Message Broker administration for dummies

David Coles – WebSphere Message Broker Level 3 Technical Lead,
IBM Hursley – _dcoles@uk.ibm.com_

Monday 8th August 2011

Agenda

- Introduction & Message Broker Recap
- Message Broker v7
- Message Broker Runtime Internals
  - Configuration Data
  - Runtime Environment
  - Userids
- Tools for Administration
  - Connecting to a Broker
  - Command Line
  - Message Broker Toolkit
  - Message Broker Explorer
  - Administration API (CMPAPI)
- Administrative Security
- Broker Backup/Restore
- Multi-instance Brokers (HA)
- Migrating to v7
- Configurable Services
- Problem Determination

Introduction & Message Broker Recap
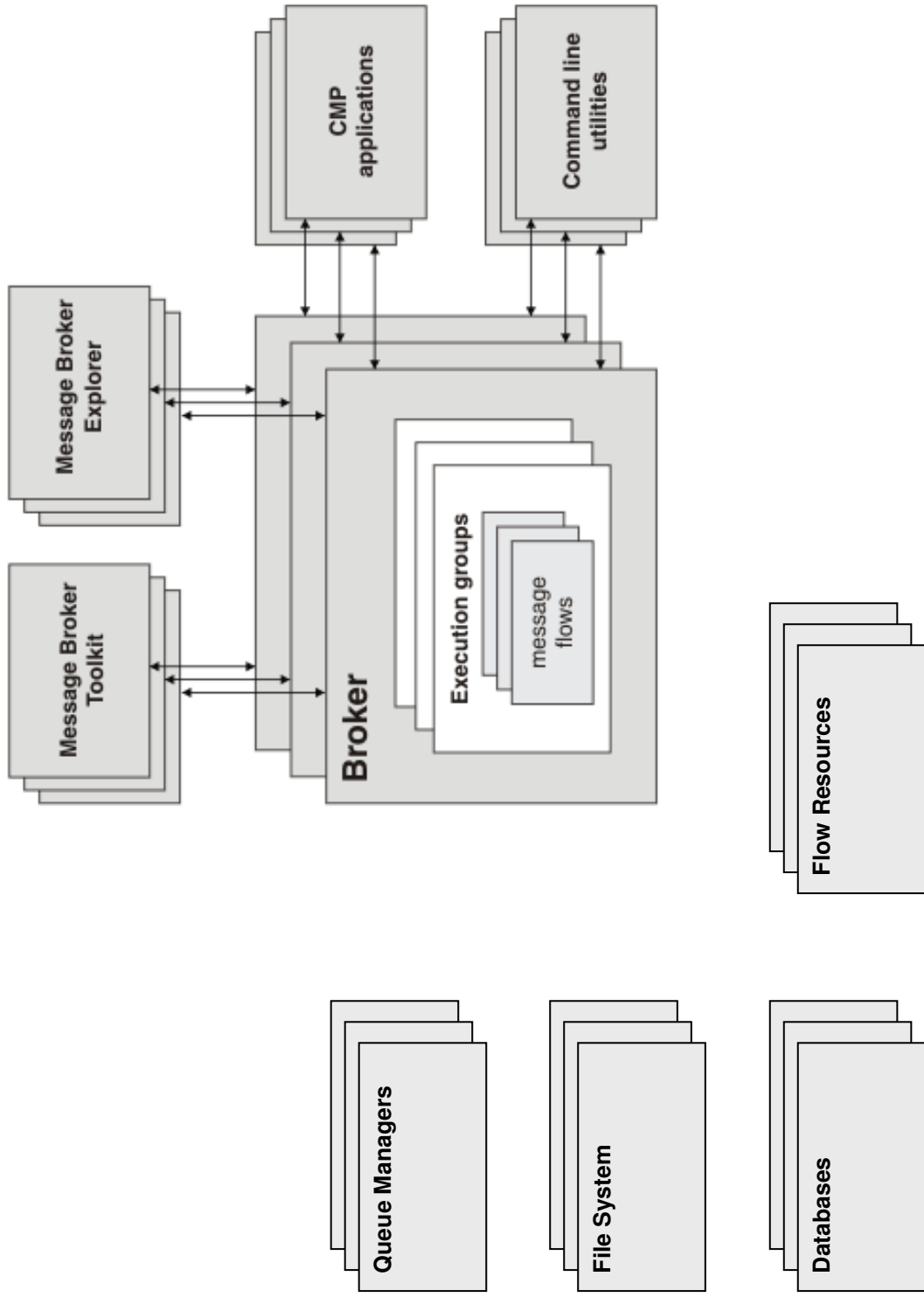
Introduction & Message Broker Recap - Notes

- A broker is a set of execution processes that hosts one or more message flows to route, transform, and enrich in flight messages.

