BASIC COMMANDS:

To List out all the Brokers created in the current installation

```
mqsilist
```

To create the Broker

```
mqsicreatebroker {BROKERNAME} -q {QMGRNAME} -i {USERNAME} -p {Password}
```

To create Execution Group

```
mqsicreateexecutiongroup  {BROKERNAME} -e {EGName}
```

To start Execution Group

```
mqsistartmsgflow {BROKERNAME} -e {EGName}
```

To stop Execution Group

```
mqsistopmsgflow {BROKERNAME} -e {EGName}
```

To delete Execution Group

```
mqsideleteexecutiongroup -n {BROKERNAME} -e {EGName}
```

To specify Debug Port for EG

```
mqsichangeproperties {BROKERNAME} -e default -o ComIbmJVMManager -n
jvmDebugPort -v 8117
```

To List out all the deployed objects under Execution Group

```
mqsilist {BROKERNAME} -e default -k myApplication
```

To List out all the deployed objects that are configured Library

```
mqsilist {BROKERNAME} -e default -k myApplication -y {myEGLibraryName}
```

To return detailed information about Application

```
mqsilist {BROKERNAME} -e default -k myApplication -d2
```

To lists all deployed objects that are configured in myApplication

```
mqsilist {BROKERNAME} -e default -k myApplication -r
```

To List out a summary of the EG that are defined on a broker

```
mqsilist {BROKERNAME}
```

To display detailed info about all resources for brokers on Local System

```
mqsilist -a -r -d2
```

MONITORING COMMANDS:

To activate the Monitoring

```
mqsichangeflowmonitoring {BROKERNAME} -e default -k {ApplicationName} -f
{FlowName} -c active
```

To report the Monitoring

```
mqsireportflowmonitoring {BROKERNAME} -e default -k {ApplicationName} -f
{FlowName} -a
```

SECURITY IDENTITY COMMANDS:

To start the Broker

```
mqsistart {BROKERNAME};
```

To stop the Broker

```
mqsistop {BROKERNAME};
```

To register DSN with IIB

```
mqsisetdbparms {BROKERNAME} -n {DSName} -u {SchemaName} -p {Password};
```

To know whether Broker is associated with DSN or Not

```
mqsicvp {BROKERNAME} -n {DSName}
```

To give security for FTP

```
mqsisetdbparms {BROKERNAME} -n ftp::{SeuID} -u {SchemaName} -p {Password};
```

To give security for SMTP(Email Receiving)

```
mqsisetdbparms {BROKERNAME} -n smtp::{SeuID} -u {emailid} -p {Password};
```

To give security for Email Sending

```
mqsisetdbparms {BROKERNAME} -n email::{SeuID} -u {emailid} -p {Password};
```

To give security for JDBC Configurable Service

```
mqsisetdbparms {BROKERNAME} -n jdbc::{SeuID} -u {SchemaName} -p {Password};
```

MQSICHANGE PROPERTY COMMANDS:

To report the HTTP Listener Property at Broker Level

```
mqsireportproperties {BROKERNAME} -b httplistener -o HTTPConnector -a
```

To report the HTTP Listener Property at EG Level

```
mqsireportproperties {BROKERNAME} -e default -o HTTPConnector -a
```

To Change the HTTP Listener Port Number(Broker Level)

```
mqsichangeproperties {BROKERNAME} -b httplistener -o HTTPConnector -n port -v 7800
```

To change the HTTP Listener Port Number at EG Level

```
mqsichangeproperties {BROKERNAME} -e default -o HTTPConnector -n port -v 7800
```

To Trace the HTTPListener

```
mqsireportbroker {BROKERNAME}
```

NORMAL COMMANDS:

To start the Application

```
mqsistartmsgflow {BROKERNAME} -e {EGName} -k {ApplicationName}
```

To stop the Application

```
mqsistoptmsgflow {BROKERNAME} -e {EGName} -k {ApplicationName}
```

To delete the Application

```
mqsideploy {BROKERNAME} -e {EGName} -d {ApplicationName}
```

To know the Deployment Status

```
mqsilist {BROKERNAME} -e {EGName} -d 2
```

To deploy the BAR

```
mqsideploy {BROKERNAME} -e {EGName} -a {BARFileName}
```

To delete the BAR

```
mqsideploy {BROKERNAME} -e {EGName} -d {BARFileName}
```

To read the BAR

```
mqsireadbar -b {BARFileName} -r
```

Example:

```
mqsireadbar -b C:\IIBWorkspace\DTPTibcoConn\BARfiles\DA_PersistUWSInfo_integrationProd_prod_v1_1.bar -r
```

