODBC64/V5.0
UseCursorLib=0
IANAAppCodePage=4
UNICODE=UTF-8
```

==============================

OK, here it is:

1.Log on as root.

2.Enter the following commands to create a file called user.log.

On Linux systems, enter the command:

```
touch /var/mqsi/user.log

chown root:mqbrkrs /var/mqsi/user.log

chmod 640 /var/mqsi/user.log
```

3.Add the following line to the /etc/syslog.conf file to redirect debug level messages to the file user.log:

On Linux, enter :

```
user.info\t/var/mqsi/user.log
```

where “\t” is the tab character.

4.Restart the syslog daemon.

On Linux, enter the command:

```
/etc/init.d/syslog restart
```

or

```
/etc/rc.d/init.d/syslog restart
```

for systems where rc.d is not a soft link

For other syslog options, see the documentation for your operating system.

==

```
cat wmbevent.log |grep "Jul  8" | grep "user:err|error" | more

cat wmbevent.log |grep "May  2" | grep " FulfilmentSourceService" | more

cat wmbevent.log |grep "2013-01-19T09” |more

C:\>mqsireportproperties BKR3 -o ComIbmJVMManager  -e TEST4 -a

C:\>mqsireportproperties BKR3 -o ComIbmXmlParserFactory  -e TEST4 -a

C:\>mqsireportproperties BKR3 -o AllReportableEntityNames -e TEST4 -a

C:\>mqsireportproperties BKR3 -o ComIbmJVMManager  -e TEST4 –r

mqsibrowse UNXS0460.MB1 -t BROKERAAEG

fteListAgents

fteListAgents -v

ftePingAgent MFTHRSAXPPRD

fteShowAgentDetails MFTHRSXAPPRD
fteShowAgentDetails -v MFTHRSAPPRD

fteStartAgent MFTETLUXATZ

fteStopAgent MFTATGBCCPRXF

fteStopAgent -i MFTHRSAPPRX  //force fully stop

fteCreateAgent -agentName MFTHRSX -agentQMgr UNXS0393.AGT.QM1 –f

fteCreateMonitor -ma MFTHRSAPPRD -mm UNXS039X.AGT.QM1 -mn ServiceNowPRDMonitor -md
/var/mqsi/tempdata/saphrmqmft

/HRMQFT/ServiceNow -mt /var/mqm/scripts/MFTHRSAPPRDScripts/ServiceNowPRDTemplate.xml -pi 5 -pu minutes -tr
match,"*.csv"


fteDeleteAgent MFTHRSAPPXD

fteDeleteMonitor -ma MFTHRSAPPRXD -mm UNXS0393.AGT.QM1 -mn ServiceNowPRDMonitor
```

===

==WS-A======================

WS-Addressing provides transport-neutral mechanisms to address Web services and messages.

Web Services Addressing (WS-Addressing) defines two interoperable constructs that convey information that is typically provided by transport protocols and messaging systems.

The two constructs are endpoint(resource where Web service messages can be targeted) references and message information headers.

sample:

```
(001) <S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"

              xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing">

(002)   <S:Header>

(003)    <wsa:MessageID>

(004)      uuid:6B29FC40-CA47-1067-B31D-00DD010662DA

(005)    </wsa:MessageID>

(006)    <wsa:ReplyTo>

(007)      <wsa:Address>http://business456.example/client1</wsa:Address>

(008)    </wsa:ReplyTo>

(009)    <wsa:To>http://fabrikam123.example/Purchasing</wsa:To>

(010)    <wsa:Action>http://fabrikam123.example/SubmitPO</wsa:Action>

(011)   </S:Header>

(012)   <S:Body>
```

```
(013)     ...
(014)   </S:Body>
(015) </S:Envelope>
```

=======================Default:HTTP & SOAP-HTTP ports==MB6.1 & 7==================================

7080 - all unencrypted traffic to all HTTPInput nodes in both ExecutionGroups

7083 - all encrypted traffic to all HTTPInput nodes in both ExecutionGroups

7800 - all unencrypted traffic for SOAPInput nodes deployed to the first ExecutionGroup that starts up

7801 - all unencrypted traffic for SOAPInput nodes deployed to the second ExecutionGroup that starts up

7843 - all encrypted traffic for SOAPInput nodes deployed to the first ExecutionGroup that starts up

7844 - all encrypted traffic for SOAPInput nodes deployed to the second ExecutionGroup that starts up

==

============================SET-IV============================

------------To get username used for a data source--

```
mqsiservice D_SOA_BRK_01 -r DataSourceName=INTDEV
```

------------To get username used for a data source (registry path)--

```
/var/mqsi/registry/D_MAXIS_BRK_01/CurrentVersion
```

------------To change broker data source password--

```
mqsichangebroker P_SOA_BRK_01 -i soaadm -a soaadm5678 -p sdp12345
```

------------To change applicaton data source password--

```
mqsisetdbparms P_SOA_BRK_01 -a -n INTPCAT -u ebizprod -p ebizprod
```

------------To create EG--

```
mqsicreateexecutiongroup -i 127.0.0.1 -q PSOACFG1 -p 1415  -b P_SOA_BRK_01 -e EG_SOA -l
```

------------To check EG port--

```
mqsireportproperties P_SOA_BRK_01 -e EG_SOA -o HTTPConnector -n port
mqsireportproperties S_MAXIS_BRK_01 -e EG_WS -o HTTPSConnector -n port
```

------------To change EG port--

```
mqsichangeproperties P_SOA_BRK_01 -e EG_SOA -o HTTPConnector -n explicitlySetPortNumber -v 9001
```

------------To reload EG--

```
mqsireload P_SOA_BRK_01 -e EG_SOA
```

------------To deploy message flow--

```
mqsideploy -i 127.0.0.1 -q PSOACFG1 -p 1415 -b P_SOA_BRK_01 -e EG_SOA -a SimLineEligibilityCheck_PRD_V1.0.bar
```

------------To start message flow--

```
mqsistartmsgflow -q PSOACFG01 -i 127.0.0.1 -p 1415 -b P_SOA_BRK_01 -e EG_SOA_BATCH_1 -m MF_SOA_EVENTS_INTDB
```

------------To stop message flow--

```
mqsistopmsgflow -q PSOACFG01 -i 127.0.0.1 -p 1415 -b P_SOA_BRK_01 -e EG_SOA_BATCH_1 -m MF_SOA_EVENTS_INTDB
```

------------To delete message flow / message set--

```
mqsideploy -i 127.0.0.1 -q PSOACFG1 -p 1415 -b P_SOA_BRK_01 -e EG_SOA_BATCH_1 -d
MF_ES_SRVCSTS_OTM.cmf:MF_SRVCSTS_OTM_Batch_File.cmf:MS_SP_OTM_CSV.dictionary:MS_SP_OTM_WSDL.xsdzip -v
PRD_ES_OTM_P_V1.0.1Del_Rollback.log -w 720
```

------------To start broker--

```
mqsistart P_SOA_BRK_01
```

------------To stop broker--

```
mqsistop P_SOA_BRK_01
```

------------To check active ports--

```
netstat -an | grep LISTEN
```

------------Path for ODBC64--

```
/opt/mqsi/6.1/ODBC64/V5.3
```

------------Path for registry--

```
/var/mqsi/registry/D_MAXIS_BRK_01/CurrentVersion
```

------------Path for DB registry--

```
/var/mqsi/registry/D_SOA_BRK_01/CurrentVersion/DSN
```

------------Path for syslog--

```
/var/adm
```

------------Path for broker error log--

```
/var/mqsi/errors
```

------------Path for QM error log--

```
/var/mqm/qmgrs/<QM>/errors
```

------------To check for listening ports--

```
netstat -an | grep LISTEN
```

------------To remove ^M in Unix--

Using the following in VI editor (if the file is small). Note that ^M is typed as Ctrl + V + M

```
:%s/^M//g
```

------------To kill an EG--

1. ps -eaf | grep DataFlowEngine

2. kill -9 <process id>

3. sample: kill -9 381108

-------------To DELETE records from EAI_AUDIT_TRAIL, older than 3 months:

```
DELETE FROM WMBRKS.EAI_AUDIT_TRAIL WHERE MSG_IN_DATE < CURRENT_TIMESTAMP - 3 MONTH
```

================================SET-V===================

ATG command :

```
runAssembler sample.ear -m PioneerCyclingJSP DafEar.Admin
```

java weblogic.DDConverter -d . sample.ear --It will read the EAR file and outside of the EAR application it will generate the fresh XML files.

```
netstat -a | find "LISTENING"
```

```
mqsisetdbparms MB7BROKER -n TESTBKDB -u db2admin -p db2admin200

mqsisetdbparms D_SOA_BRK_01 -n INTDEV -u wmbrks -p wmbrks -a // '-a' option: incase of MB 6.1.0.8 will work when broker is running

mqsisetdbparms L_BROKER1_MAXISADM -n L_BRKDB1 -u maxisadm -p maxisadm11 // not possible to change uid and pass for Broker DB.

mqsisetdbparms UNXS0231.MB1 -n MBREPOS -d --deleted all previously created identifiers
```

```
mqsicvp MB7BROKER -n MBREPOS -v // Test Database connectivity from The broker , shows all details of the DSN to use by the broker

mqsicvp -n MBREPOS -u MBREPOS -p MBREPOS
```

```
mqsichangebroker L_BROKER1_MAXISADM -i maxisadm -a maxisadm11 -p maxisadm11  // in case service password been changed, use it
```

```
mqsireload D_SOA_BRK_01 -e EG_InternetHotticket
```

```
mqsireportproperties MB7BROKER -e EG_CommonServices -o HTTPSConnector -n port // Show the port number for the execution group

mqsireportproperties MB7BROKER -c JDBCProviders -o INF_ECOMM_INT -r  //jdbcProviders details

mqsireportproperties MB7BROKER -c JDBCProviders -a -o AllReportableEntityNames //to view the list of available JDBCProvider services

mqsireportproperties DEV_BRKR1 -c JMSProviders -a -o AllReportableEntityNames

mqsireportproperties DEV_BRKR1 -c AllTypes -o AllReportableEntityNames -r
```

```
mqsiservice -v  // product version

mqsiservice DEV_BRK5 -t  /time details
```

```
mqsicreatebroker MB7BROKER -i testUser -a pwdtest -q MB7QMGR
```

```
mqsideletebroker MB7BROKER -q


sc delete MQSeriesBrokerMB7BROKER //delete service from window service list


================upto MB6.1=================================


mqsicreateconfigmgr CONFIG_MANAGER_MAXISADM -i maxisadm -a maxisadm -q QM.MAXISADM

mqsicreatedb L_BRKDB1 -i maxisadm -a maxisadm -e DB2  //ODBC Data Source name 'L_BRKDB1' created succesfully

mqsicreatebroker L_BROKER1_MAXISADM -i maxisadm  -a maxisadm -q QM.MAXISADM -n L_BRKDB1

mqsideleteconfigmgr CONFIG_MANAGER_MAXISADM

mqsideletedb L_BRKDB1

mqsideletebroker L_BROKER1_MAXISADM

==========================================================

mqsilist

mqsistart CONFIG_MANAGER_MAXISADM

mqsistart L_BROKER1_MAXISADM

=================================

mqsichangetrace L_BROKER1_MAXISADM -u -e Test  -l debug -r -c 50000

mqsichangetrace L_BROKER1_MAXISADM -u -e Test  -l none

mqsireadlog L_BROKER1_MAXISADM -u -e Test -f -o UMB_HT1.xml

mqsiformatlog -i UMB_HT1.xml -o UMB_HT2.txt

notepad UMB_HT2.txt

=====================================================

mqsichangetrace UNXS0231.MB1 -u -e DarwinTest  -k Location -f com.kingfisher.ukbq.LocationService.LocationService -l debug
-c 5000 -r


mqsichangetrace UNXS0231.MB1 -u -e DarwinTest -k Location -f com.kingfisher.ukbq.LocationService.LocationService -l none

mqsireadlog UNXS0231.MB1  -u -e DarwinTest -o trace2.xml

mqsiformatlog -i trace2.xml -o formattraceSync.log


=================

mqsireportproperties D_SOA_BRK_01 –e EG_InternetHotticket -o HTTPConnector -n explicitlySetPortNumber  -v 9003
```

```
mqsichangebroker L_BROKER1_MAXISADM -l "C:\lilfolder"

mqsireportbroker  L_BROKER1_MAXISADM //shows Broker's Install path,Work path,Broker UUID,Process id,Queue Manager,User Name Server Queue Manager,Broker database name,Broker database userId,Broker database password,User lil path,User exit path,Active user exits,LDAP principal,LDAP credentials,ICU converter path,HTTP listener port,Pubsub migration,Pubsub access control,Trusted (fastpath) Queue Manager application,Configuration change timeout,Internal configuration timeout,Statistics major interval,Operation mode,Fixpack capability level,Broker registry format

mqsichangeflowmonitoring DEV_BROKER2 -e DEV_EG_DEFAULT -f testevtFlow -s "MQ Input.TransactionStart" -i enabled

mqsichangeflowmonitoring DEV_BROKER2 -e DEV_EG_DEFAULT -k TestEvent -j  //allmessage Flow

mqsichangeflowmonitoring DEV_BRK5 -g -j -c active  //all eg and all msgflow

========================================================

mqsicreateexecutiongroup -i 172.21.33.65 -q DSOABRK1 -b D_SOA_BRK_01 -e EG_InternetHotticket -l -v Test_EG_InternetHotticket.log

START/STOP - wmb configuration manager : mqsistart D_SOA_CFG_01

mqsistop D_SOA_CFG_01


mqsicreateexecutiongroup -i 127.0.0.1 -q DSOACFG1 -b D_SOA_BRK_01 -e EG_InternetHotticket -l -v Test_EG_InternetHotticket.log

mqsicreateexecutiongroup -i 127.0.0.1 -p 1414 -q QM.MAXISADM -b L_BROKER1_MAXISADM -e EG_InternetHotticket -l

mqsideleteexecutiongroup -i 127.0.0.1 -p 1414 -q QM.MAXISADM -b L_BROKER1_MAXISADM -e EG_InternetHotticket

DEFINE ql(CIBCHARGING.MSG) maxdepth(999999)

DEFINE ql(PROVISIONING.INCOMPLETE) maxdepth(999999)


mqsideploy -i 127.0.0.1 -q DSOACFG1 -p 2415 -b D_SOA_BRK_01 -e EG_InternetHotticket -a Test_ProvisionConsumerService.bar  -v Test_ProvisionConsumerService.log -w 720

mqsideploy -i 127.0.0.1 -q QM.MAXISADM -p 1414 -b L_BROKER1_MAXISADM -e Test -d MF_AsynRequest.cmf

ps -ef | grep runmqlsr // know the quuemaneger's listners details

DISPLAY LISTENER(*) ALL //shows all listner of MQ



================================MQ================================

dltmqm MB7QMGR

dspmqver //know MQ version

crtmqm -c "local queue" -ll -q QM.MAXISADM

strmqm QM.MAXISADM
```

```
amqmdain reg QM.MAXISADM -c  display -s Log -v * // display log setting

dspmqfls -m QM.MAXISADM -t qlocal *  //display queue filenames for queues

------------------runmqsc----------------------------

display q(*) where (usage eq XMITQ)

DIS QMGR               // display qm setting

DIS Q(*) CURDEPTH WHERE(CURDEPTH GT 0)  // display non empty queues

DIS CHL(*) ALL  //Display channels,name & type

DIS CHL(*) CONNAME //Display channels with specific atributes

DIS SERVICE(*) ALL  // Display service name & types all attributes

DIS LSTR(*) ALL  // Display Listners name & types all attributes

-----------------------------------------
```

```
mqsilist BRK_01 -e EG // list of deployed flows
```

```
mqsichangetrace L_BROKER1_MAXISADM -n off -e Test  -r //changes Trace node's setting to OFF
```

```
mqsichangeproperties DEV_BROKER1 -c EISProviders -o SAP -n jarsURL,nativeLibs -v C:\SAP_JCO
```

===========mqsideploybaroverride interim solution in MB8.0.0.0.0=================================

Create a configuration file called mbconfig.properties at some path with the following lines (with the proper values inserted):

```
STEPURL=path to the STEP web service for the tier

com.kingfisher.ukbq.ProductSetService.ProductSet_STEP_Sub#USERNAME=STEP User ID

com.kingfisher.ukbq.ProductSetService.ProductSet_STEP_Sub#PASSWORD=STEP Password
```

E.g.