- Administering brokers and associated broker resources includes the tasks that you perform frequently to activate and manage those resources. Choose the method you prefer to administer your brokers and associated resources.

- Administration of brokers includes the following tasks:
  - Managing brokers
  - Managing execution groups
  - Managing message flows
  - Developing applications that use the Administration API
  - Accessing Administration log information
  - Changing the location of the work path
  - Managing resources used by brokers
  - Backing up resources
  - Administering Java applications

- These tasks can be performed by using one, or more, of the administrative techniques supported by WebSphere Message Broker:

- Administer the broker by using the WebSphere Message Broker Explorer, the Broker view in the WebSphere Message Broker Toolkit, or the product commands. Alternatively, you can write your own programs to use the Message Broker Administration API (also known as the CMP API).

- Manage the application resources of the broker, which include message flows and message sets, by using the WebSphere Message Broker Toolkit or WebSphere Message Broker Explorer; these two applications connect to the broker by using a WebSphere MQ server connection, which is defined to the broker queue manager when you create the broker.

- The picture on the previous page shows the relationship between the resources that exist at run time, and how they interact with the WebSphere Message Broker Explorer and WebSphere Message Broker Toolkit.

Message Broker v7

- Making administration easier was a huge focus area in WMB V7
  - V7 Themes
    - Simplicity and Productivity
    - Universal Connectivity for SOA
    - Dynamic Operational Management
    - Platforms, Environments and Performance

- Key administration enhancements
  - Config Manager removal
 - No database pre-req
  - Message Broker Explorer

Message Broker V7.0.0.0 Recap

Simplicity and Productivity

- Radically streamlined product prerequisites and components
- Simplified connectivity solution development using IBM pre-supplied patterns
- Impact Analysis to manage development artefact changes including ESQL, Maps and Message sets
- MB Explorer for dedicated administration tooling
- SCA nodes for WPS Interoperability

Universal Connectivity for SOA

- Extended & integrated publish subscribe: common management & security with new MQ capabilities
- PHP nodes for Web 2.0 support
- Enhanced SAP, Siebel, PeopleSoft packaged application support
- New Sequence and Resequence nodes

Dynamic Operational Management

- New operational facilities for audit and monitoring, including WBM
- Enhanced statistics to understand broker performance, including memory usage
- Improved user trace to easily understand message flow behaviour
- Enhancements for WSRR processing including support for FSM protocol
- Support and Exploit MQ Multi-instance Queue Managers for High Availability

Platforms, Environments and Performance

- Exclusively 64bit Broker support
- Performance monitoring tools and very reduced memory footprint
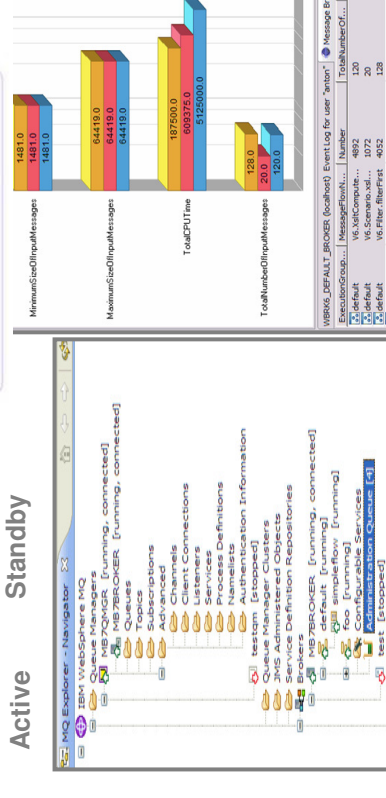
Message Broker V7.0.0.1 Recap

Simplicity and Productivity
- User Defined Patterns for custom reuse
- User Defined Sub flows: encapsulate & distribute
- Expanded Patterns Explorer

Universal Connectivity for SOA
- SOAP/JMS & more Web Service enhancements
- Database input node processing of relational data
- Multi-platform CICS node for direct connectivity
- FTE file nodes for end-to-end file processing
- CORBA request node to facade CORBA systems