BAR Override Command

```
mqsiapplybaroverride -b {BARFileName} -k {ApplicationName} -m {MessageFlowName}#{Property to change}
```

Example:

```
mqsiapplybaroverride -b C:\IIBWorkspace\iib9\BARfiles\emp.bar -k Test12App -m Test12Flow#TABLE=DEPT
```

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

MqsiReadBar Command

————————————————-

1) mqsireadbar -b <barfilename> (name of the bar file to be read)

2) mqsireadbar -b <location of barfilename> > <location of propertiefile> -r (Run the coammnd recursively content of application and libria is display)

Mqsiapplybaroverride Command

——————————————————

1)mqsiapplybaroverride -b <location of the bar file> -p <location of changed propetie file> -r

2)mqsiapplybaroverride -b <original.bar> -k application -p <location of changed bar filename> -r(-b bar file name,-k application name,-r recursivley content display)

3)mqsiapplybaroverride -b myflow.bar -k application -y <libraryfilename> -p myOtherBroker.xml(-p property file name)

mqsideploybar command

——————————————————-

1)mqsidepolybar <brokername> -e <Executiongroup> -a <barfilename> (-a barfileapplicationname)

2)mqsidepolybar <brokername> -e <executiongroup> -d <barfilename>(-d for delete,-e execution group)

mqsistopmsgflow command
————————————-

```
1)mqsistopmsgflow  <brokername> -e <executionname> -k <applicationname>
2)mqsistopmsgflow  <brokername> -e <executionname> -k <applicationname> -m <msgflowname>
3)mqsistopmsgflow  <brokername> -e <executiongroupname> -m <myFlowname> -f <restartExecutionGroup>(-f for restart the execution groupname)
```

mqsistartMsgflow command
————————————

```
1)mqsistartmsgflow  <brokername> -e <executiongroupname> -k <applicationname>
2)mqsistartmsgflow  <brokername> -e <executionname> -k <applicationname> -m <msgflowname>
3)mqsistartmsgflow  <brokername> -e <executiongroupname> -m <myFlowname> -f <restartExecutionGroup>(-f for restart the execution groupname)
```

mqsichangeproperties command
———————————————-

```
1)mqsichangeproperties <brokername> -b httplistener -o HTTPListener -n startListener -v false(disable http port,-o object,-v value,-n component,-b property name)
2)mqsichangeproperties <brokername> -b httplistener -o HTTPListener -n startListener -v true(enable http port )
3)mqsichangeproperties <brokername> -o ComIbmJVMManager -n jvmMaxHeapSize -v size_in_bytes(to change jvm heap size)
4)mqsichangeproperties <brokername> -e <ExecutionGroup> -o ComIbmJVMManager -n jvmDebugPort -v 8018
5)mqsichangeproperties <brokername> -b httplistener -o HTTPListener -n port -v 7843
6)mqsichangeproperties BRKR -o BrokerRegistry -n brokerKeystoreFile -v /tmp/mb7brokerkeystore1.jks  (To add a keystore to the Broker)
```

Mqsibackup Command
——————————

```
1)mqsibackupbroker <brokername> -d <filedirectorylocationpath> -v <pathfilename>
2)mqsirestorebroker <brokername> -d <filedirectorylocatiopath> -a
```

<zipfilelocation>

mqsireportproperties command
———————————

1)mqsireportproperties <brokername> -b httplistener -o HTTPListener -a(Display all the current HTTPListener settings associated with HTTP and SOAP nodes)

2)mqsireportproperties <brokername> -b httplistener -o HTTPListener -n startListener(Check if the broker-wide listener is active for deployed HTTP and SOAP nodes)

3)mqsireportproperties <brokername> -b cachemanager -o CacheManager -r(Display the properties for the cache manager)

4)mqsireportproperties <brokername> -b httplistener -o HTTPSConnector -n port(Displays httpsconnector ports)

5)mqsireportproperties <brokername> -c JDBCProviders -o Oracle -r(Report the properties of the Oracle JDBCProvider configurable service)

6)mqsireportproperties <brokername> -o brokerregistry -r

mqsisetdbparms command
—————————————–

1)mqsisetdbparms <brokername> -n <DSNNAME> -u userID -p password(For setting database)

2)mqsisetdbparms <brokername> -n smtp::mySecurityIdentityObjectName -u myUserID -p myPassword(for setting SMTP SERVER)

3)mqsisetdbparms <brokername> -n jdbc::JDBC -u Username -p password(For setting jdbc database)

4)mqsisetdbparms <brokername> -n ftp::identityName -u user1 -p MyPassword(for setting ftp securityidentity)