```
STEPURL=http://localhost:7800/step/ws

com.kingfisher.ukbq.ProductSetService.ProductSet_STEP_Sub#USERNAME=mbsuperuser

com.kingfisher.ukbq.ProductSetService.ProductSet_STEP_Sub#PASSWORD=mbsuperuser
```

4.1.2.1 Deployment of the .Bar file

There are some properties in the .Bar file that need to be overridden with tier-specific values before deployment.

Perform the following steps whenever you build the .Bar file:

- Extract the bar file (with a zip file extractor program like WinZip) contents into a directory
- Run the following command from MQSI command console with the

o mqsiapplybaroverride -b “<appzip file path>/ProductSetService.appzip” -p "<config file name with complete path>/mbconfig.properties"

o Here <appzip file path> should be replaced by the path where the contents of the bar file are extracted (including the file ProductSetService.appzip); and "<config file name with complete path> should be replaced with the path where mbconfig.properties has been created

- Now add the ProductService.appzip file back to the .Bar file

The .Bar file can now be deployed.

===========

tracert/traceroute <IP> <Port>

RFUTILS.SVRCONN/TCP/172.19.174.41(1430) --to connect to remote QM from local rfhutilc

Broker's home from where mqsc command can be run: /opt/IBM/mqsi/8.0.0.0/bin

Odbc path: /mbud1/mqsi/odbc

===

SSL over HTTP:

=================

Create a keystore file to store the broker's certificates

Configuring the broker to use SSL on a particular port

Creating a message flow to process HTTPS requests

Testing your example

=================

```
"C:\Program Files\IBM\MQSI\7.0\jre16\bin\keytool" -genkey -keystore myTestKeyStoreFile -storepass db2admin200 -alias myTestKey
```

Choose the keystore file to be used, by setting a value for keystoreFile

```
mqsichangeproperties BROKER_TEST1 -b httplistener -o HTTPSConnector -n keystoreFile -v "C:\Program Files\IBM\MQSI\7.0\jre16\bin\myTestKeyStoreFile"
```

Specify the password for the keystore file, by setting a value for keystorePass

```
mqsichangeproperties BROKER_TEST1 -b httplistener -o HTTPSConnector -n keystorePass -v db2admin200
```

Specify the port on which WebSphere Message Broker will listen for HTTPS requests

```
mqsichangeproperties BROKER_TEST1 -b httplistener -o HTTPSConnector -n port -v 1418
```

Turn on SSL support in message broker, by setting a value for enableSSLConnector

```
mqsichangeproperties BROKER_TEST1 -b httplistener -o HTTPListener  -n enableSSLConnector -v true
```

create a flow only having "HTTPInput" and "HTTPReply" node

Test in IE,Mozila:

https://localhost:1418/testHTTPS

==

Extracting a certificate from another keystore:

```
"C:\Program Files\IBM\MQSI\7.0\jre16\bin\keytool" -export -alias myTestKey -file myTestCert -keystore myTestKeyStoreFile -storepass db2admin200
```

Importing a certificate into the cacerts file:

```
"C:\Program Files\IBM\MQSI\7.0\jre16\bin\keytool" -import -alias myTestKey -file myTestCert -keystore cacerts -storepass changeit
```

==

If you need DTD support, the answer (for now) is to use XMLNS not the XMLNSC.DTD support in XMLNSC parser is limited.

=========================

```
mqsichangetrace L_BROKER1_MAXISADM -u -e Test -f MF_TemService -l debug -r -c 50000

mqsichangetrace L_BROKER1_MAXISADM -u -e Test -f MF_TemService -l none

mqsireadlog L_BROKER1_MAXISADM -u -e Test -f -o INE.xml

mqsiformatlog -i INE.xml -o koyel.txt

notepad INE.xml
```

```
================================Configuring HTTP Proxy servlet============================

$mqsichangeproperties L_BROKER1_MAXISADM -e Test -o HTTPConnector -n enableMQListener -v true

$mqsireportproperties L_BROKER1_MAXISADM -e Test -o HTTPConnector -n enableMQListener

===========================
```

Call different(from which is specified at node) DSN from ESQL:

```
SET admqry = 'SELECT sysdate from dual;';

SET sDate.Rows[] = PASSTHRU(admqry TO Database.{ADMDB});
```

```
========================================================================
```

stop broker before mqsisetdbparms incase of MB 6.1.0.4 else(in case of 6.1.0.8 use -a option, no restart)

```
=============DB2 SQL command============

select * from WMBRKS.EAI_ERRORLOG order by MSG_OUT_DATE desc fetch first 10 row only;

select * from WMBRKS.EAI_AUDIT_TRAIL order by MSG_OUT_DATE desc fetch first 10 row only;

=========================

CREATE FIELD OutputRoot.XMLNSC.payload.Body.Order TYPE Name;

CREATE FIRSTCHILD OF OutputRoot.XMLNSC.payload.Body.Order Domain('XMLNSC') NAME 'QueueName' VALUE 'out2';

========================================================================================

$ ps -ef | grep runmqchl/runmqlsr
```

dis chs(*) all -this will show you the process ID of the receiver channel - it's in JOBNAME (this is usually process-id/thread id, in hex).

Try a STOP CHL(rcvr chl name) MODE(FORCE) on the remote end.

Wait until all the channels say STOPPED (not STOPPING), then do a START CHL(chl name) on both the remote and local side (i.e. start the channel at both ends).

On Queue Manager: SSOABRK1

```
--------------------------
```

```
 DEFINE QL(QL.GEN.INSL.BRK.RES.02) maxdepth(999999)

 DEFINE QL(QL.GEN.INSL.BRK.RES.01) maxdepth(999999)


 DEFINE QREMOTE(QR.CIBCHR.BRK.INSL.REQ.01) DESCR('Queue for CIB charging reqest') RNAME(QL.CIBCHR.BRK.INSL.REQ.01)
RQMNAME(SSOAIN1) XMITQ(SSOAIN1)


1. STOP CHL(SSOAIN1.SSOABRK1.01) MODE(FORCE) -- receiver

4. STOP CHL(SSOABRK1.SSOAIN1.01) MODE(FORCE) -- sender


5. START CHL(SSOABRK1.SSOAIN1.01) --sender

8. START CHL(SSOAIN1.SSOABRK1.01) --receiver
```

On Queue Manager: SSOAIN1

```
DEFINE QL(QL.CIBCHR.BRK.INSL.REQ.01) maxdepth(999999)

DEFINE QREMOTE(QR.GEN.INSL.BRK.RES.02) DESCR('Queue for CIB charging response') RNAME(QL.GEN.INSL.BRK.RES.02)
RQMNAME(SSOABRK1) XMITQ(SSOABRK1)


2. STOP CHL(SSOABRK1.SSOAIN1.01) MODE(FORCE) -- receiver

3. STOP CHL(SSOAIN1.SSOABRK1.01) MODE(FORCE) -- sender


7. START CHL(SSOAIN1.SSOABRK1.01) -- sender

6. START CHL(SSOABRK1.SSOAIN1.01) -- receiver
```

==

```
crtmqm -c "SOA Adapter QM" -lc PSOAADP1 //To create queue manager

strmqm PSOAADP1                                    //To start queue manager

stpmqm PSOAADP1                                            //To stop queue manager

runmqsc PSOAADP1 < WMQ_Script_PRD_PSOAADP1_SMS.txt >> WMQ_Script_PRD_PSOAADP1_SMS.log //To run mq scripts in
batch

start chinit initq(SYSTEM.CHANNEL.INITQ) OR start chinit //start the channel initiator from RUNMQSC (Windows NT, UNIX and
OS/2). The

start runmqlsr -t tcp -m QMA -p 1414

start runmqchi                                     //use the channel initiator to start channels
```

```
runmqlsr -t tcp -m MB7QMGR -p 1414 /start TCP MQ listner
```

The following are to be run in runmqsc.

```
DEFINE LISTENER(PSOAADP1) TRPTYPE(TCP) CONTROL(QMGR) PORT(1416) //To define listener

START LISTENER(PSOAADP1)                                  //To start listener

DEFINE CHL(PSOAADP1.SVRCONN.01) CHLTYPE(SVRCONN) MCAUSER ('mqm') //To define server connection channel

DEFINE CHL(PSOABRK1.PSOAADP1.01) CHLTYPE(RCVR)                              //To define receiver channel

DEFINE QL(PSOABRK1) USAGE(XMITQ) TRIGDATA(PSOAADP1.PSOABRK1.01) INITQ(SYSTEM.CHANNEL.INITQ) //To define transmission queue

DEFINE CHL(PSOAADP1.PSOABRK1.01) CHLTYPE(SDR) TRPTYPE(TCP) CONNAME('172.16.136.27(1414)') XMITQ(PSOABRK1) //To define sender channel

START CHL(PSOAADP1.PSOABRK1.01)                                             //To start channel

DEFINE QL(QL.SMS.SOA.RES.00) DEFPSIST(YES) MAXDEPTH(9999) DESCR(Rpl) //To define local queue

DEFINE QR(QR.SMS.SOA.REQ.01) RNAME(QL.SMS.SOA.REQ.01) RQMNAME(PSOAADP1) XMITQ(PSOAADP1) //To define remote queue
```

==
======================

----Process Triggering--------

1. Create an performance event when queue reaches 80%

```
ALTER QMGR PERFMEV(ENABLED)

ALTER QLOCAL('MYQUEUE') QDEPTHHI(80) QDPHIEV(ENABLED)
```

2. Define a trigger when a msg put into the queue and it calls for a process definition

```
DEFINE QLOCAL(TRIG.EXAMPLE.QLOCAL) +

DESCR('Example Queue for Triggering') +

DEFPRTY(0) +

DEFSOPT(SHARED) +

GET(ENABLED) +

MAXDEPTH(5000) +

MAXMSGL(4194304) +

MSGDLVSQ(PRIORITY) +

PUT(ENABLED) +
RETINTVL(999999999) +
TRIGTYPE(EVERY) +
PROCESS(TRIG.EXAMPLE.PROCESS) +
INITQ(TRIG.EXAMPLE.INITQ) +
USAGE(NORMAL) +
REPLACE
```

3. Process definition calls an application script written in .bat,.sh,etc

```
DEFINE PROCESS (TRIG.EXAMPLE.PROCESS) +
DESCR('Example Process for Triggering') +
APPLTYPE(UNIX) +
APPLICID(/MQ/emailMessage.sh) +
REPLACE
```

5. runmqtrm -m QMNAME -q TRIG.EXAMPLE.INITQ &

4. Script sends an email alert.

```
--------------------------------------
DEFINE QLOCAL('QL.TRIGGER_MSGS') REPLACE
DEFINE QLOCAL('TRIG.EXAMPLE.QLOCAL') +
    TRIGGER +
    TRIGTYPE (EVERY) +
    INITQ('QL.TRIGGER_MSGS') +
    PROCESS('TRIG.EXAMPLE.PROCESS') +
    REPLACE
```

win:

```
DEFINE PROCESS('TRIG.EXAMPLE.PROCESS') +
    USERDATA('PARAM1 PARAM2') +
    APPLICID('D:\var\TriggerAppByMQ.bat') REPLACE
```

```
win:
DEFINE PROCESS('TRIG.EXAMPLE.PROCESS') +
    USERDATA('') +
    APPLICID('D:\var\test.bat') REPLACE
---------------------------------------------------
unix:
DEFINE PROCESS('TRIG.EXAMPLE.PROCESS') +
    USERDATA('PARAM1 PARAM2') +
    APPLICID('touch /support/home/wmbtcsdv/Trace/testii.txt') REPLACE



//start runmqtrm  -m UNXS0231.MB.QM1  -q QL.TRIGGER_MSGS
//runmqtrm -m DEV_QM5 -q QL.TRIGGER_MSGS
//runmqtrm -m UNXS0231.MB.QM1 -q QL.TRIGGER_MSGS


///usr/bin/runmqtrm
win:
DEFINE SERVICE('TRIG_MON_START') +
    CONTROL(QMGR) +
    SERVTYPE(SERVER) +
    STARTCMD('C:\Program Files\IBM\WebSphere MQ\bin\runmqtrm.exe') +
    STARTARG('-m +QMNAME+ -q "QL.TRIGGER_MSGS"') +
    STOPCMD('C:\Program Files\IBM\WebSphere MQ\bin\amqsstop.exe') +
    STOPARG('-m +QMNAME+ -p +MQ_SERVER_PID+') +
    STDERR('C:\Program Files\IBM\WebSphere MQ\errors\outerror.log') +
    STDOUT('C:\Program Files\IBM\WebSphere MQ\errors\output.log') +
    REPLACE
unix:
DEFINE SERVICE('TRIG_MON_START') +
    CONTROL(QMGR) +
    SERVTYPE(SERVER) +
    STARTCMD('/usr/bin/runmqtrm') +
    STARTARG('-m +QMNAME+ -q QL.TRIGGER_MSGS') +
    STOPCMD('/usr/bin/amqsstop') +

    STOPARG('-m +QMNAME+ -p +MQ_SERVER_PID+') +

    REPLACE
```

==================

Queries for Audit Logging:

==============

```
   INSERT INTO WMBRKS.MBRK_AUDIT_CONTROL

   (PROJECT_NAME, FLOW_NAME, AUDIT_IN, AUDIT_OUT, AUDIT_ERR, AUDIT_ERR_LOG, AUDIT_IN_QNAME,
AUDIT_OUT_QNAME, AUDIT_ERR_QNAME, AUDIT_ERR_LOG_QNAME, PAYLOADREQ, LAST_UPDATED, REMARKS)

 VALUES

   ('INTERNET_HOTTICKET', 'MF_HT_BroadBand_ProvisioningConsumer', 'Y', 'Y', 'Y', 'Y', 'QL.MBRK.AUDIT.INMSG.01',
'QL.MBRK.AUDIT.OUTMSG.01', 'QL.MBRK.ERR.01', 'QL.MBRK.ERR.LOG.01', 'Y', CURRENT_DATE, 'AbhijitK');
```

==============

```
UPDATE MBRK_AUDIT_CONTROL SET AUDIT_IN = 'N', AUDIT_OUT = 'N' WHERE PROJECT_NAME =
'MF_HT_BroadBand_ProvisioningConsumer';

COMMIT;

DELETE FROM WMBRKS.MBRK_AUDIT_CONTROL WHERE FLOW_NAME = 'MF_HT_BroadBand_ProvisioningConsumer';

SELECT * from WMBRKS.MBRK_AUDIT_CONTROL where FLOW_NAME='MF_HT_CustomerFlow';
```

===

JVM not found issue:- installer cannot find the correct version of Java in the standard directories of your system

SET Java home to the JDK or JRE as appropriate

===========Creating JDBC4typeconnection for Oracle==============