Dynamic Operational Management
- SAML, Kerberos, LTPA and RACF pass tickets
- PEP node for mid-flow security processing
- Comprehensive operational resource statistics
- Web Services Policy Analytics for WSRR

Platforms, Environments and Performance
- Windows 7, Server 2008 with 64 bit processes
- More databases: solidDB, SQL Server z/Linux

SOAP Input   SOAP Request   SOAP Reply

Database Input   FTE Input   File Output

CORBA Request   CICS Request   Security PEP

Windows 7

name	InitialMemoryInMB	UsedMemoryIn...	CommittedMemoryInMB	MaxMemor
summary	32	23	46	-1
Heap Memory	32	13	32	256
Non-Heap Mem...	0	10	14	-1
Garbage Collect...				

Message Broker V7.0.0.2 Recap

Simplicity and Productivity

- Patterns Refinement to create highly customizable user defined patterns
- Patterns Communities for packaging, sharing, uploading and rating
- Tooling enhancements for Mapping, Unit Test & Debugging

Universal Connectivity for SOA

- Async and transactional SOAP/JMS; New JSON parser for Web 2.0
- File Read node and other file processing enhancements
- Email input node to retrieve data from POP and IMAP mail systems
- JD Edwards nodes to extend ERP processing, and other ERP node enhancements
- CICS and TCP/IP node enhancements
- Database input node enhancements for code-free query and WBIA migration

Dynamic Operational Management

- WCA Hypervisor edition to simplify provisioning of new and updated brokers
- Web Services Gateway function for more manageable processing
- Per Execution Group Profiles for multi-tenancy configuration
- Resource Manager Statistics for parsers storage usage

Platforms, Environments and Performance

- Enhanced platforms: AIX 7.1, Oracle 11gR2, Informix XA,
- New support for Oracle AQ and JBoss JMS providers

Email Input

Cloudburst

JDE
JDEdwards Input

JDE
JDEdwards Request

File Read

Patterns Community

This site shows you how to build a comm nodes and much more!

The web site is built using Drupal, a ven

default Resources Statistics (Snapshot time 07:54:49 - 07:55:09)

name	Threads	ApproxMemKB	MaxReadKB	MaxWrittenKB	Parsers	Fie
summary	2	105761.03	0.41	7910.53	60
CREATELARGE.MQMD	1	15.97	0.36	0.43	2
CREATELARGE.MQROOT	1	55.89	0.38	0.00	7
CREATELARGE.Properties	1	15.97	0.38	0.00	4
CREATELARGE.XMLNSC	1	65639.55	0.03	7910.53	60
MAINYPARSERS.BLOB	1	15.97	0.05	0.05	6

CICS | CORBA | FTEAgent | FTP | File | JDBCConnectionPools | JVM | ODBC | Parsers

Configuration Manager Removal Benefits

- The broker environment will be a lot easier to manage

- One view of the world

- More information returned to tools

- Much improved connect and deploy times

- Long-standing niggles have been eliminated. V7 has:

  - One-step broker creation (i.e. no CM association step)