5)mqsisetdbparms <brokername> -n sftp::identityName -u user1 -p MyPassword(for setting sftp securityidentity)

mqsireadlog command
—————————————————–

1)mqsireadlog <brokername> -t -b services -f -o <pathofoutputfilename>

2)mqsifromatelog -i <locationofinputfilename> -o <locationofoutputfilename>

othercommands
————-

1)mqsideleteexecutiongroup <brokername> -e <executiongroupname>

2)mqsicreateexecutiongroup <brokername> -e <executiongroupname>

3)mqsilist <brokername> -e <executionname>

4)mqsistopbroker -i <brokername>

5)mqsistartbroker <brokername>
6)mqsicreatebroker <brokername> -q <queuemanager>
7)mqsideletebroker <brokername>
8)mqsicvp <brokername> -n <servicename>
9)mqsilist brokername -d2(To get all execution group Process id and running message flows)

TO set the log files in iib
——————————————-
1)Go to /var/logs create user.log and giver full permissions
2) go to /etc/rsys.log
set user.info /var/logs/user.log

MQCOMMAND
———

Application trigger

3)Three types of trigger(every,first,depth)
———————-
1)ALTER ql(QM1.LQ) TRIGGER TRIGTYPE(EVERY) INITQ(SYSTEM.DEFAULT.INITIATION.QUEUE) PROCESS(PROC)
2)switch to main root and create the script in location /tmp/filename(Ex /tmp/ashok.txt)
3)Define or open ashok.txt insert the follwing command
/opt/mqm/samp/bin/amqsget QM1.LQ QM1 > /tmp/filename(Ex /tmp/sama.txt)
apply full permissions chmod -R 777 ashok.txt
4) define process in runmqsc qm1
command::define process(proc) appltype(unix) applicid('/tmp/ashok.txt') (Appliction id is nothing but script location)
5)su to mqm and run the command runmqtrm -m QM1 -q SYSTEM.DEFAULT.INITIATION.QUEUE.

Channel trigger
—————————-
1)Create one way or two way commincation
2)Dont start the sdr channel
3)Three types of trigger(every,first,depth)
4)Alter the tranmission queue
command::ALTER QL(QM2.TQ) TRIGGER TIGTYPE(EVERY) TRIGDATA(QM2.TO.QM3) INITQ(SYSTEM.CHANNEL.INITQ) USAGE(XMITQ)

SSL on two way commuincation
————————————————————-
1)First completed the two way commication
2)dis qmgr all (it display all properties of queue manager)
3)ALTER QMGR SSLKEY('/var/mqm/qmgrs/QM1/ssl/qm1') SSLEV(enable)(it is applicable to QM1 queue manager)
4)same apply for QM2 queue manager also ALTER QMGR SSLKEY('/var/mqm/qmgrs/QM1/ssl/qm2') SSLEV(enable)
5)open new tab switch to mqm user go to these location (/opt/mqm/java/jre64/jre/bin) {QM1 queue manager}
6)enter command ./ikeyman
7)open new tab switch to mqm user go to these location (/opt/mqm/java/jre64/jre/bin) {QM2 queue manager}
8)enter command ./ikeyman
9)alter both QM1 and QM2 sender and reciver channels
10) alter channel(QM1.TO.QM2) CHLTYPE(SDR) TRPTYPE(TCP) SSLCIPH(TLS_RSA_WITH_AES_128_CBC_SHA256)
11) ALTER CHANNEL(QM2.TO.QM1) CHLTYPE(RCVR) TRPTYPE(TCP) SSLCIPH(TLS_RSA_WITH_AES_128_CBC_SHA256)
— Do this three steps in BOTH Qqueue manager QM1 and QM2—-
12)stop channel(senderchannel)
13)refresh security type(ssl)
14)start channel(senderchannel)
Client server communication
——————————————
1)create queue manager
2)create listener
3)create local queue
4)create server connection channel
command::DEFINE CHANNEL(TO.QM3) CHLTYPE(SVRCONN) TRPTYPE(TCP) MCAUSER('mqm')
SET AUTHREC PROFILE(LocalQueuename) OBJTYPE(QUEUE) PRINCIPAL('test') AUTHADD(PUT,GET)
SET AUTHREC OBJTYPE(QMGR) PRINCIPAL('test') AUTHADD(CONNECT)
SET CHLAUTH(S.TO.C) TYPE(ADDRESSMAP) ADDRESS('192.168.1.37') MCAUSER('test')
5)setting athentication for channel
command:set channelauth(*) type(blockuser) userlist('nobody','mqm')
set channelauth(To.QM3) type(blockuser) userlist('nobody')
6)create user test
7)vi .bash_profile

```
8)EXPORT MQSERVER=TO.QM3/TCP/’ipaddress(portnumber)’
```

ArquitecturaIBM Consulting