```
mqsideleteconfigurableservice MB7BROKER -c JDBCProviders -o SIMPLERROUTEDB


mqsicreateconfigurableservice MB7BROKER -c JDBCProviders -o SIMPLERROUTEDB

-n connectionUrlFormat,connectionUrlFormatAttr1,description,jarsURL,portNumber,

serverName,type4DatasourceClassName,type4DriverClassName

-v "jdbc:oracle:thin:[user]/[password]@[serverName]:[portNumber]:[connectionUrlFormatAttr1],

<SID>,Simplified Database Routing Sample Database,<JARS URL>,<PORT NUMBER>,

<SERVER NAME>,oracle.jdbc.xa.client.OracleXADataSource,oracle.jdbc.OracleDriver"
```

to check that the default broker registry was correctly updated with the sample SIMPLERROUTEDB JDBCProvider entry:

```
mqsireportproperties MB7BROKER -o SIMPLERROUTEDB -c JDBCProviders -r
```

If the update was successful, the reported property output from the command matches the following example:

```
JDBCProviders
  SIMPLERROUTEDB
    connectionUrlFormat='jdbc:oracle:thin:[user]/[password]@[serverName]:[portNumber]:[connectionUrlFormatAttr1]'
    connectionUrlFormatAttr1='orc1'
    connectionUrlFormatAttr2=''
    connectionUrlFormatAttr3=''
    connectionUrlFormatAttr4=''
    connectionUrlFormatAttr5=''
    databaseName='default_Database_Name'
    databaseType='default_Database_Type'
    databaseVersion='default_Database_Version'
    description='Simplified Database Routing Sample Database'
    environmentParms='default_none'
    jarsURL='C:\oracle\oraxx\jdbc\lib'
    portNumber='1521'
    securityIdentity='default_User@default_Server'
    serverName='localhost'
    type4DatasourceClassName='oracle.jdbc.xa.client.OracleXADataSource'
    type4DriverClassName='oracle.jdbc.OracleDriver'
```

Specify the user identifier and password to associate with the JDBC provider SIMPLERROUTEDB:

Use the mqsisetdbparms and mqsichangeproperties commands to specify a user identifier and password for the broker to use with JDBC provider SIMPLERROUTEDB. This user identifier must be the same user identifier that you used when you created the database.

a.Enter the following command to associate the user identifier and password with a security identity:

```
mqsisetdbparms MB7BROKER -n jdbc::mySecurityIdentity -u <user ID> -p <password>
```

b.Enter the following command to associate the security identity, which you defined in the preceding step, with the securityIdentity property of the JDBC provider SIMPLERROUTEDB:

mqsichangeproperties MB7BROKER -c JDBCProviders -o SIMPLERROUTEDB -n securityIdentity -v mySecurityIdentityc.You must stop and restart the broker for the changes to the JDBC provider to become available to a message flow that is already deployed. If you have already deployed the sample, stop and restart the broker.

=====Setting MQMD & MQRFH2 Header==============

```
SET OutputRoot.MQMD.Encoding = 273;

SET OutputRoot.MQMD.CodedCharSetId = 819;

SET OutputRoot.MQMD.Format = MQFMT_RF_HEADER_2;


SET OutputRoot.MQMD.MsgType = MQMT_DATAGRAM;

SET OutputRoot.MQMD.Persistence = MQPER_PERSISTENT;

SET OutputRoot.MQMD.Expiry = MQEI_UNLIMITED;


CREATE NEXTSIBLING OF OutputRoot.MQMD DOMAIN 'MQRFH2';

SET OutputRoot.MQRFH2.(MQRFH2.Field)Version = 2;

SET OutputRoot.MQRFH2.(MQRFH2.Field)Format = 'MQSTR';

SET OutputRoot.MQRFH2.(MQRFH2.Field)Encoding = 273;

SET OutputRoot.MQRFH2.(MQRFH2.Field)CodedCharSetId = 819;


SET OutputRoot.MQRFH2.usr.test='scm';

SET OutputRoot.MQRFH2.usr.test2=CURRENT_GMTTIMESTAMP;

SET OutputRoot.MQRFH2.psc.Command              = 'Publish';

SET OutputRoot.MQRFH2.psc.Topic      = '/TEST/MQRFSSS/TESTI';
```

=====SOA Principles implemented==========

Principles, in order of preference are;

Abstraction -expose only the required functionality, control access to service documentation and utilise our integration architecture to encapsulate legacy applications

Contract Standardisation - maintain data (XSD), policy and contract (WSDL) separately, define contracts prior to implementation

Loose Coupling -enforce contract centralisation, maximise logic-to-contract coupling

Reuse - Use a service registry to enable re-use through discoverability

Service Statelessness -build short-lived, stateless, services wherever possible. Allow for the storage of configuration information only

Discoverability -use design time discoverability as our primary means of discovery. Use standardised service contracts as an aid to discovery. Create a repository of services

Autonomy – acknowledge that our services can not be entirely autonomous due to their reliance of Enterprise systems such as SAP

==

1) Edit odbc64.ini(brokers_install_dir/ODBC64.ini)

2) mesisetdbparam command

3)

odbc64.ini sample file (ah14452_)

The correct content for HP-UX (PA-RISC platform) is:

```
################################################################

## 64 bit ODBC database driver manager initialisation file. ##

################################################################

# It is recommended that you take a copy of this file and #

# then edit the copy. #

# #

# 1. Complete the 'Mandatory information stanza' section #

# at the end of the file. #

# #

# 2. For each data source, add the name of the data source #

# into the 'List of data sources stanza' section. #

# #

# 3. For each data source, create a stanza in the #

# 'Individual data source stanzas' section. #

################################################################
```

```
#######################################

#### List of data sources stanza ######

#######################################

[ODBC Data Sources]

DB2DB=IBM DB2 ODBC Driver

ORACLEDB=DataDirect 5.0 64-bit Oracle Wire Protocol

ORACLERACDB=DataDirect 5.0 64-bit Oracle Wire Protocol (Real Application Clusters)

SYBASEDB=DataDirect 5.0 64-bit Sybase Wire Protocol

SYBASEDBUTF8=DataDirect 5.0 64-bit Sybase UTF8 Driver

INFORMIXDB=IBM Informix ODBC Driver


##########################################################

########## Individual data source stanzas ################

##########################################################

# DB2 stanza

[DB2DB]

Driver=libdb2Wrapper64.sl

Description=DB2DB DB2 ODBC Database

Database=DB2DB

# Oracle stanza

[ORACLEDB]

Driver=<Your install directory>/ODBC64/V5.0/lib/UKora20.sl

Description=DataDirect 5.0 Oracle Wire Protocol

HostName=<Your Oracle Server Machine Name>

PortNumber=<Port on which Oracle is listening on HostName>

SID=<Your Oracle SID>

CatalogOptions=0

ProcedureRetResults=1

EnableStaticCursorsForLongData=0

ApplicationUsingThreads=1

EnableDescribeParam=1

OptimizePrepare=1
```

```
WorkArounds=536870912
ColumnSizeAsCharacter=1
# Oracle Real Application Clusters stanza
[ORACLERACDB]
Driver=<Your Broker install directory>/ODBC64/V5.0/lib/UKora20.sl
Description=DataDirect 5.0 64bit Oracle Wire Protocol
HostName=<Your Oracle Server Machine Name>
PortNumber=<Port on which Oracle is listening on HostName>
ServiceName=<Your Oracle RAC Net Service Name>
CatalogOptions=0
EnableStaticCursorsForLongData=0
ApplicationUsingThreads=1
EnableDescribeParam=1
OptimizePrepare=1
WorkArounds=536870912
ProcedureRetResults=1
ColumnSizeAsCharacter=1
# Sybase stanza
[SYBASEDB]
Driver=<Your Broker install directory>/ODBC64/V5.0/lib/UKase20.sl
Description=DataDirect 5.0 Sybase Wire Protocol
Database=<Your Database Name>
ServerName=<YourServerName>
EnableDescribeParam=1
OptimizePrepare=1
SelectMethod=0
NetworkAddress=<YourServerName>,<YourPortNumber>
SelectUserName=1
# Sybase Stanza for a UTF8 datasource
[SYBASEDBUTF8]
Driver=<Your Broker install directory>/ODBC64/V5.0/lib/UKase20.sl
Description=DataDirect 5.0 64bit Sybase Wire Protocol
Database=<Your Database Name>
```

```
ApplicationUsingThreads=1

EnableDescribeParam=1

OptimizePrepare=1

SelectMethod=0

NetworkAddress=<Your Sybase Server Name>,<Your Sybase Port Number>

SelectUserName=1

ColumnSizeAsCharacter=1

Charset=UTF8

# Informix stanza

[INFORMIXDB]

Driver=libinfWrapper64.sl

Description=IBM Informix ODBC Driver

ServerName=<Your Informix Server Name>

Database=<Your Database Name>


##########################################

###### Mandatory information stanza ######

##########################################

[ODBC]

Trace=0 # To turn on ODBC trace set Trace=1

TraceFile=<A Directory with plenty of free space>/odbctrace.out

TraceDll=<Your Broker install directory>/ODBC64/V5.0/lib/odbctrac.sl

InstallDir=<Your Broker install directory>/ODBC64/V5.0

UseCursorLib=0

IANAAppCodePage=4

UNICODE=UTF-8
```

=====================================WS-A======================

WS-Addressing provides transport-neutral mechanisms to address Web services and messages.

Web Services Addressing (WS-Addressing) defines two interoperable constructs that convey information that is typically provided by transport protocols and messaging systems.

The two constructs are endpoint(resource where Web service messages can be targeted) references and message information headers.

sample:

```
<S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
            xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing">
  <S:Header>
   <wsa:MessageID>
     uuid:6B29FC40-CA47-1067-B31D-00DD010662DA
   </wsa:MessageID>
   <wsa:ReplyTo>
     <wsa:Address>http://business456.example/client1</wsa:Address>
   </wsa:ReplyTo>
   <wsa:To>http://fabrikam123.example/Purchasing</wsa:To>
   <wsa:Action>http://fabrikam123.example/SubmitPO</wsa:Action>
  </S:Header>
  <S:Body>
    ...
  </S:Body>
</S:Envelope>
```

=======================Default:HTTP & SOAP-HTTP ports==MB6.1 & 7==================================

7080 - all unencrypted traffic to all HTTPInput nodes in both ExecutionGroups

7083 - all encrypted traffic to all HTTPInput nodes in both ExecutionGroups

7800 - all unencrypted traffic for SOAPInput nodes deployed to the first ExecutionGroup that starts up

7801 - all unencrypted traffic for SOAPInput nodes deployed to the second ExecutionGroup that starts up

7843 - all encrypted traffic for SOAPInput nodes deployed to the first ExecutionGroup that starts up

7844 - all encrypted traffic for SOAPInput nodes deployed to the second ExecutionGroup that starts up

==

Connect to a remote queue manager using server connection channel

The most interesting and important feature introduced with RFH v3.5 is SSL capabilities. RFHUTIL now works as you would expect from a native WebSphere MQ client application, reading the MQSSLKEYR environment variable to locate the keystore.

I came across a script like this to connect to a remote queue manager using a server connection channel with a NULL_SHA SSL configuration as in here . Please test this before you can impact any of your systems.

```
@echo off

set RFHUTIL_HOME=”%cd%”

set SVRCONN=CLIENTS.ADMIN
set HOST=10.10.10.10
set PORT=1414

set MQSSLKEYR=C:\ih03\ssl\key

set MQSERVER=%SVRCONN%/TCP/%HOST%(%PORT%)

%RFHUTIL_HOME%\rfhutilc.exe
```

==SET-VI===================

```
mqsicreateconfigurableservice UNXS0383.MB1 -c JDBCProviders -o MBODS

mqsichangeproperties UNXS0383.MB1 -c JDBCProviders -o MBODS -n connectionUrlFormat,connectionUrlFormatAttr1,connectionUrlFormatAttr2,connectionUrlFormatAttr3,connectionUrlFormatAttr4,connectionUrlFormatAttr5,databaseName,databaseSchemaNames,databaseType,databaseVersion,description,environmentParms,jarsURL,jdbcProviderXASupport,maxConnectionPoolSize,portNumber,securityIdentity,serverName,type4DatasourceClassName,type4DriverClassName -v "jdbc:oracle:thin:[user]/[password]@[serverName]:[portNumber]:[connectionUrlFormatAttr1]","IBRS1","","","","","default_Database_Name","","Oracle","10.2.0.1.0","default_Description","default_none","/dmbu3/mqsi/config/UNXS0458.MB1/shared-classes/","true","10","16444","MBODSSecurityIdentity","unxs0190","oracle.jdbc.xa.client.OracleXADataSource","oracle.jdbc.OracleDriver"

mqsisetdbparms BKR2 -n jdbc::MBODSSecurityIdentity -u CWODS -p cradmin2k11


mqsicreateconfigurableservice UNXS0383.MB1 -c JDBCProviders -o MBREPOS

mqsichangeproperties BKR2 -c JDBCProviders -o MBREPOS -n connectionUrlFormat,connectionUrlFormatAttr1,connectionUrlFormatAttr2,connectionUrlFormatAttr3,connectionUrlFormatAttr4,connectionUrlFormatAttr5,databaseName,databaseSchemaNames,databaseType,databaseVersion,description,environmentParms,jarsURL,jdbcProviderXASupport,maxConnectionPoolSize,portNumber,securityIdentity,serverName,type4DatasourceClassName,type4DriverClassName -v "jdbc:oracle:thin:[user]/[password]@[serverName]:[portNumber]:[connectionUrlFormatAttr1]","dmbu1","","","","","default_Database_Name","","Oracle","10.2.0.1.0","default_Description","default_none","/dmbu1/mqsi/config/UNXS0383.MB1/shared-
```

```
classes/","true","10","1527","MBREPOSSecurityIdentity","unxs0383","oracle.jdbc.xa.client.OracleXADataSource","oracle.jdbc.OracleDriver"
```

```
mqsisetdbparms BKR2 -n jdbc::MBREPOSSecurityIdentity -u MBREPOS -p "RxP_~1(4,y*,Bt"
```

```
mqsicvp BKR2 -n MBREPOS
```

```
mqsireportproperties UNXS0460.MB1 -c JDBCProviders -r
```

```
mqsireportproperties UNXS0460.MB1 -c JDBCProviders -o MBREPOS -r
```

===================================SET-VII======================

```
select * from WMB_MSGS where MSGFLOW_NAME='com.kingfisher.ukbq.PurchaseOrder.PurchaseOrder_synchronisePurchaseOrder_ImportVendor_ApexPro_Sub' and GLOBAL_TRANSACTION_ID IN(2065878090) order by EVENT_TIMESTAMP desc;
```

```
--select * from WMB_MSGS where MSGFLOW_NAME='com.kingfisher.ukbq.AppointmentBooking' order by EVENT_TIMESTAMP desc;
```

```
--select * from WMB_MSGS where MSGFLOW_NAME='com.kingfisher.ukbq.CancelAppointment' order by EVENT_TIMESTAMP desc;
```

```
select STORE_CODE,TRANSACTION_ID,TeNDER_ID,AMOUNT,BUSINESS_DAY_DATE,SOURCE_SYSTEm,MASKED_ACCOUNT_ID AS CARD_NUMBER,TOKEN,SETTLEMENTRESP,COMMIDEA_TRANSACTION_ID,COMMIDEA_AUTH_DB from sales_transactions_tender where store_code='0182' and business_day_date='20140403' and settled='N';
```

```
select * from TRANSFER_EVENT where destination_agent='MFTMERCPRD' and source_agent='MFTMERCPRD' and originator_host='GBBQPWH001AP169.uk.b-and-q.com.'  order by TRANSFERSET_TIME desc;
```

```
destination_qm='UNXS0391.DP.QM1'
```

============PCI Trial Query – verify if Verifone settlement with null token did not happen.