  - No "Deployment already in progress" messages

  - No CM/Broker Synchronization problems

    - Cancel Deployment

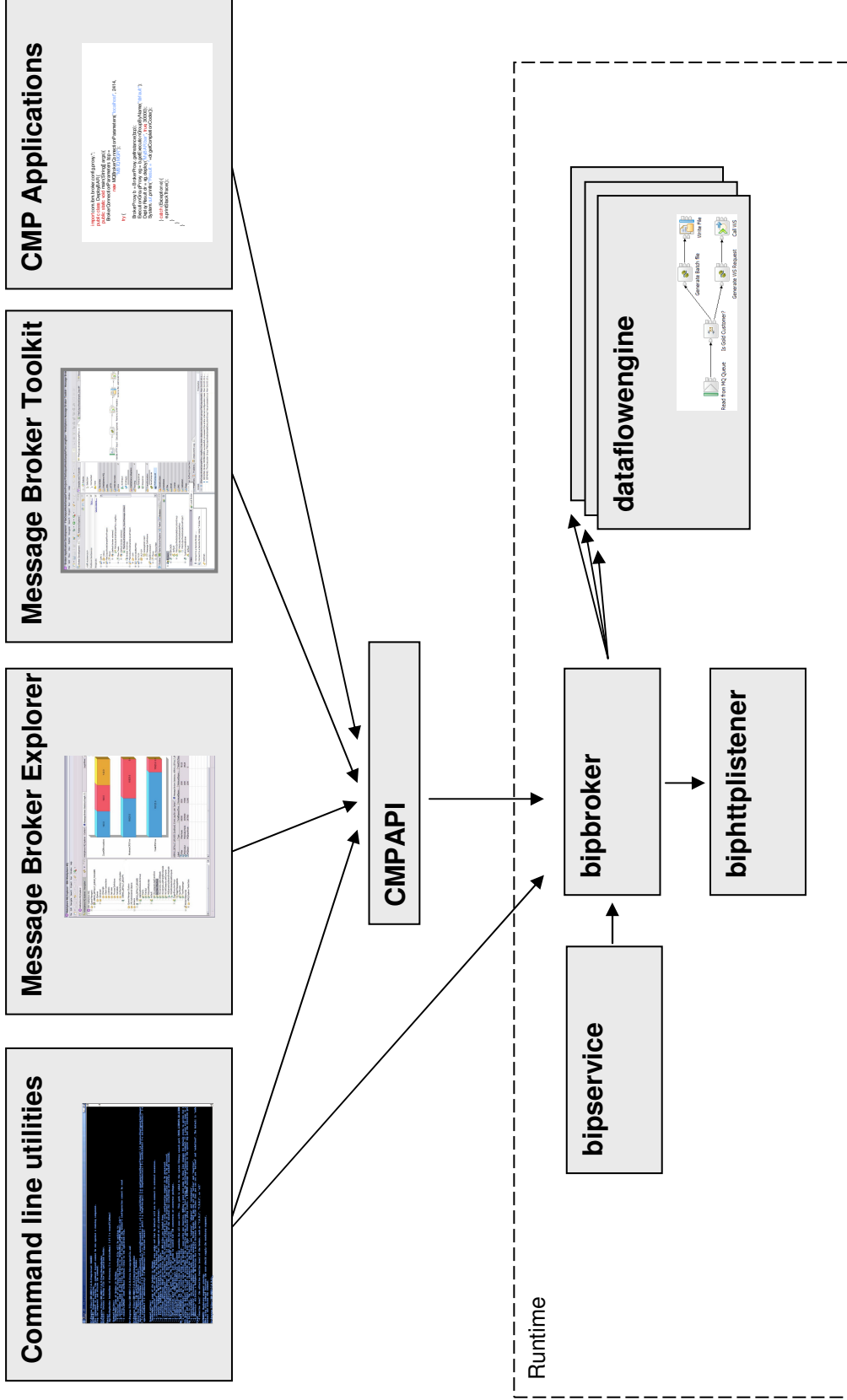   - Performance

Publish Subscribe v7

- WMB V7 uses WMQ 7.0.1 as its pub/sub engine
  - Common topic space
  - Publication node uses WMQ
  - Content-based pub/sub handled by WMB
    - For example, `<publish> if msg.price > 100`
  - New 'noMatch' terminal when no subscribers

in → out

no match

Publication
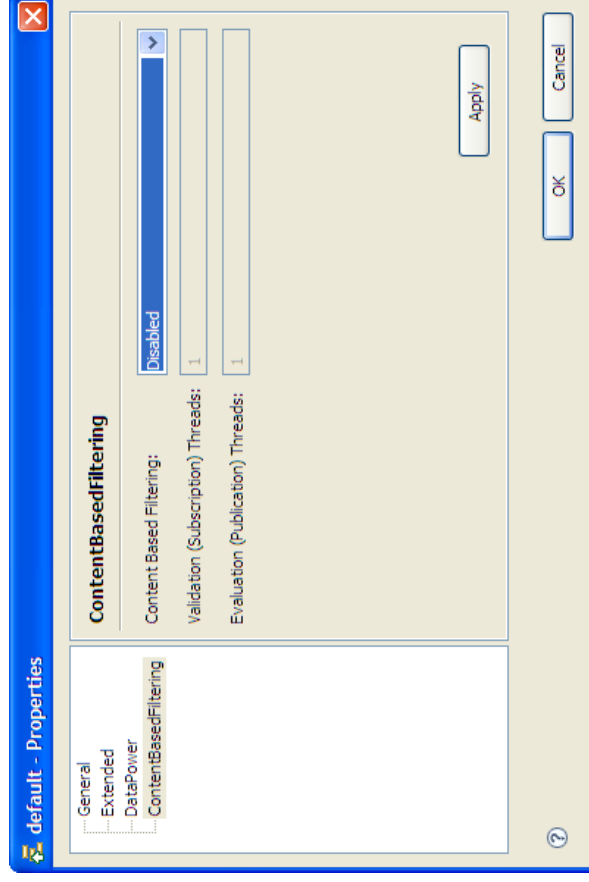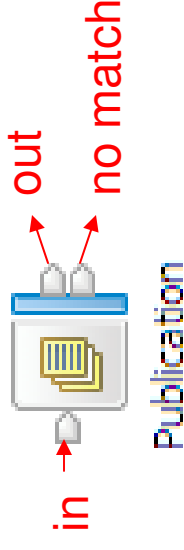
- Default Execution Group
  - Used in v6.x to handle pub/sub engine
  - No default execution group in v7
  - Configure which execution groups (if any) you would like to handle content-based filters