```
select TRANSACTION_ID,store_code,TOKEN,SETTLED,business_day_date from sales_transactions_tender where store_code = '0186' and token is null and settled = 'N' and business_day_date = 20140331;
```

```
select * from sales_transactions_tender where store_code='0186' and token is null and

settled='N' and business_day_date = 20140331 and  commidea_transaction_id is not null and commidea_auth_db is not null;
```

```
select * from sales_transactions_tender where store_code IN ('0186','0182') and token is null and
```

```
settled='N' and business_day_date =20140416;
```

```
update sales_transactions_tender set SOLVE_PFG_FILTER= ‘N’  where store_code='0186' and token is null and

settled='N' and  and business_day_date = 20140331 and commidea_transaction_id is not null and commidea_auth_db is not null;
```

```
select * from sales_transactions_tender where transaction_id='0186-20140402-004-4799-924401';
```

```
--RELEASE RESPONSE: RELEASED;

--0182-20140403-007-9956-184189

--0182-20140403-007-9955-184188

select * from sales_transactions_tender where settled='N' and business_day_Date not in ('20140409') and tender_id not in ('Cash');
```

```
--update sales_transactions_tender set settlementresp='RELEASE RESPONSE: RELEASED;',settled='S' where transaction_id='0186-20140402-004-4799-924401';
```

```
select * from SALES_TRANSACTIONS_TENDER where tender_type in('P') and STORE_CODE in ('0182','0186') and token is null and SOLVE_PFG_FILTER='N' and settled is null and business_day_date>20140326;

transaction_id='0186-20140412-006-2822-12707'
```

```
--update SALES_TRANSACTIONS_TENDER set Processed_SEOD='U' , Processed_ASEOD='U',SOLVE_PFG_FILTER='N' where transaction_id='0186-20140412-006-2822-12707'
```

Show %Used SWAP of UNIX:

```
lsps –a

amqsdlq

runmqdlq QL QM < rule.tb

runmqdlq QL QM

wait(no)

action(retry)

enter

ctrl-d
```

mqsisetdbparms no longer requires a broker restart – since v7

```
git reset –merge to revert back normal  // error: you need to resolve your current index first
```

```
echo 'dis qs(SYSTEM.ADMIN.COMMAND.QUEUE) type(handle) all' | runmqsc TESTQMGR

#!/bin/bash

mqsichangeproperties ${MQSI_BROKER_NAME} -e $1 -o ComIbmJVMManager -n jvmVerboseOption -v"gc"

param1=$1

mqsichangeproperties ${MQSI_BROKER_NAME} -e $1 -o ComIbmJVMManager -n jvmSystemProperty -v"-verbose:gc -Xverbosegclog:/tmp/${param1}.gc.trc"
```

Abend file: /var/mqsi/common/errors/

JMS Setup:

http://pic.dhe.ibm.com/infocenter/cmgmt/v8r4m0/index.jsp?topic=%2Fcom.ibm.administeringcm.doc%2Fbpmi0035.htm

Space Allocated: select sum(bytes/1024/1024) size_mb from user_segments;

Space Used :

```
SELECT SUM(bytes) FROM user_segments

select MBRECORD, sum(bytes)/1024/1024 size

from dba_segments

group by MBRECORD

order by size desc;
```

```
SELECT SUM(bytes) FROM user_segments;

select sum(bytes/1024/1024) size_mb from user_segments;
```

```
#!/bin/bash

mqsichangeproperties ${MQSI_BROKER_NAME} -e $1 -o ComIbmJVMManager -n jvmVerboseOption -v"none"

mqsichangeproperties ${MQSI_BROKER_NAME} -e $1 -o ComIbmJVMManager -n jvmSystemProperty -v" "
```

mqsireportproperties UNXS0376.MB1 -b httplistener -o HTTPConnector -r // The default Message Broker value is 100. Therefore, when an HTTP client connects, it can send 100 requests before Message Broker issues aConnection: close and closes the socket. The next request the client sends creates a new socket and again can send up to 100 requests. In many environments this setting may be enough. But when maximum throughput is required, you may need to set this value higher or make it unlimited, in order to achieve the desired throughput.

```
mqsichangeproperties MyBroker -b httplistener -o HTTPConnector -n maxKeepAliveRequests -v 0
mqsichangeproperties MyBroker -b httplistener -o HTTPConnector -n maxThreads -v 2000 // You can set the maxThreads parameter to determine the maximum number of concurrent connections that Message Broker will accept.
```

```
ps -eaf | grep DataFlow| grep -v start  // to see when EG restared
```

http://publib.boulder.ibm.com/infocenter/aix/v7r1/index.jsp?topic=%2Fcom.ibm.aix.cmds%2Fdoc%2Faixcmds3%2Flsps.htm

```
mqsiapplybaroverride -b ProductReservation_processProductReservation.bar  -p SCREFIXYRL.properties  -r -o ProductReservation_processProductReservation1.bar
```

```
mqsireportproperties UNXS0391.MB1 -b webadmin -o HTTPConnector -a

mqsireportproperties UNXS0391.MB1 -b webadmin -o AllReportableEntityNames -r

mqsireportproperties UNXS0391.MB1 -c AllTypes -o AllReportableEntityNames -r

mqsireportproperties UNXS0389.MB1 -c SAPConnection -o AllReportableEntityNames –a
```

```
mqsideploy UNXS0390.MB1 -e  SWEB -d "ProductReservation_processProductReservation" -w 720 >> "DeploymentStatus.txt"

mqsideploy UNXS0390.MB1 -e  SWEB -a "ProductReservation_processProductReservation1.bar" -w 720 >> "DeploymentStatus.txt"DBA Team: @62547
```

Project Code: 17281 CR-1

Topas - memory utilization graphical

```
cat wmbevent.log |grep "Aug  4" | grep "user:err|error" | more

cat wmbevent.log |grep "May  2" | grep " FulfilmentSourceService" | more

cat wmbevent.log |grep "2013-01-19T09” |more

C:\>mqsireportproperties BKR3 -o ComIbmJVMManager  -e TEST4 –a

mqsireportproperties ${MQSI_BROKER_NAME} -e $1 -o ComIbmJVMManager -a

C:\>mqsireportproperties BKR3 -o ComIbmXmlParserFactory  -e TEST4 -a

C:\>mqsireportproperties UNXS0389.MB1 -o AllReportableEntityNames -a

C:\>mqsireportproperties BKR3 -o ComIbmJVMManager  -e TEST4 –r

mqsireportproperties UNXS0391.MB1  -b webadmin -o HTTPConnector –a

mqsireportproperties UNXS0391.MB1  -b webadmin -o server –a

mqsireportproperties UNXS0389.MB1 -c DataCaptureSource -o AllReportableEntityNames -r

mqsireportproperties UNXS0389.MB1 -c DataCaptureStore -o AllReportableEntityNames -r

C:\>mqsireportresourcestats UNXS0389.MB1 -e MDAT

cat wmbevent.log |grep "Sep 17" >> /support/home/wmbadmin/WMB.log

mqsibrowse UNXS0460.MB1 -t BROKERAAEG

echo `netstat -i | grep unxs | grep -v "bk" | awk '{ print $4 }' | tr '[:lower:]' '[:upper:]'`.MB.QM1
```

```
----Display specific status of queue attributes

DISPLAY QSTATUS(ERRORS.WMB.ERRORLOG.INPROGRESS) CURDEPTH UNCOM IPPROCS LGETDATE LGETTIME LPUTDATE LPUTTIME MONQ OPPROCS

DISPLAY CONN(*) TYPE(CONN) ALL

echo 'DISPLAY CONN(*) WHERE(UOWSTATE EQ ACTIVE)' |runmqsc UNXS0383.MB.QM1 |more

echo 'DISPLAY CONN(*) TYPE(CONN) ALL' | runmqsc UNXS0115.QM1 |grep 'AMQ8276'|grep 'UOWSTATE'|wc –l

echo 'DISPLAY CONN(*) TYPE(CONN) ALL' | runmqsc UNXS0115.QM1  >> CONNECTSTAT.txt


./qload -m UNXS03911.MB.QM1 -I ERRORS.WMB.ARCHIVE -f message1 -r1

mqsireportbroker UNXS0460.MB1

fteListAgents

fteListAgents -v

ftePingAgent MFTHRSAXPPRD

fteShowAgentDetails MFTHRSXAPPRD

fteShowAgentDetails -v MFTHRSAPPRD

fteStartAgent MFTETLUXATZ

fteStopAgent MFTATGBCCPRXF

fteStopAgent -i MFTHRSAPPRX  //force fully stop

fteCreateAgent -agentName MFTHRSX -agentQMgr UNXS0393.AGT.QM1 –f

fteCreateMonitor -ma MFTHRSAPPRD -mm UNXS039X.AGT.QM1 -mn ServiceNowPRDMonitor -md /var/mqsi/tempdata/saphrmqmft

/HRMQFT/ServiceNow -mt /var/mqm/scripts/MFTHRSAPPRDScripts/ServiceNowPRDTemplate.xml -pi 5 -pu minutes -tr match,"*.csv"


fteDeleteAgent MFTHRSAPPXD

fteDeleteMonitor -ma MFTHRSAPPRXD -mm UNXS0393.AGT.QM1 -mn ServiceNowPRDMonitor


./showMFTPRDAgents.sh

./startMFTHRSAPPRDAgent.sh

stopMFTHRSAPPRDAgent.sh

stopMFTHRSAPPRDAgent.sh

./startMFTHRSAPPRDA>
```

```
unxa0393:/var/mqm/scripts/MFTHRSAPPRDScripts>ll
total 168
drwxrws---    2 wmbadmin mqbrkrs       4096 12 Apr 11:15 .
drwxrwsrwx   11 wmbadmin mqbrkrs       4096 13 Jun 12:18 ..
-rwxrwxrwx    1 wmbadmin mqbrkrs        140 21 Feb 12:29 agent.properties
-rwxrwxrwx    1 wmbadmin mqbrkrs      12176 11 Apr 14:23 archive_53.pl
-rw-rw----    1 wmbadmin mqbrkrs       1320 24 Jun 14:02 archive_snow.log
-rwxrwxrwx    1 wmbadmin mqbrkrs        266 21 Feb 12:31 createServiceNowPRDMonitor.sh
-rwxrwxrwx    1 wmbadmin mqbrkrs         54 21 Feb 12:31 deleteMFTHRSAPPRDAgent.sh
-rwxrwxrwx    1 wmbadmin mqbrkrs        113 21 Feb 12:32 deleteServiceNowPRDMonitor.sh
-rwxrwxrwx    1 wmbadmin mqbrkrs         56 21 Feb 12:32 forceMFTHRSAPPRDAgent.sh
-rwxrwxrwx    1 wmbadmin mqbrkrs        852 21 Feb 12:32 MFTPRDControl.sh
-rwxrwxrwx    1 wmbadmin mqbrkrs         55 21 Feb 12:34 pingMFTHRSAPPRDAgent.sh
-rwxrwxrwx    1 wmbadmin mqbrkrs         74 21 Feb 12:34 pingMFTSNOWPRDAgent.sh
-rwxrwxrwx    1 wmbadmin mqbrkrs        965 21 Feb 12:38 ServiceNowPRDTemplate.xml
-rwxrwxrwx    1 wmbadmin mqbrkrs         97 21 Feb 12:38 setupMFTHRSAPPRDAgent.sh
-rwxrwxrwx    1 wmbadmin mqbrkrs         69 21 Feb 12:39 showMFTHRSAPPRDAgent.sh
-rwxrwxrwx    1 wmbadmin mqbrkrs         47 06 Sep 2012  showMFTPRDAgents.sh
-rwxrwxrwx    1 wmbadmin mqbrkrs       3910 12 Apr 11:16 snow_archive_control.ksh
-rwxrwxrwx    1 wmbadmin mqbrkrs         95 21 Feb 12:40 startMFTHRSAPPRDAgent.sh
-rwxrwxrwx    1 wmbadmin mqbrkrs         53 21 Feb 12:41 stopMFTHRSAPPRDAgent.sh
```

Issue with Agent when no reachable – Restart Agent & update the file to be processed – using touch command

```
/var/mqsi/tempdata/saphrmqmft/HRMQFT/ServiceNow>
```

```
grep -r -n "1891641427" .
```

jbossadm

test1234

```
cat wmbevent.log |grep "Jun 23" | grep "A schema" |more
```

```
D:\Program Files\IBM\MQSI\8.0.0.0>runmqsc QM1

5724-H72 (C) Copyright IBM Corp. 1994, 2009.  ALL RIGHTS RESERVED.

Starting MQSC for queue manager QM1.


cat KITS_ErrorLog_20112012.txt | grep "2012-12-20T17:19" | more

find . -type f -print |xargs grep -li 'B28013'


du -ak * | sort -nr >> /support/home/wmbadmin/diskUsage.txt

awk '{c+=gsub(s,s)}END{print c}' s='/HazardousMaterial' KITS_ErrorLog_08032013.txt

awk '{c+=gsub(s,s)}END{print c}' s='/HazardousMaterial' *.txt

2012-11-20T


alter ql(TEST_IN) PUT(DISABLED)

     1 : alter ql(TEST_IN) PUT(DISABLED)


alter qa(FULFILMENTSOURCE.STEP.INBOUND.WMB) GET(ENABLED)


mqsireportproperties UNXS0462.MB1   -e TEST_INSTANCE -o ComIbmJVMManager –a

mqsireportproperties UNXS0378.MB1   -e SWEB  -o ComIbmJVMManager -r

mqsireportproperties UNXS0391.MB1   -e MDAT  -o ComIbmJVMManager -r



mqsichangeproperties  UNXS0378.MB1   -e  SWEB  -o ComIbmJVMManager -n jvmMaxHeapSize -v 2147483648

mqsistopmsgflow UNXS0231.MB1  -e DarwinTest

mqsistartmsgflow UNXS0231.MB1 -e DarwinTest


mqsireportflowmonitoring UNXS0391.MB1   -g –j

mqsireportflowmonitoring UNXS0391.MB1 -g -j | grep "active" | more

mqsireportflowmonitoring UNXS0391.MB1 -g -j | grep "inactive" | more

mqsichangeflowmonitoring ${MQSI_BROKER_NAME} -g -j -c active

mqsireportflowmonitoring UNXS0391.MB1 -e SUPC -k Shipment_synchroniseShipment-ASN –j
```

```
mqsichangeflowmonitoring UNXS0379.MB1 -e SWEB -k CarrierBooking_getCarrierBookingAvailability -j -c active

mqsichangeflowmonitoring UNXS0393.MB1 -e SWEB -k SharePrice_synchroniseSharePrice -j -c inactive

mqsireportflowmonitoring UNXS0379.MB1 -e SWEB -k CarrierBooking_getCarrierBookingAvailability -j

mqsistopmsgflow UNXS0389.MB1  -e MDAT -k ProductRangingService -m
com.kingfisher.ukbq.ProductRangingService.ProductRanging_STEP_Pub

mqsistartmsgflow UNXS0389.MB1  -e MDAT -k ProductRangingService -m
com.kingfisher.ukbq.ProductRangingService.ProductRanging_STEP_Pub


mqsistartmsgflow UNXS0378.MB1  -e SAPR3OUT -k SAPR3GenericOutbound


mqsisetdbparms UNXS0383.MB1 -n MBREPOS -d

mqsisetdbparms UNXS0383.MB1 -n MBODS -d


mqsisetdbparms BKR1 -n MBREPOS -u MBREPOS -p MBR3P0S

mqsisetdbparms UNXS0383.MB1 -n MBODS -u MBODS -p MB0D5


mqsicvp -n MBRECORD -u MBRECORD -p hjfdhhcd$i1

mqsicvp -n MBREPOS -u MBREPOS -p MBR3P0S

mqsicvp -n MBODS -u MBODS -p MB0D5

mqsireadbar -b Communication_processCommunication.bar –r

tcpdump -i en1 host 10.246.7.13

lsps –a

topas

nomon


PATH=/usr/bin:/etc:/usr/sbin:/usr/ucb:$HOME/bin:/usr/bin/X11:/sbin:.


export PATH


if [ -s "$MAIL" ]          # This is at Shell startup.  In normal

then echo "$MAILMSG"       # operation, the Shell checks

fi                         # periodically.