default - Properties

General
Extended
DataPower
ContentBasedFiltering

ContentBasedFiltering

Content Based Filtering: Disabled

Validation (Subscription) Threads: 1

Evaluation (Publication) Threads: 1

OK   Cancel

Apply

Message Broker Runtime Internals

Command line utilities

Message Broker Explorer

Message Broker Toolkit

CMP Applications

CMP API

Runtime

bipservice

bipbroker

biphttplistener

dataflowengine

Message Broker Runtime Internals - Notes

- Several processes make up the Message Broker runtime environment

  - bipservice

    - Main lightweight 'control' process which starts and monitors bipbroker

  - bipbroker

    - Admin process which receives and initiates administration requests

      - *Also referred to as the AdminAgent or DeployManager*

  - dataflowengine

    - Execution group process which runs the message flows

  - biphttplistener

    - Process which provides the http server when using httpInput nodes in your message flows

- Most of the provided administration tools, be they graphical or command line, use the CMP API to communicate with the broker. Previously the CMP API used to communicate with config manager which would then route the requests to the appropriate broker. Now all requests are made directly to the broker.

- Some of the commands still talk to directly to the broker without using the CMP API.

Where does the configuration data go?

- As of Message Broker v7 all configuration data is stored on the file system
  - No database pre-req
    - So no extra admin overhead when not using databases for message flow applications
- Default location for the data depends on the platform
  - Windows
    - C:\Documents and Settings\All Users\Application Data\IBM\MQSI
  - Unix
    - /var/mqsi
  - z/OS
    - It depends!
      - *Chosen by the user when customizing a z/OS Broker*
      - *Configured by the ++COMPONENTDIRECTORY++ JCL variable*
- This location is generally referred to as the workpath and/or registry

Configuration Data Directories Explained

- \<workpath\>
- /common
  - /errors
    - *Abend/error files are written here – always worth monitoring!*
  - /log
    - *Internal binary trace files*
  - /profiles
    - *Additional user profile scripts*
- /components/\<broker name\>
  - Internal configuration data for a given broker
- /config/\<broker name\>/\<eg name\>
  - Execution group specific command environment scripts
- /registry/\<broker name\>
  - Internal configuration data for a given broker
- /shared-classes
  - Non deployed JAR files
- /XSL
  - Non deployed stylesheets used by the XSLT node

Moving the Configuration Data

- On z/OS you can use a different component directory per broker to store each broker's configuration data in a different location
  - Specify a per broker location for the ++COMPONENTDIRECTORY++ JCL variable when customizing

- On all platforms you can specify mqsicreatebroker options to move certain configuration data
  - -w workPath
    - The directory in which working files for this broker are stored
      - *common/errors*
      - *common/log*
      - *components/<broker name>*
      - *components/shared-classes*
  - -e sharedWorkPath (Unix / Windows only)
    - Primarily used for enabling multi-instance broker support
    - Can also be used to move most broker internal configuration data to another location for HA purposes
      - *components/<broker name>*
        - *-e overrides –w setting*
      - *registry/<broker name>*
    - Only 1 file is then stored in the default location pointing at the 2nd location

Broker Runtime Environment

- Broker requires certain environment variables to run
- On distributed platforms mqsiprofile provides the defaults
  - Present in the bin directory of the installation
  - Needs to be applied before running any commands
  - Unix:
    - Broker is started in the same environment as mqsistart is run in
  - Windows:
    - mqsistart kicks off the service definition which creates a new shell and applies the profile
  - If you need to set additional variables then create a new profile in the <workpath>/common/profiles directory
  - Any scripts found in the profiles directory are run after the broker profile
  - Any edits to mqsiprofile will be overwritten when a fix pack is applied
- On z/OS the ENVFILE sets up the runtime environment
  - Generated during broker customization in the broker's home directory
  - BIPBPROF member contains the default values
  - Edit BIPBPROF and submit BIPGEN to generate the ENVFILE
  - BIPBPROF can be used for user settings

Per Execution Group Profiles (7.0.0.2)

- Extend or change the environment for a specific execution group

- Distributed
  - Add a script (or scripts) to the appropriate directory
  - Windows:
    - <MQSI_WORKPATH>\config\<my_broker_name>\<my_eg_label>\profiles
  - Linux & Unix
    - <MQSI_WORKPATH>/config/<my_broker_name>/<my_eg_label>/profiles
  - Scripts are run after mqsiprofile and any scripts in the common/profiles directory are run

- z/OS
  - Customize BIPPROF as normal for all execution group parameters
  - Copy and customize BIPEPROF for each appropriate execution group
  - Edit BIPGEN adding an additional step for each new BIPEPROF
  - Submit BIPGEN
  - ENVFILE & ENVFILE.<eg name> generated in broker's home directory

Runtime UserIDs

- Windows:
  - Runs as the userid of the services definition
  - mqsicreate/changebroker –i –a to set/update
  - Can also use the LocalSystem account
- Unix
  - Runs as the userid who issues the mqsistart command
- z/OS
  - Runs as the user defined in the started task definition
  - The user requires an OMVS segment with a home directory
- Windows/Unix: The userid that starts the broker no longer requires mqm authority
  - But it is required to create a broker
- No Database UserID and Password required from v7
  - Use mqsisetdbparms to control default ODBC and JDBC access control
  - Any v6.x defaults are migrated

Runtime Resources UserIDs

- You can control the credentials used by the broker to connect to external resources by using mqsisetdbparms

- Associate credentials, normally username/password with a resource name

- Resource name is referenced from a flow or configurable service definition

- mqsisetdbparms used to create, alter & delete credentials

- Credetials can be set for nearly all external resources which broker can connect to
  - CICS, ODBC/JDBC databases, Email POP/IMAP/SMTP, FTP, IMS, JMS/JNDI, Kerberos Key Distribution Center (KDC), SFM, keystores, EIS providers, WSRR

- After updating any userid/password definitions you must restart the relevant execution group/broker to pick up the changes

Connecting to a broker

- You need to connect to the broker's Queue Manager to perform administration actions

  - MQ Bindings connect to local brokers
  - MQ Client connect to local or remote brokers

  - Connecting to a local broker
    - Just the broker name required
      - As it's a local broker we can look everything else up
    - Graphical tools will automatically show local brokers

  - Connecting to a remote broker
    - Hostname, port and broker name required
    - More advanced options available
      - SVRCONN channel name if not using the default (SYSTEM.BKR.CONFIG)
      - The class name and JAR file location of a Java security exit if one is required for the channel
      - If using SSL on the channel then the following options are also required:
        - *Cipher Suite, Distinguished Names, CRL Name List, Key Store, and Trust Store*

Command-line

- Message Broker ships with commands for performing configuration and administration actions

- These complement and extend our graphical administration options

- On distributed platforms you need to apply the mqsiprofile to be able to run the commands

- On z/OS the commands are available as jobs, console commands, or both

  - During broker customization you should copy the sample jobs from the SBIPSAMP/SBIPPROC libraries to the broker's component dataset

  - The commands run by jobs are run as the user submitting the job

    - Unless a USER=<user> statement is added to the JCL

  - Console commands are run by the broker userid and are run inside the main broker started task address space

Command-line

- Two types of commands:
  - Java based ones which used to be the commands that talked to the Config Manager
    - These use the CMP API
    - Can work with local and remote brokers
    - Eg: mqsideploy, mqsicreateexecutiongroup, mqsistartmsgflows

```
BIP1121I: Creates an execution group.

Syntax:
mqsicreateexecutiongroup  brokerSpec  -e egName [-w timeoutSecs] [-v traceFileName]

Command options:
'brokerSpec' is one of:
   (a)  'brokerName' : Name of a locally defined broker
   (b)  '-n brokerFileName' : File containing remote broker connection parameters (*.broker)
   (c)  '-i ipAddress -p port -q qMgr' : hostname, port and queue manager of a remote broker
   '-e egName' name of the new execution group
   '-w timeoutSecs' maximum number of seconds to wait for the execution group to be created
   '-v traceFileName' send verbose internal trace to the specified file.
```