TMOUT=43200 ; TIMEOUT=43200
```

```
export readonly TMOUT TIMEOUT


#Set the Prompt

PS1=`hostname`':$PWD'">"

export PS1


alias ll='ls -al'

set -o vi


export ODBCINI=/var/mqsi/odbc/odbc.ini

export ODBCSYSINI=/var/mqsi/odbc/odbcinst.ini

export MQSI_BROKER_NAME=`netstat -i | grep unxs | grep -v "bk" | awk '{ print $4 }' | tr '[:lower:]' '[:upper:]'`.MB1

export CLASSPATH=/var/mqsi/shared-classes/ojdbc6.jar:$CLASSPATH


alias taillog='tail -f /var/mqsi/log/${MQSI_BROKER_NAME}/wmbevent.log'

ls -1 S000006[89].*|xargs rm –fr

awk '{c+=gsub(s,s)}END{print c}' s='UNXS0391.MB1' KITS_ErrorLog_08032013.txt


/usr/mqm/bin/runmqlsr -r -m UNXS0391.MB.QM1 -t TCP -p 1430

/usr/mqm/bin/runmqchi -m UNXS0391.MB.QM1 -q SYSTEM.CHANNEL.INITQ -rUNXS0391.MB.QM1

ps -eaf|grep mqm |grep UNXS0391.MB.QM1

/usr/mqm/bin/runmqchl -c UNXS0391.TO.LNXS0236 -m UNXS0391.MB.QM1


find . -exec grep -l B290139073 {} \;

grep -l FulfilmentSource_STEP_Pub *


B280138116


mqsichangetrace UNXS0390.MB1 -u -e SWEB  -k CommunicationService_processCommunication -f
com.kingfisher.ukbq.Communication.processCommunication -l debug -c 5000 -r


mqsichangetrace UNXS0390.MB1 -u -e SWEB -k CommunicationService_processCommunication -f
com.kingfisher.ukbq.Communication.processCommunication -l none
```

```
mqsireadlog UNXS0390.MB1  -u -e SWEB -o trace.xml

mqsiformatlog -i trace.xml -o trace.log

====================
```

```
mqsichangetrace UNXS0240.MB1 -u -e RETL  -l debug -r -c 50000

mqsichangetrace UNXS0240.MB1 -u -e RETL  -l none

mqsireadlog UNXS0240.MB1 -u -e RETL  -f -o MB1_HT1.xml

mqsiformatlog -i MB1_HT1.xml -o MB1_HT2.txt
```

Service Trace:

```
mqsichangetrace BKR2 -t -b -l debug

mqsichangetrace BKR2 -t -b -l none

mqsireadlog BKR2 -t -e EG3  -f -o MB1_HT3.xml

mqsiformatlog -i MB1_HT3.xml -o MB1_HT3.txt

--------------------------------------
```

:FTE Alert:

This is a Known issue on the system.

Defect already raised with Mike Park

Reference Incident: 1938468,1993207

Host Name : unxs0007.uk.b-and-q.com

Host IP   : 172.19.174.16 / 28.4.174.16

System Number : 01

Client : 901

Language Code : EN

Code Page: 1100

UserName: MBROKER

Password: Wmb@dm1n

```
# ipcs -l //show limit

# ipcs -s /show semafore

# impcs -m / show Shared Memory

# ipcs -s 4194315 3145740 13

# ipcrm -s 818937948
```

==MFTSetup============

app0548.gha.kfplc.com

1. Create MFT Agent MFTSTEPGHAPRD
2. Create 7 MFT Monitors
3. Create neccessary stop/start and status scripts

unxs0393.gha.kfplc.com

1. Create Sender and receiver channels on UNXS0393.CDN.QM1 to Agent queue manager LNXS0330.AGT.QM1
2. Create Sender and receiver channels on UNXS0393.CMD.QM1 to Agent queue manager LNXS0330.AGT.QM1

lnxs0330.gha.kfplc.com

1. Create Agent queue manager
2. Create sender and receiver channels to UNXS0393.CMD.QM1 and UNXS0393.CDN.QM1
3. Create server conn channel MFT.SFX.SVRCONN

======BackOut===============

1. app0548.gha.kfplc.com

Delete newly created agent, monitors and scripts

2. unxs0393.gha.kfplc.com

Remove newly created channels

3. lnxs0330.gha.kfplc.com

Delete queue managers which will remove the channels

============Test Plan===============

1. Run fteListAgents command on STEP to confirm agent is active

2. Run fteListMonitors command to confirm monitors are running

3. From LNX0330 ping channels to UNXS0393.CDN.QM1 and UNXS0393.CMD.QM1

4. From UNIX0393 ping channels from UNXS0393.CDN.QM1 and UNXS0393.CMD.QM1 to LNXS0330

=======================

Dead Letter Handler =================

```
C:\>runmqdlq SYSTEM.DEAD.LETTER.QUEUE bowmanga

WAIT(YES) RETRYINT(20)
```

```
REASON(MQRC_Q_FULL) ACTION(RETRY) +
RETRY(5)
REASON(MQRC_PUT_INHIBITED) ACTION(RETRY) RETRY(5)
REASON(*) ACTION(FWD) FWDQ('DEADQ')
^Z
^Z
2005-04-08 02.20.09 AMQ8708: Dead-letter queue handler started to process
INPUTQ(SYSTEM.DEAD.LETTER.QUEUE).
```

disable the queue manager accounting or the statistics monitoring features using runmqsc commands. Example:

```
ALTER QMGR ACCTQ(OFF) STATQ(OFF)
```

CAVEAT: You need to be very careful because if you make a typo and accidentally delete the contents of other SYSTEM queues, you may corrupt the queue manager and then you may have to delete the entire queue manager and recreate it.

```
CLEAR QLOCAL(SYSTEM.ADMIN.ACCOUNTING.QUEUE)
CLEAR QLOCAL(SYSTEM.ADMIN.STATISTICS.QUEUE)
```

In case the CLEAR command fails (such as reporting that the queue has been opened by another process), then you could use the following sample program, which will get every message from the queue (in a destructive get mode):

```
amqsget SYSTEM.ADMIN.STATISTICS.QUEUE QueueManager
ps -ef | grep DataFlow
ps -ef | grep DataFlow | grep -v start
topas 25231456
ls -lrt /tmp
changeEGJVMSizes.sh
cd  /var/mqm/scripts/WMBS
./turnEGJVMGCOn.sh RETL
./turnEGJVMGCOn.sh RETLODS1
./turnEGJVMGCOn.sh RETLODS2
./turnEGJVMGCOn.sh RETLPOSBSKT1
./turnEGJVMGCOn.sh RETLPOSBSKTORCH
./turnEGJVMGCOn.sh RETLSOSYNCBW
./turnEGJVMGCOn.sh SAPR3OUT
./reportEGJVM.sh RETL | grep Heap
./reportEGJVM.sh RETLODS1 | grep Heap
```

```
./reportEGJVM.sh RETLODS2 | grep Heap

./reportEGJVM.sh RETLPOSBSKT1 | grep Heap

./reportEGJVM.sh RETLPOSBSKTORCH | grep Heap

./reportEGJVM.sh RETLSOSYNCBW | grep Heap

./reportEGJVM.sh SAPR3OUT | grep Heap
```

```
1.        No Broker restart is required and to be configured for both – Email In and Out

2.        mqsicreateconfigurableservice UNXS0389.MB1 -c SMTP -o SMTP_Alias

3.        mqsichangeproperties UNXS0389.MB1 -c SMTP -o SMTP_Alias -n serverName -v oa.ghanp.kfplc.com:25

4.        mqsichangeproperties UNXS0389.MB1 -c SMTP -o SMTP_Alias -n securityIdentity -v BQIdentity

5.        mqsisetdbparms UNXS0389.MB1 -n SMTP::BQIdentity -u SE3TEST1 -p <pass???>
```

```
mqsideleteconfigurableservice UNXS0389.MB1 -c SMTP -o SMTP_Alias
```

```
unxa0389:/var/mqsi/log/UNXS0389.MB1>telnet  mailhost.gha.kfplc.com 25

Trying...

Connected to mailhost.gha.kfplc.com.

Escape character is '^]'.

220 KITSFDC1WG1EB01.gsd.kingfisher.com Microsoft ESMTP MAIL Service, Version: 6.0.3790.4675 ready at  Fri, 21 Feb 2014
15:12:55 +0000

mail from: indrani.pal@kingfisher.com

503 5.5.2 Send hello first

helo

250 KITSFDC1WG1EB01.gsd.kingfisher.com Hello [10.249.0.16]

mail from:indrani.pal@kingfisher.com

250 2.1.0 indrani.pal@kingfisher.com....Sender OK

rcpt to:abhijit.karmakar@kingfisher.com

250 2.1.5 abhijit.karmakar@kingfisher.com

data

354 Start mail input; end with <CRLF>.<CRLF>

test

mail

.

250 2.6.0 <KITSFDC1WG1EB01UxP90000670c@KITSFDC1WG1EB01.gsd.kingfisher.com> Queued mail for delivery
```

```
h = Help information     q = Quit nmon            0 = reset peak counts                                          x
+ = double refresh time  - = half refresh         r = ResourcesCPU/HW/MHz/AIX                                    x
c = CPU by processor     C=upto 128 CPUs          p = LPAR Stats (if LPAR)                                       x
l = CPU avg longer term  k = Kernel Internal      # = PhysicalCPU if SPLPAR                                      x
m = Memory & Paging      M = Multiple Page Sizes  P = Paging Space                                               x
d = DiskI/O Graphs       D = DiskIO +Service times o = Disks %Busy Map                                           x
a = Disk Adapter         e = ESS vpath stats      V = Volume Group stats                                         x
^ = FC Adapter (fcstat)  O = VIOS SEA (entstat)   v = Verbose=OK/Warn/Danger
x
n = Network stats        N=NFS stats (NN for v4)  j = JFS Usage stats                                            x
A = Async I/O Servers    w = see AIX wait procs   "="= Net/Disk KB<-->MB                                         x
b = black&white mode     g = User-Defined-Disk-Groups (see cmdline -g)                                           x
t = Top-Process --->     1=basic 2=CPU-Use 3=CPU(default) 4=Size 5=Disk-I/O                                      x
u = Top+cmd arguments    U = Top+WLM Classes      . = only busy disks & procs
x
W = WLM Section          S = WLM SubClasses                                                                      x
[ = Start ODR            ] = Stop ODR                                                                            x
~ = Switch to topas screen
```

The nmon command provides the following views in interactive mode:

- Adapter I/O statistics (using the a key)
- AIO processes view (using the A key)
- Detailed Page Statistics (using the M key)
- Disk busy map (using the o key)
- Disk groups (using the g key)
- Disk statistics (using the D key)
- Disk statistics with graph (using the d key)
- ESS vpath statistics view (using the e key)
- Fibre Channel adapter statistics (using the ^ key)
- JFS view (using the j key)
- Kernel statistics (using the k key)
- Long term processor averages view (using the l key)
- Large page analysis (using the L key)

- Memory and paging statistics (using the m key)
- NFS panel (using the N key)
- Network interface view (using the n key)
- Paging space (using the P key)
- Process view (using the t and u keys)
- Processor usage small view (using the c key)
- Processor usage large view (using the C key)
- Shared Ethernet adapter statistics (using the O key)
- Shared-processor logical partition view (using the p key)
- System resource view (using the r key)
- Volume group statistics (using the V key)
- Verbose checks OK/Warn/Danger view (using the v key)
- WLM view (using the W key)

==SET-VIII===============

```
mqsideploy UNXS0391.MB1 -e  RETL -d "SalesOrderService" -w 720 >> "DeploymentStatus.txt"
mqsideploy UNXS0391.MB1 -e  RETL -a "SalesOrderService.bar" -w 720 >> "DeploymentStatus.txt"
```

```
mqsideploy BKR2  -e  EG1 -d "ProductService" -w 720 >> "DeploymentStatus.txt"
mqsideploy BKR2  -e  EG2 -a "ProductService.bar" -w 720 >> "DeploymentStatus.txt"
```

```
mqsireportproperties UNXS0387.MB1 -e SUPC -o HTTPConnector -a -n port
```

```
/var/mqsi/log/
```

B&Q House Networks and Remote access networks

TCP/IP 1630

UNXS0393.gha.kfplc.com

MBR3P0S

dmbu1

MBREP0S@0383

```
mqsisetdbparms BKR2 -n MBREPOS -u MBREPOS -p MBREP0S@0383
mqsisetdbparms UNXS0231.MB1 -n MBREPOS -d
mqsisetdbparms UNXS0389.MB1 -n eis::SAPR3IDOCINBOUND.inadapter -u MBROKER -p Wmb@dm1n
```

--------- not working

```
mqsicvp UNXS0389.MB1 -n SAPR3IDOCINBOUND.inadapter -u MBROKER -p Wmb@dm1n
mqsicvp -n SAPR3IDOCINBOUND.inadapter -u MBROKER -p Wmb@dm1n
mqsicvp UNXS0391.MB1 -n eis::SAPR3IDOCINBOUND.inadapter
```

Report the dependencies for the WebSphere Adapter for SAP:

```
mqsireportproperties UNXS0385.MB1  -c EISProviders -o SAPR3IDOCINBOUND.inadapter -r
```

Display all SAPConnection configurable services:

```
mqsireportproperties UNXS0391.MB1 -c SAPConnection -o AllReportableEntityNames -r
```

```
mqsicvp UNXS0379.MB1 -n MBREPOS -v
mqsicvp -n MBREPOS -u MBREPOS -p MBR3P0S
```

```
mqsichangeproperties  BKR2  -e  EG2  -o ComIbmJVMManager -n jvmMaxHeapSize -v 2147483648
```

```
mqsireportproperties UNXS0387.MB1 -e SUPC -o ComIbmJVMManager -r


mqsistopmsgflow UNXS0383.MB1  -e SUPC -k FulfilmentService

mqsistartmsgflow UNXS0383.MB1 -e SUPC -k FulfilmentService


============================================

mqsichangetrace UNXS0387.MB1 -u -e MDAT -l debug -r -c 5000


mqsichangetrace UNXS0387.MB1 -u -e MDAT   -l none

mqsireadlog UNXS0387.MB1 -u -e MDAT  -o trace_MDAT.xml

mqsiformatlog -i trace_MDAT.xml -o formattrace_MDAT.log


notepad formattrace.log

============================================

mqsichangetrace UNXS0387.MB1 -u -e SAPR3IN -k SAPR3GenericInbound -f
com.kingfisher.ukbq.sapr3genericinbound.SAPR3GenericInbound -l debug -r -c 5000


mqsichangetrace UNXS0387.MB1 -u -e SAPR3IN -k SAPR3GenericInbound -f
com.kingfisher.ukbq.sapr3genericinbound.SAPR3GenericInbound -l none

mqsireadlog UNXS0387.MB1 -u -e SAPR3IN -o trace_SAPR3IN.xml

mqsiformatlog -i trace_SAPR3IN.xml -o formattrace_SAPR3IN.log


notepad formattrace.log


=======MBR3P0S=====================================


cat wmbevent.log |grep "Apr 14" | grep "user:err|error" | more

ZIT_ARTMAS01.SAP.INBOUND.WMB
```

****

**GIT
COMMANDS**

Installation & configuration Guide:

Forking and public repository:

```
$ git clone git@github.com:KITSAbhijitKarmakar/ninject.extensions.messagebroker.git
$ git clone http://git@github.com:KITSAbhijitKarmakar/WMB-Test.git
```

private repos:

```
$ git clone https://username@github.com/username/project.git
```

===============================

Global setup:

 Set up git

```
  git config --global user.name "Abhijit Karmakar"

  git config --global user.email abhijit.karmakar@kingfisher.com
```

Add your public key

     Next steps:

```
  mkdir WMB-Test

  cd WMB-Test

  git init

  touch README

  git add README

  git commit -m 'first commit'

  --git remote add origin git@github.com:KITSAbhijitKarmakar/WMB-Test.git

  git remote add origin  git@github.com:KITSGitHubAdmin/KITS-WS_MessageBroker.git

  git push -u origin2 master
```

Existing Git Repo?

```
  cd existing_git_repo

  git remote add origin git@github.com:KITSAbhijitKarmakar/WMB-Test.git

  git push -u origin master
```

==============================

Setup SSH public keys as follows:

1.cd ~/.ssh

2.Generate pub/private key: ssh-keygen -t rsa -C "abhijit.karmakar@kingfisher.com"

3.Copy whole string from “public key for pasting into OpenSSH Authorized_keys

file” from the genereated poublic key.

4. Send the key to the GitHub repository owner. GitHub owner must add add it to github (account/SSH Public Keys).

========================verification======

1. ssh -vT github.com

2. To verify that you have a key generated and loaded into SSH: ssh-add -l

3. map host

```
        207.97.227.239                    github.com
```

=============

```
git show 4fda14eefe0388e836aac8feaca68ab18bfad6b1:path/to/file.c > savetohere.txt

git show master~2:path/to/file.c > savetohere.txt
```

===========problem!=======

If problem occure due to parallel pushing/pulling ; use below way to replace the whole local repo with the one from GitHub.com

1.Enter into your Repo directory

2. rm -rf .git

3.

```
git config --global user.name KITSAbhijitKarmakar

git config --global user.email abhijit.karmakar@kingfisher.com
```

4. git init

5. crate & stage new file to master & commit.

6. git checkout -b BL16281_01

7.git remote add origin git@github.com:KITSGitHubAdmin/KITS-WS_MessageBroker.git

8.git pull origin BL16281_01:BL16281_01

==================problem!===========

git stash // use only when you think you have staged dome file but those to be ignored for now

==================problem!===========

Before pushing, do a git pull with rebase option. This will get the changes that you made online (in your origin) and apply them locally, then add your local changes on top of it.

```
git pull --rebase
```

and then which will totally replace your local changes:

```
git pull --force origin BL16281_01:BL16281_01
```

Error:

"

```
$ git pull origin BL16281_01:BL16281_01
Enter passphrase for key '/c/Users/571521/.ssh/id_rsa':
warning: no common commits
remote: Counting objects: 6382, done.
remote: Compressing objects: 100% (1728/1728), done.
remote: Total 6382 (delta 3574), reused 6271 (delta 3468)
Receiving objects: 100% (6382/6382), 69.11 MiB | 241 KiB/s, done.
Resolving deltas: 100% (3574/3574), done.
From github.com:KITSGitHubAdmin/KITS-WS_MessageBroker
 ! [rejected]        BL16281_01 -> BL16281_01  (non-fast-forward)
```

Trick:

If you only want the shortened hash:

```
git log --pretty=format:'%h' -n 10
git whatchanged -p
```

****

******************************STATEMENTS**

```
SET Environment.PersonMatch[]=SELECT P FROM
   InputRoot.XMLNSC.ns2:GetPersonResponse.ns2:PayloadArea.ns2:Person[] AS P
WHERE P.(XMLNSC.Attribute)EID=EIDMatch;
```

```
#sendMessagesInBulk.ksh
#!/bin/ksh -e
```

```
typeset -i NUM_OF_MSGS=5000
QUEUEMANAGER1=TEST1
QUEUE1=DELVRY03
QUEUEMANAGER2=TEST1
QUEUE2=WMMBID02

## FUNCTION TO SEND MESSAGES TO THE QUEUE SPECIFIED
send_messages() {
  QUEUEMANAGER=$1
  QUEUE=$2
  MSGCOUNT=$3
  echo "Sending XML message to $QUEUE on $QUEUEMANAGER $MSGCOUNT times." >&2
  i=0
  while [ $i -lt $MSGCOUNT ]
  do
    createTaxXMLInput.ksh $QUEUEMANAGER $QUEUE
    i=$(($i + 1))
  done
  echo "Completed sending messages to $QUEUE on $QUEUEMANAGER." >&2
}

## SPAWN THREADS
send_messages $QUEUEMANAGER1 $QUEUE1 $NUM_OF_MSGS &
send_messages $QUEUEMANAGER2 $QUEUE2 $NUM_OF_MSGS &

## WAIT FOR CHILDREN
echo "Waiting for children processes to complete." >&2
wait

echo "Done." >&2
===================createTaxXMLInput.ksh===============================
#!/bin/ksh -x
echo "Starting script at $(date)." >&2
```

```
## FUNCTIONS
show_usage() {
  echo "Usage:\t$(basename $0) QUEUEMANAGERNAME QUEUENAME" >&2
  echo "\t$(basename $0) TEST1 TESTQUEUE" >&2
  return 0
}

## SANITIZING INPUT PARAMETERS
if [ $# -ne 2 ]
then
  echo "Script must be supplied two and only two parameters." >&2
  show_usage
  echo "Exiting script at $(date)."
  exit 20
fi

## INITIALIZING VARIABLES
QMGR=$1
QNAME=$2
CURRENTDATE=$(date +"%Y-%m-%d")
RTNCODE=0
TMPFILE=${$}_SapZRetRfcVatsitetabget.log

## START OF MAIN SCRIPT
echo "Target details:" >&2
echo "\tQueue Manager=${QMGR}" >&2
echo "\tQueue=${QNAME}" >&2

echo "Pushing data to target." >&2

## PUSHING XML TO QUEUE
/usr/mqm/samp/bin/amqsput $QNAME $QMGR >$TMPFILE 2>&1 <<ENDOFDATA
```

```
<?xml version="1.0" encoding="UTF-8"?><sapzretrfcvatsitetabget:SapZRetRfcVatsitetabget
"><IDateValid>$CURRENTDATE</IDateValid><IValidOnly>X</IValidOnly></sapzretrfcvatsitetabget:SapZRetRfcVatsitetabget>

ENDOFDATA


RTNCODE=$?


grep -i 'Reason Code' $TMPFILE >/dev/null 2>&1

RTNCODE2=$?


if [ $RTNCODE -ne 0 -o $RTNCODE2 -eq 0 ]

then

  echo "Script failed to push the data to the queue." >&2

  echo "Return code was ${RTNCODE}." >&2

  echo "Output from the process was:" >&2

  echo "*************** START ***************" >&2

  cat $TMPFILE >&2

  echo "***************  END  ***************" >&2

  echo "Check that the queue manager and queue names provided to the script were correct and that the apropriate privileges have been granted." >&2

  rm $TMPFILE

  echo "Ended script at $(date)." >&2

  exit 10

fi


echo "Data successfully pushed to target." >&2

echo "Ended script at $(date)." >&2

rm $TMPFILE

exit 0

==============================================

#bin/ksh


#Simple script that holds all the deployment commands for all bar files inscope                    for F1
```

```
        echo "Enter Full path to bar file location"

        read barfilelocation


            echo "Removing older Deployment status file\n\n"

            rm "$barfilelocation/DeploymentStatus.txt"


            echo "Deployment of all F1 Bar Files commencing....\n\n"


            mqsideploy TEST_BROKER -e TEST_EG  -a "$barfilelocation/SimpleMQtoMQMsgFlow.msgflow.generated.bar" -w180
>> "$barfilelocation/DeploymentStatus.txt"


            ##mqsideploy TEST_BROKER -e  TEST_EG  -a "$barfilelocation/ForecastService.bar" -w180 >>
"$barfilelocation/DeploymentStatus.txt"

         echo "Deployment Command Executed. Check Deployment status @ $barfilelocation/DeploymentStatus.txt \n\n"


        echo "Do you want to confirm deployment status? Enter y/n"

        read confirmYorN


        if [ $confirmYorN = "Y" -o $confirmYorN = "y" ];then

            mqsilist UNXS0231.MB1 -r

        fi



exit
```

====What's new in Version 7.0?===================================

1.Brokers maintain configuration data in the local file system

Brokers create and manage configuration data in an internal repository in the local file system, and have no requirement for a database

2.Set up broker administration security to control the authority that is required by users to complete specific administrative tasks. You can enable security when you create a broker, or change it later on an existing broker. This option, which uses WebSphere MQ facilities, replaces Access Control Lists (ACLs) that were managed by the Configuration Manager in previous versions.

3.Multi-instance brokers and queue managers store their configurations on shared network storage so that if a failure occurs in an active component, the passive component assumes the configuration and operation of the active component. The use of queue managers in this way avoids the requirement for a high availability solution

4.Audit and monitoring -You can now generate comprehensive audit and monitoring events from message flows, either at design time or operationally, for new and existing message flows. These events can be consumed by a diverse range of applications and systems, including WebSphere Business Monitor, WebSphere MQ and JMS applications, and vendor applications

5.Resource statistics -You can start and stop statistics collection at broker or execution group level by using the WebSphere Message Broker Explorer, the CMP API, or the mqsichangeresourcestats command.

====What's new in Version 8.0?=================================

1.Applications and libraries introduce a new way of creating and managing resources when you are developing and deploying applications

2.Resource statistics for JMS

3.Activity Logs

4.Deploying flows and applications in different mode

5.Deployable ESQL ,subflow,

6.Improved message modeling -DFDL support

7.Record and replay - If you have configured your message flow to emit event messages, the monitoring events publish selected message data, which you can then view or replay (resubmit for processing). The record function subscribes to the published monitoring data, and stores the data in a database. You can then view the data through the web user interface, or replay it by resending the message to an MQ queue

8.Web Services Reliable Messaging(WS-RM)[WS-RM is applicable only to HTTP transport.,The purpose of WS-RM is to ensure delivery of messages in situations such as the destination endpoint being temporarily unavailable (for example, in the case of a server restart) or the message path crossing multiple transport connections, any of which might fail (for example, across a firewall). WS-RM offers greater reliability when using HTTP transport, but has a performance impact.

],The client's MakeConnection request allows the server to respond with any queued messages that have not been received by the client

9 removing previous connection latency by using connection on flow starts

****

*************************FUNDA*************************************

```
cat syslog | grep -i 'Jul 27' >/tmp/syslog.txt
```

```
DECLARE l_CReturn CHAR CAST(CAST('X''0A''' AS BLOB) AS CHAR CCSID 1208);

SET orgProth_blob =  CAST ( ASBITSTREAM(INPUT_ROOT.MQMD )  AS BLOB);

SET orgPldBlob = ASBITSTREAM(INPUT_ROOT.XMLNSC,INPUT_ROOT.MQMD.Encoding,INPUT_ROOT.MQMD.CodedCharSetId);

SET prid=CAST(LOCAL_ENV.Destination.HTTP.RequestIdentifier AS CHARACTER);
```

```
CAST(CURRENT_GMTTIMESTAMP AS CHARACTER FORMAT 'yyyy-MM-dd HH:mm:ss.SSS');

DECLARE Hdr_cursor REFERENCE TO INPUT_ROOT.XMLNSC.*[<].*:Header;

PASSTHRU ('UPDATE slg SET n_ts =?  WHERE p_m_id=? OR tid=? AND sec=?',nS,PId,TId,SC);

 ================================================================================


 ----------------------Create BLOB of Exception List---------------------------------------

SET OutputRoot.XML = InputExceptionList;

SET Environment.Variables.ExceptionList = ASBITSTREAM(OutputRoot.XML);

SET OutputRoot.XMLNSC.Test.errStkTrace = CAST(Environment.Variables.ExceptionList AS BLOB);

DELETE FIELD OutputRoot.XML;

---------------------------------------


MbMessage inGlobalEnv = inAssembly.getGlobalEnvironment();

MbMessage inExceptionList = inAssembly.getExceptionList();

MbElement glbTemp = inGlobalEnv.getRootElement();

byte[] baMsgStream = null;

MbElement tmpException = glbTemp.createElementAsLastChild(MbXMLNSC.PARSER_NAME);

tmpException.addAsLastChild(ExList.getRootElement().getLastChild().copy());

baMsgStream = glbTemp.getLastChild().toBitstream("", "", "", 0, 0, 0);

================================================================================

CREATE FUNCTION getIPHostName()

/*****************************************************************************

* A Global Function that will get the details of the HostName in String format

* of the Computer on which Calling Component is running

*****************************************************************************/

RETURNS CHAR

LANGUAGE JAVA

EXTERNAL NAME "com.ibm.test.GetIPAddress.getHostString";



CREATE PROCEDURE getLastExceptionDetail(IN InputTree reference,OUT meNu integer,

OUT mText char,OUT erC CHAR,OUT eId CHAR)

/****************************************************************************
```

```
* A Global procedure that will get the details of the last exception from a message
****************************************************************************/
BEGIN
-- Create a reference to the first child of the exception list
DECLARE ptrException reference to InputTree.*[1];
SET errorComponentId = ptrException.Label;
-- keep looping while the moves to the child of exception list work
WHILE LASTMOVE(ptrException) DO
    -- store the current values for the error number and text
    IF ptrException.Number IS NOT NULL THEN
        SET meNu = ptrException.Number;
        SET eId = ptrException.Label;
        SET mText = ptrException.Text||':'||'['||ptrException.*[<].Text||']';
        SET erC = FIELDNAME(ptrException);
    END IF;
    -- now move to the last child which should be the next exceptionlist
    MOVE ptrException LASTCHILD;
END WHILE;
END;
------------------------------------
    public String[] getLastExceptionDetails(MbMessageAssembly inAssembly)
            throws MbException {
        String[] lastExceptionDetails = new String[4];
        MbElement excepTypeRoot = inAssembly.getExceptionList()
                .getRootElement().getLastChild();
        String errorCompID = excepTypeRoot.getFirstElementByPath("Label")
                .getValueAsString();

        while (excepTypeRoot.getLastChild() != null) {
            excepTypeRoot = excepTypeRoot.getLastChild();
        }

        lastExceptionDetails[1] = errorCompID;// The component in Error
        lastExceptionDetails[3] = excepTypeRoot.getValueAsString();

        excepTypeRoot = excepTypeRoot.getParent().getParent(); // change pointer

        // to parent's

        // parent

        // element

        lastExceptionDetails[0] = excepTypeRoot.getName().toString();// Type of

        // LastException

        lastExceptionDetails[2] = "BIP"

                + excepTypeRoot.getFirstElementByPath("Number")

                        .getValueAsString();// Error Code


        return lastExceptionDetails;

    }


====================================================

    public void copyMessageHeaders(MbMessage inMessage, MbMessage outMessage)

            throws MbException {

        MbElement outRoot = outMessage.getRootElement();