  - Older commands which just work with local brokers
    - Eg: mqsistart, mqsistop, mqsichangeproperties

Command-line examples

- **mqsilist**
  - Displays a list of local brokers
  - Displays detailed information about brokers and their deployed resources via **–d** option
    - This works with local remote brokers

```
BIP1288I: Message flow 'simpleflow' on execution group 'ello' is running.
Additional thread instances: '0'
Deployed: '24/07/09 16:37' in Bar file 'C:\My Documents\BAR Files\test.bar'
Last edited: '08/08/07 17:42'
User-defined property names:
Keywords:
    Author = 'Matt'
    Information = 'This flow simply removes messages from SYSTEM.DEFAULT.LOCAL.QUEUE'
    Usage = 'This usage is buried inside the CMF' VERSION = 'v1.1'
```

- **mqsistart / mqsistop**
  - Use to stop and start brokers

- **mqsireload**
  - Use to restart a restart a broker, or a single execution group to pickup configuration changes

Message Broker Toolkit

Message Broker Toolkit - Notes

- Use the Brokers view to create and work with brokers in the WebSphere® Message Broker Toolkit.

- The brokers view offers a limited set of administration actions and is primarily aimed at developers who want to deploy and test their message flows.

- By default, the Brokers view is displayed at the bottom of the Broker Application Development perspective in the WebSphere Message Broker Toolkit. If the Brokers view is not displayed, you can show it by clicking Window > Show View > Other > Broker Runtime > Brokers.

- Brokers that are created on the local system are automatically displayed in the Brokers view. You can add remote brokers to the Brokers view. When you open or switch to the Brokers view, the WebSphere Message Broker Toolkit attempts to connect to brokers on the local system, and any remote brokers that have been defined. Warnings and errors might be displayed if the WebSphere Message Broker Toolkit cannot connect to brokers, for example, if the broker is stopped, or the queue manager listener is not running.

- Right-click the Brokers folder in the Brokers view to display the following options:
  - New Local Broker
  - Connect to a Remote Broker
  - If you specify a keystore or truststore in the remote connection information, you are prompted to enter a password for the keystore or truststore when you connect to the remote broker.
  - Connect to a Remote Broker Using *.broker File
  - Refresh

Message Broker Explorer (MBX)

- New advanced broker management option designed for administrators
- Plug-in to MQ Explorer