        MbElement header = inMessage.getRootElement().getFirstChild();


        while (header != null && header.getNextSibling() != null) {

            outRoot.addAsLastChild(header.copy());

            header = header.getNextSibling();

        }

    }

============================================================


public class GetIPAddress {


public static String getIPString(){

    InetAddress thisInternet=null;

    try {

        thisInternet =InetAddress.getLocalHost();
        }
        catch(Exception e) {
                e.printStackTrace();
        }
        if (thisInternet != null)
                return thisInternet.getHostAddress();
        else
                return "Unable to get IPAddress ";
}

public static String getHostString(){
        InetAddress thisInternet=null;
        try {
                thisInternet =InetAddress.getLocalHost();
        }
        catch(Exception e) {
                e.printStackTrace();
        }
        if (thisInternet != null)
                return thisInternet.getHostName();
        else
                return "Unable to get HostName ";
}


}

============================================================
MbElement mbDomain = outMessage.getRootElement().getLastChild();
String jdbcProvider = (String) getUserDefinedAttribute("dataSource");
Connection con = null;
                try{
                        con = getJDBCType4Connection(jdbcProvider, JDBC_TransactionType.MB_TRANSACTION_AUTO);
        }catch(Exception e){
            String str = e.getMessage();
        }



        try {
            ts_intval=doSelectLNTS(con,mbDomain);
        } catch (SQLException e) {
            // TODO Auto-generated catch block
            throw new MbUserException("doSelectLNTS()", "Sql Exception while selecting from the table : "+e.getMessage(), "", "", "", null);
        }


------
private long doSelectLNTS(Connection con, MbElement element) throws MbException, SQLException{
        Timestamp ts=null;
        long ts_diff=0;
        String query = cnts.nts;
        String mgId = testu.getElementValue(element,cnts.N_P_MSG_ID);
        String tid = testu.getElementValue(element,cnts.N_TRANSACTION_ID);
        String erc = testu.getElementValue(element,cnts.N_ER_C);
        PreparedStatement pstmt = con.prepareStatement(query);

        pstmt.setString(1, mgId);
        pstmt.setString(2, tid);
        pstmt.setString(3, erc);
        ResultSet rs=pstmt.executeQuery();

        while(rs.next()){
            ts=rs.getTimestamp("n_ts");
        }

        if(null!= ts){
            java.util.Date  toDay=new java.util.Date();

            Timestamp tsToday= new java.sql.Timestamp(toDay.getTime());

            ts_diff=(tsToday.getTime() - ts.getTime())/(1000*60); // returns minutes

        }

        return ts_diff;

    }


    //public static final String nts="SELECT n_ts FROM sog " + "WHERE (mid=? AND tid=?) AND ec=?";


public String getCurrentGMTTS() {

                    String currentGMTTimstamp;

                    Format formatter;

                    Date date = new Date();

                    formatter = new SimpleDateFormat(BCs.GMT_TIME_FORMAT);

                    currentGMTTimstamp = formatter.format(date);

                    return currentGMTTimstamp;

    }


========================




import java.text.SimpleDateFormat;

import java.util.ArrayList;

import java.util.Calendar;

import java.util.Date;

import java.util.List;

import java.util.Map;


import com.ibm.broker.plugin.MbElement;

import com.ibm.broker.plugin.MbException;

import com.ibm.broker.plugin.MbService;

import com.ibm.broker.plugin.MbXPath;
```

```
public class testu
{

    public testu()
    {

    }

    private static void addURI(MbXPath str_xpath ) throws MbException
    {
        str_xpath
        .addNamespacePrefix(
                "abc",
                "http://www.xxx/wwrr/");
        str_xpath
        .addNamespacePrefix(
                "ns1",
                "http://www.xxx/wwrr");

        str_xpath
        .addNamespacePrefix(
                "cde",
                "http://www.xxx/wwrr");

        str_xpath
        .addNamespacePrefix(
                "SOAP-ENV",
                "http://schemas.xmlsoap.org/soap/envelope/");
    }


    @SuppressWarnings("unchecked")
```

```
public static String getElementValue(MbElement message, String xpath) throws MbException
{

    MbXPath str_xpath = new MbXPath(xpath, message);

    addURI(str_xpath);

    List<MbElement> listOfMbElements= (List<MbElement>) message.evaluateXPath(str_xpath);

    String elementValue = new String ("");


    if (listOfMbElements != null &&  listOfMbElements.size() > 0 )
    {

        // get last element in the list
        MbElement anElement = (MbElement) listOfMbElements.get(listOfMbElements.size()-1);

        if ( anElement != null)
        {
            elementValue = (String) anElement.getValue();
        }
    }

    return elementValue;
}


@SuppressWarnings("unchecked")
public static byte[] getElementValueAsByteArray(MbElement message, String xpath) throws MbException
{

    MbXPath str_xpath = new MbXPath(xpath, message);
    addURI(str_xpath);

    List<MbElement> listOfMbElements= (List<MbElement>) message.evaluateXPath(str_xpath);

    // hard coded to 1000.
    // byte[] bValue = new byte[1000];

    byte[] bValue = null;

    if (listOfMbElements != null &&  listOfMbElements.size() > 0 )
    {
        // get last element in the list
        MbElement anElement = (MbElement) listOfMbElements.get(listOfMbElements.size()-1);

        if ( anElement != null)
        {
            bValue = (byte[]) anElement.getValue();
        }
    }

    if ( null == bValue)
    {

        bValue = new byte[0];
    }

    return bValue;
}
```

```
private static void addVariable(MbXPath str_xpath, String elementValue) throws MbException

{
        str_xpath.assignVariable( "elementValue", elementValue );
}


public static void setElementValue(MbElement message, String xpath, String elementValue) throws MbException
{

        MbXPath str_xpath = new MbXPath(xpath, message);

        addURI(str_xpath);

        addVariable(str_xpath, elementValue);

        message.evaluateXPath(str_xpath);

}




@SuppressWarnings("unchecked")
public static List<MbElement> getAllElementsByXPath(MbElement message, String xpath) throws MbException
{

        MbXPath str_xpath = new MbXPath(xpath, message);

        addURI(str_xpath);

        List<MbElement> listOfMbElements= new ArrayList<MbElement>();
```

```
    }

    listOfMbElements= (List<MbElement>) message.evaluateXPath(str_xpath);

    return listOfMbElements;
}



@SuppressWarnings("unchecked")
public static Integer getElementIntValue(MbElement message, String xpath) throws MbException
{

    MbXPath str_xpath = new MbXPath(xpath, message);

    addURI(str_xpath);

    Integer elementValue = new Integer (0);

    List<MbElement>  listOfMbElements= (List<MbElement>) message.evaluateXPath(str_xpath);
    if (listOfMbElements != null &&  listOfMbElements.size() > 0 )
    {

        // get last element in the list
        MbElement anElement = (MbElement) listOfMbElements.get(listOfMbElements.size()-1);

        if ( anElement != null)
        {
            //elementValue = (Integer) anElement.getValue();
            elementValue = Integer.parseInt(anElement.getValue().toString());
        }
    }
    return elementValue;
}
```

```
  public static String asciiToHex(String ascii)
  {
  StringBuilder hex = new StringBuilder();

  for (int i=0; i < ascii.length(); i++)
  {
    hex.append(Integer.toHexString(ascii.charAt(i)));
  }
  return hex.toString();
}



  public static String overlay(String baseStr, String overlayStr, int begin, int end)
  {
 if (baseStr == null)
 {
   return null;
 }
 if (overlayStr == null)
 {
   overlayStr = "";
 }
 int len = baseStr.length();
 if (begin < 0)
 {
   begin = 0;
 }
 if (begin > len)
 {
   begin = len;
 }
 if (end < 0)
 {
   end = 0;
 }
 if (end > len)
 {
   end = len;
 }
 if (begin > end)
 {
   int temp = begin;
   begin = end;
   end = temp;
 }
 return new StringBuffer(len + begin - end + overlayStr.length() + 1)
   .append(baseStr.substring(0, begin))
   .append(overlayStr)
   .append(baseStr.substring(end))
   .toString();
   }

  public static String stringToHex(String base)
{
          StringBuffer buffer = new StringBuffer();

          int intValue;

          for(int x = 0; x < base.length(); x++)
          {
              int cursor = 0;
              intValue = base.charAt(x);
```

```
       String binaryChar = new String(Integer.toBinaryString(base.charAt(x)));

       for(int i = 0; i < binaryChar.length(); i++)
       {
         if(binaryChar.charAt(i) == '1')
        {
           cursor += 1;
         }
       }
       if((cursor % 2) > 0)
       {
         intValue += 128;
       }
       buffer.append(Integer.toHexString(intValue) + " ");

     }

     return buffer.toString();
}

  public static String convertHexStringToAsciiString(String stringDataAsHex)
  {
      StringBuilder sb = new StringBuilder();

      StringBuilder temp = new StringBuilder();

      for( int i=0; i<stringDataAsHex.length()-1; i+=2 )
      {

          String output = stringDataAsHex.substring(i, (i + 2));

          int decimal = Integer.parseInt(output, 16);
        sb.append((char)decimal);

        temp.append(decimal);
    }

    return sb.toString();
}


public static String  getRFH2ElementValueAsString(MbElement message, String xpath) throws MbException
{
    String elementValue = new String ("");

    MbElement element = message.getFirstElementByPath(xpath);

    if ( element != null)
    {
        elementValue = (String) element.getValue();
    }

    return elementValue;
}



public static int  getRFH2ElementValueAsInteger(MbElement message, String xpath) throws MbException
{
    int elementValue = 0;

    MbElement element = message.getFirstElementByPath(xpath);

    if ( element != null)
    {
        }



            elementValue = Integer.parseInt(element.getValue().toString());
        }

        return elementValue;
    }

    public static byte[] hexStringToByteArray(String s)
    {
        int len = s.length();

        byte[] data = new byte[len / 2];

        for (int i = 0; i < len; i += 2)
        {
            data[i / 2] = (byte) ((Character.digit(s.charAt(i), 16) << 4) + Character.digit(s.charAt(i+1), 16));
        }
        return data;
    }
    private static Map<String, String> map = System.getenv();

    public static String getEnv(String arg)
    {
        String envValue = new String("");

        if ( null != map.get(arg) )
        {
            envValue = (String) map.get(arg);
        }

        return envValue;
    }
}
```

```
================================================================

com.ibm.mq.constants.MQConstants

private void pteMQRFH2(MbMessage outMessage, int iRTh,String sQ) throws MbException

    {


        MbElement outRoot = outMessage.getRootElement();


        MbElement  bodyElement =outRoot.getLastChild();


        MbElement rfh2 = outRoot.getFirstElementByPath("/MQRFH2");


        boolean bRFH2Exists = false;


        if ( null == rfh2)

        {

            rfh2 = bodyElement.createElementBefore("MQHRF2");

        }

        else

        {

            bRFH2Exists = true;

            MbElement mbRElement = rfh2.getFirstElementByPath("usr/tst/re");


            if (null != mbRElement)

            {

                return;

            }

        }

        rfh2.createElementAsFirstChild(MbElement.TYPE_NAME_VALUE, "Version", new Integer(2));

        rfh2.createElementAsLastChild(MbElement.TYPE_NAME_VALUE, "Format",  MQConstants.MQFMT_STRING);




        if ( rfh2.getFirstElementByPath("jms") == null )

        {
```

```
    rfh2.createElementAsLastChild( MbElement.TYPE_NAME, "jms", null);
}
MbElement jmsFolder = rfh2.getFirstElementByPath("jms");
String chrRQ = (String) getUserDefinedAttribute(BCs.REPLAY_QUEUE);
jmsFolder.createElementAsLastChild(MbElement.TYPE_NAME_VALUE,"Dst", "queue:///" + chrRQ);

if ( rfh2.getFirstElementByPath("mcd") == null )
{
    rfh2.createElementAsLastChild( MbElement.TYPE_NAME, "mcd", null);
}
MbElement mcdFolder = rfh2.getFirstElementByPath("mcd");

if ( mcdFolder.getFirstElementByPath("Msd") == null )
{
    mcdFolder.createElementAsLastChild(MbElement.TYPE_NAME_VALUE, "Msd", "jms_text");
}
else
{
    mcdFolder.getFirstElementByPath("Msd").setValue("jms_text");
}



if ( rfh2.getFirstElementByPath("usr") == null )
{
    rfh2.createElementAsLastChild( MbElement.TYPE_NAME, "usr", null);
}
MbElement usrFolder = rfh2.getFirstElementByPath( "usr");
if ( usrFolder.getFirstElementByPath("tsu") == null )
{
    usrFolder.createElementAsLastChild( MbElement.TYPE_NAME, "tsu", null);
}

MbElement tsuFolder = usrFolder.getFirstElementByPath( "tsu");
```

```
            tsuFolder.createElementAsFirstChild(MbElement.TYPE_NAME_VALUE, "rCt", Integer.toString(iRTh));

            tsuFolder.createElementAsFirstChild(MbElement.TYPE_NAME_VALUE, "SQ",
outRoot.getFirstElementByPath("/MQMD/SourceQueue").getValueAsString());


            if ( !bRFH2Exists)

            {

                  tsuFolder.createElementAsLastChild(MbElement.TYPE_NAME_VALUE, "deleteRFH2Flag", "TRUE");

            }


            this.updateMQMD(outRoot);

      }


      // make it a datagram message, MQRFH2 format

      private void updateMQMD(MbElement root) throws MbException

      {

            // MQFMT_RF_HEADER_2

            root.getFirstElementByPath("/MQMD/Format").setValue(MQConstants.MQFMT_RF_HEADER_2);

      }

=====================Timer node use=============

----------------Iterate_On_Flag---------------------------

CREATE FUNCTION Main() RETURNS BOOLEAN

BEGIN

            SET Environment.Variables.Flag = TRUE;

            WHILE Environment.Variables.Flag DO

                  PROPAGATE;

            END WHILE;


            RETURN FALSE;

END;

----no message------Set_Flag_To_Terminate--------------------

BEGIN

            SET Environment.Variables.Flag = FALSE;
```

```
        RETURN TRUE;
END;
=================================================


public static void runInsert(Connection con,MbMessage outMessage){
byte[] inpmsg=outMessage.getBuffer();
byte[] outpmsg=outMessage.getBuffer().clone();

String insertString="insert into test(c1,c2) values(?,?) ";


        try{
                PreparedStatement psmt=conn.preparedstatement(insertString);
                psmt.setString(1,"testString");
                psmt.setBytes(2,outpmsg);
                psmt.setTimstamp(3,new Timstamp(Calender.getInstance().getTimeInMilliseconds));


                psmt.executeUpdate();
                psmt.close();


        }
        catch(SQLException e){
                throw new MbUserException(e.getClass().getName(),""SQL Exception whiletsting,"","","",new Object[]{":Exception Message"+e.getMessage() });
        }


}
********************************************************************
*****************************FIXDPMI.JAVA******************************
package com.kits.util;


import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileWriter;
import java.io.InputStreamReader;
import java.util.ArrayList;

public class FixDPMI {

        private static String HOME_DIR_LOCATION="C://DATA//DPMI//";
        private static String SOURCE_DIR="source";
        private static String DUSTBIN_DIR="dustin";
        private static String FINAL_DIR="final";
        private static String DUPLICATE_DIR="duplicate";

        private static boolean isDuplicateOrdersPresent;

        private static ArrayList<String> orderList;
        private static ArrayList<File> duplicateFileList;



        private static String ORDER_LIST_FILE_NAME="orders.txt";
        private static String DUPLICATE_FILE_LIST_FILE_NAME="duplicateFiles.txt";
        private static String UNATTENDED_FILE_LIST_FILE_NAME="unattened.txt";
        private static String FAULT_FILE_NAME="fault.txt";

        private static ArrayList<String> faultStringList;
        private static ArrayList<String> correctStringList;
        private static ArrayList<String> reportStringList;
        private static ArrayList<String> fileWithKnownFaultStringList;
        private static ArrayList<String> allFileList;
        private static ArrayList<File> filesModified;

        public static void main(String[] args) {
```

```
		loadFaultAndCorrectList();

		recordAllOrderNumber();

		listDuplicateFiles();


		if(isDuplicateOrdersPresent){

			removeDuplicateFiles();

			recordAllOrderNumber();

		}


		reportFailureCategory();

		reportUnattendedFiles();


		replaceFaultString();


		//removeReplacedFiles();

	}


	private static void removeReplacedFiles(){

		try{

			for(File file : filesModified){

				if(!file.renameTo(new File(HOME_DIR_LOCATION+"//"+DUPLICATE_DIR+"//" + 
file.getName()))){

					System.out.println(file.getName() + " is not moved to dustbin");

				}


			}

		}catch(Exception e){

			e.printStackTrace();

		}

	}


	private static void replaceFaultString(){

		try{
```

```
            if(null != filesModified && filesModified.size() > 0){
                for(File file : filesModified){
                    BufferedReader br1 = null;
                    FileInputStream fis1 = null;
                    String strLine1 = null;
                    StringBuilder bld = new StringBuilder();
                    fis1 = new FileInputStream(file);
                    br1 = new BufferedReader(new InputStreamReader(fis1));
                    String tempStr = null;
                    String strLine = null;
                    boolean isDone = false;
                    while ((strLine = br1.readLine()) != null){
                        for(String fault : faultStringList){
                            if(strLine.contains(fault)){
                                tempStr = strLine.replaceAll(fault, correctStringList.get(faultStringList.indexOf(fault)));
                                bld.append(tempStr);
                                isDone = true;
                            }
                        }
                        if(!isDone){
                            bld.append(strLine);
                        }
                    }

                    FileWriter fw = new FileWriter(new File(HOME_DIR_LOCATION+"//" +FINAL_DIR+"//" +file.getName()));
                    BufferedWriter bw = new BufferedWriter(fw);
                    bw.write(bld.toString());
                    bw.close();

                    //System.out.println(file.delete());
                }
            }

        }catch(Exception e){
            e.printStackTrace();
        }
    }

    private static void reportUnattendedFiles(){
        try{
            if(null != fileWithKnownFaultStringList && fileWithKnownFaultStringList.size() > 0){
                allFileList.removeAll(fileWithKnownFaultStringList);
                if(allFileList.size() > 0){
                    StringBuilder tmp = new StringBuilder();
                    for(String fileName : allFileList){
                        tmp.append(fileName);
                        tmp.append("\n");
                    }
                    if(allFileList.size() > 0){
                        FileWriter fw = new FileWriter(new
File(HOME_DIR_LOCATION+UNATTENDED_FILE_LIST_FILE_NAME));
                        BufferedWriter bw = new BufferedWriter(fw);
                        bw.write(tmp.toString());
                        bw.close();
                    }
                }
            }else{
                System.out.println("No unattended file present");
            }

        }catch(Exception e){
            e.printStackTrace();
        }
    }
    private static void reportFailureCategory(){
        File directory = new File(HOME_DIR_LOCATION+SOURCE_DIR);
        ArrayList<String> uniqueOrderList = new ArrayList<String>();
        duplicateFileList = new ArrayList<File>();
        int currentIndex ;
        StringBuilder faultRecords = new StringBuilder();
        allFileList = new ArrayList<String>();
        fileWithKnownFaultStringList = new ArrayList<String>();
        filesModified = new ArrayList<File>();
        try{
            if(directory.exists()){
                if(directory.isDirectory()){
                    BufferedReader br = null;
                    FileInputStream fis = null;
                    File[] fList = directory.listFiles();
                    String strLine = null;
                    for (int i=0 ; i < fList.length; i++){
                        allFileList.add(fList[i].getName());
                        fis = new FileInputStream(fList[i]);
                        br = new BufferedReader(new InputStreamReader(fis));
                        while ((strLine = br.readLine()) != null){
                            for(String fault : faultStringList){
                                if(strLine.contains(fault)){
                                    faultRecords.append(fList[i].getName()
+ " : " + reportStringList.get(faultStringList.indexOf(fault)));
                                    faultRecords.append("\n");

    fileWithKnownFaultStringList.add(fList[i].getName());
                                    filesModified.add(fList[i]);
                                }
                            }
                        }
                        fis.close();

                        br.close();

                    }

                }



                if(faultRecords.length() > 0){

                    FileWriter fw = new FileWriter(new
File(HOME_DIR_LOCATION+FAULT_FILE_NAME));

                    BufferedWriter bw = new BufferedWriter(fw);

                    bw.write(faultRecords.toString());

                    bw.close();

                }


            }else{

                System.out.println("---------Directory does not exists");

            }


        }catch(Exception e){

            System.out.println("Error while preparing list");

            e.printStackTrace();

        }

    }


    private static void removeDuplicateFiles(){

        StringBuilder tmp = new StringBuilder();

        try{

            for(File file : duplicateFileList){

                tmp.append(file.getName());

                tmp.append("\n");

            }

            if(duplicateFileList.size() > 0){
                FileWriter fw = new FileWriter(new
File(HOME_DIR_LOCATION+DUPLICATE_FILE_LIST_FILE_NAME));
                BufferedWriter bw = new BufferedWriter(fw);
                bw.write(tmp.toString());
                bw.close();
            }

            for(File duplicateFile : duplicateFileList){
                if(!duplicateFile.renameTo(new File(HOME_DIR_LOCATION +"//"+ DUPLICATE_DIR
+"//"+ duplicateFile.getName()))){
                    System.out.println("File moving failed for : " + duplicateFile.getName());
                }

            }
        }catch(Exception e){
            e.printStackTrace();
        }
    }


    private static void listDuplicateFiles(){
        File directory = new File(HOME_DIR_LOCATION+SOURCE_DIR);
        String orderString = "<PurchaseOrderHeader><oa:DocumentID><oa:ID>";
        ArrayList<String> uniqueOrderList = new ArrayList<String>();
        duplicateFileList = new ArrayList<File>();
        try{
            if(directory.exists()){
                if(directory.isDirectory()){
                    BufferedReader br = null;
                    FileInputStream fis = null;
                    String strLine = null;
                    File[] fList = directory.listFiles();
                    String orderNumber = null;
                    for (int i=0 ; i < fList.length; i++){
                        fis = new FileInputStream(fList[i]);
                        br = new BufferedReader(new InputStreamReader(fis));
                        while ((strLine = br.readLine()) != null){
                            if(strLine.contains(orderString)){
                                orderNumber = 
strLine.substring(strLine.indexOf(orderString)+43,strLine.indexOf(orderString)+51);
                                if(uniqueOrderList.contains(orderNumber)){
                                    duplicateFileList.add(fList[i]);
                                }else{
                                    uniqueOrderList.add(orderNumber);
                                }
                            }
                        }
                        fis.close();
                        br.close();
                    }
                }
                if(orderList.size() > uniqueOrderList.size()){
                    isDuplicateOrdersPresent = true;
                    System.out.println("There are : " + String.valueOf(orderList.size() - 
uniqueOrderList.size()) + " : extra orders");
                }else{
                    System.out.println("No duplicate orders");
                }
            }else{
                System.out.println("---------Directory does not exists");
            }

        }catch(Exception e){
            System.out.println("Error while preparing list");
            e.printStackTrace();
        }
```

```
    }


    private static void recordAllOrderNumber(){
        File directory = new File(HOME_DIR_LOCATION+SOURCE_DIR);
        StringBuilder orderBld = new StringBuilder();
        String orderString = "<PurchaseOrderHeader><oa:DocumentID><oa:ID>";
        orderList = new ArrayList<String>();
        try{
            if(directory.exists()){
                if(directory.isDirectory()){
                    BufferedReader br = null;
                    FileInputStream fis = null;
                    String strLine = null;
                    File[] fList = directory.listFiles();
                    String orderNumber = null;
                    for (int i=0 ; i < fList.length; i++){
                        fis = new FileInputStream(fList[i]);
                        br = new BufferedReader(new InputStreamReader(fis));
                        while ((strLine = br.readLine()) != null){
                            if(strLine.contains(orderString)){
                                orderNumber =
strLine.substring(strLine.indexOf(orderString)+43,strLine.indexOf(orderString)+51);
                                orderList.add(orderNumber);
                                orderBld.append(orderNumber);
                                orderBld.append("\n");
                            }
                        }
                        fis.close();
                        br.close();
                    }
                }
            }else{
                System.out.println("---------Directory does not exists");
            }
            if(orderList.size() > 0){
                System.out.println("Total Order Count  : " + orderList.size());


                FileWriter fw = new FileWriter(new
File(HOME_DIR_LOCATION+ORDER_LIST_FILE_NAME));
                BufferedWriter bw = new BufferedWriter(fw);
                bw.write(orderBld.toString());
                bw.close();
            }
        }catch(Exception e){
            System.out.println("Error while preparing list");
            e.printStackTrace();
        }


    }


    private static void loadFaultAndCorrectList(){
        faultStringList = new ArrayList<String>();
        correctStringList = new ArrayList<String>();
        reportStringList = new ArrayList<String>();
        try{
            faultStringList.add("KÄRCHER");
            faultStringList.add("DECK²");
            faultStringList.add("º");
            faultStringList.add(" – ");
            faultStringList.add("£");
            faultStringList.add("®");
            faultStringList.add("Nüssli");
            faultStringList.add("n’t");


            correctStringList.add("KARCHER");
            correctStringList.add("DECK");
            correctStringList.add(" ");
            correctStringList.add(" ");
            correctStringList.add("Pound ");
            correctStringList.add(" ");
            correctStringList.add("Nussli");
            correctStringList.add("not");

            reportStringList.add("KARCHER");
            reportStringList.add("DECK");
            reportStringList.add("Power");
            reportStringList.add("Hyphen");
            reportStringList.add("Pound");
            reportStringList.add("RoundR");
            reportStringList.add("Nussli");
            reportStringList.add("not");
            /*BufferedReader br = null;
            FileInputStream fis = null;
            String tmp = null;
            fis = new FileInputStream(new File(HOME_DIR_LOCATION+FAULT_FILE_NAME));
            br = new BufferedReader(new InputStreamReader(fis));
            while ((tmp = br.readLine()) != null){
                StringTokenizer tkn = new StringTokenizer(tmp, ";");
                while(tkn.hasMoreElements()){
                    System.out.println(tkn.nextElement().toString());
                    System.out.println(tkn.nextElement().toString());
                }
            }*/
        }catch(Exception e){

        }
    }
}
```

```
****************************************************************************************
*************FAULTY_TXN_CHK********************************************
###########################################################
######## UNIX SCRIPT TO CHECK FAULTY TRANSACTION DATA FROM IBODS/MBODS:SALES_TRANSACTIONS_TENDER
#########
######## AUTHOR: ABHIJIT KARMAKAR #######################
######## VERSION: 1.0 DATE: 17-APR-2014 ##################
###########################################################
#!/bin/ksh


cd /bq/home/mqm/scripts/PCI_CHK_Scripts
LOGFILE=./log/`date +\%Y\%m\%d`_PCI_FAULTY_TRANSACTION_CHK.log


export ORACLE_HOME=/ibpr1/oracle/10.2_64
export ORACLE_SID=IBPR1


exec 2>&1 > $LOGFILE
echo "Job runing at: `date`......"
trim() {
   local var=$@
   var="${var#"${var%%[![:space:]]*}"}"   # remove leading whitespace characters
   var="${var%"${var##*[![:space:]]}"}"   # remove trailing whitespace characters
   echo "$var"
}
Zero_result=0
row_count=`$ORACLE_HOME/bin/sqlplus -s CWODS/cwods052k03d@IBPR1 << EOF
set pagesize 0 feedback off verify off heading off ;
SELECT COUNT(*) FROM CWTOWNER.SALES_TRANSACTIONS_TENDER WHERE STORE_CODE IN ('0186','0182') AND TOKEN IS
null AND SETTLED='Y';
exit;
```

```
EOF`
echo "\n##########################"
echo "JOBSQL Status: Successful"
row_count=`trim "$row_count"`
echo "Faulty PCI Transaction count: $row_count"
if [ "$row_count" = "$Zero_result" ]
 then
        echo "Row Returned Zero"
        exit 0;
else
        echo "Row Returned NON-Zero:'$row_count'"
        exit 1;
fi
echo "\n##########################"
echo "Job Ended at: `date`......"



******************************************************************
********************purgeMBRecord.sh*************************
###########################################################
######## UNIX SCRIPT TO PURGE MBRECORD(RECORD & REPLAY) DATA FROM DATABASE #########
######## AUTHOR: ABHIJIT KARMAKAR #######################
######## VERSION: 1.0 DATE: 24-OCT-2013 ##################
###########################################################
#!/bin/ksh -x

cd /var/mqm/scripts/PURGEScripts
LOGFILE=./log/`date +\%Y\%m\%d`_Purging_mbrecord.log
VAR_HOSTNAME=`hostname`

if [ "$#" -lt 3 ] || [ "$#" -eq 0 ]
then
```

```
if [ `echo $VAR_HOSTNAME | grep -c "0376" ` -gt 0 ]
 then
  echo "executing in unxs0376.uk.b-and-q.com"
  exec 2>&1 > $LOGFILE
  export ORACLE_HOME=/oracle/TMBU1/11203_64
  export ORACLE_SID=TMBU1
else
  echo "executing in unknown host"
  exit 2
fi

RETAIN_DAYS=32
DELETE_FLAG="Y"
BATCH_LIMIT=1000
#LOGFILE="`date +\%Y\%m\%d`_Purging_mbrecord.log"
exec 2>&1 > $LOGFILE
echo " Job runing at: `date`......"
echo "Invalid number of arguments OR 'NO' arguments.But this is running with Default args with DELETE_FLAG='Yes' now."
echo "MBRECORD.KF_PURGE($RETAIN_DAYS,'$DELETE_FLAG',$BATCH_LIMIT);"
$ORACLE_HOME/bin/sqlplus MBREPOS/MBR3P0S <<EOF
SET SERVEROUTPUT ON;
exec MBRECORD.KF_PURGE($RETAIN_DAYS,'$DELETE_FLAG',$BATCH_LIMIT);
exit
EOF

echo "Job ran with arguments-->>RETAIN_DAYS=$RETAIN_DAYS,DELETE_FLAG="$DELETE_FLAG",BATCH_LIMIT=$BATCH_LIMIT"
echo "Job ended at: `date`......"
echo "######################################################"



fi
```

```
*******************************************************************

dmpmqcfg -m UNXS0383.AGT.QM1 -a > UNXS0383.AGT.mqsc

dmpmqaut -m UNXS0383.AGT.QM1 -l > UNXS0383.AGT.mqsc

saveqmgr.aix -m TEST_QM -f TEST_QMqmgr_data.mqsc -z TEST_QMqmgr_auth.sh

uncompress -fv ms03_unix.tar.Z

tar -xvf ms03_unix.tar

mqsireportresourcestats UNXS0389.MB1 -e MDAT

mqsiwebuseradmin UNXS0383.MB1 -l

mqsichangeresourcestats UNXS0383.MB1 -e SWEB -c inactive -v trace.txt

mqsichangeresourcestats UNXS0383.MB1 -e SWEB -c active -v trace.txt

*******************************************************************
```