- Extra features
  - Create/Manage Configurable Services
  - Performance Views
  - Group brokers using broker sets
  - Offload WS-Security onto Datapower
  - Administration Log
  - Administration Queue
  - Security & Policy Set editors
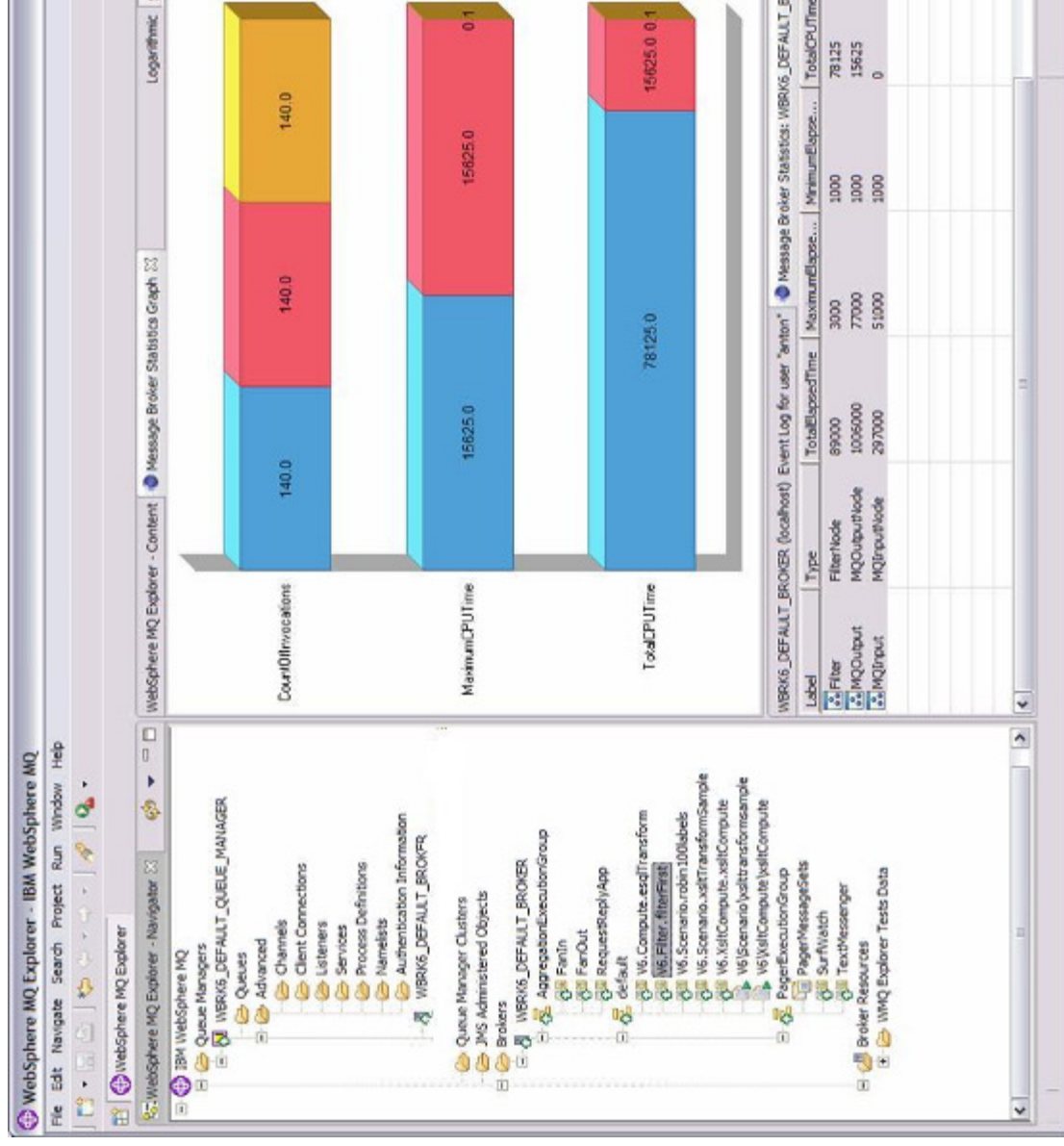
Administration Log

- Administration Log in MBX shows all recent activity on the broker
  - Deployments, deletions, starts, stops, property changes etc.
  - Save/Clear log option
  - Double-click for more information
  - Log is not persisted over broker restarts

Administration Queue

- The broker can now process administration requests concurrently
- Use the MBX Administration Queue to view all outstanding administration requests
- Administrator can select individual pending items and cancel them if necessary

Brokers

MB7BROKER [running, connected]
  default [running]
    simpleflow [running]
  foo [running]
  Configurable Services
  Administration Queue [4]
  test [stopped]

MQ Explorer - Content

Administration Queue

Administration Queue QuickView:

Order	Status	Username	Operation Type	Object Name	Object Type	Creation Time	Elapsed ...	Identifier
1	submitted	Matt	stop	foo	Execution Group	15-Sep-2009 11:55:50	0	DM9715bd43-
2	pending	Matt	deletechild	default	Execution Group	15-Sep-2009 11:57:46	0	DMe66acfe-c
3	pending	Matt	deploy	foo	Execution Group	15-Sep-2009 11:58:57	3	DM1942c375-
4	pending	Matt	deploy	foo	Execution Group	15-Sep-2009 11:59:55	1	DM1a81e52f-(

Resource Statistics

- Find out the current resource usage of a broker or execution group

  - **CICS** – successful requests, failures, security failures…
  - **CORBA** – Invocations, Success, Failures
  - **FTE** – Inbound/Outbound transfers, bytes sent/received…
  - **JDBC** – Requests, Cached requests, Providers…
  - **JVM** – Memory used, thread count, heap statistics…
  - **ODBC** – Connections, Closures, Errors, Successes
  - **SOAPInput** – Inbound messages, Replies, Failures, Policy Sets
  - **Security** – Operations, Success, Failures, Cache usage….
  - **Sockets** – Total sockets, message sizes, Kb sent/received
  - **Parsers** – Memory usage; message elements created/deleted; parser count

- More resource types being added in the future
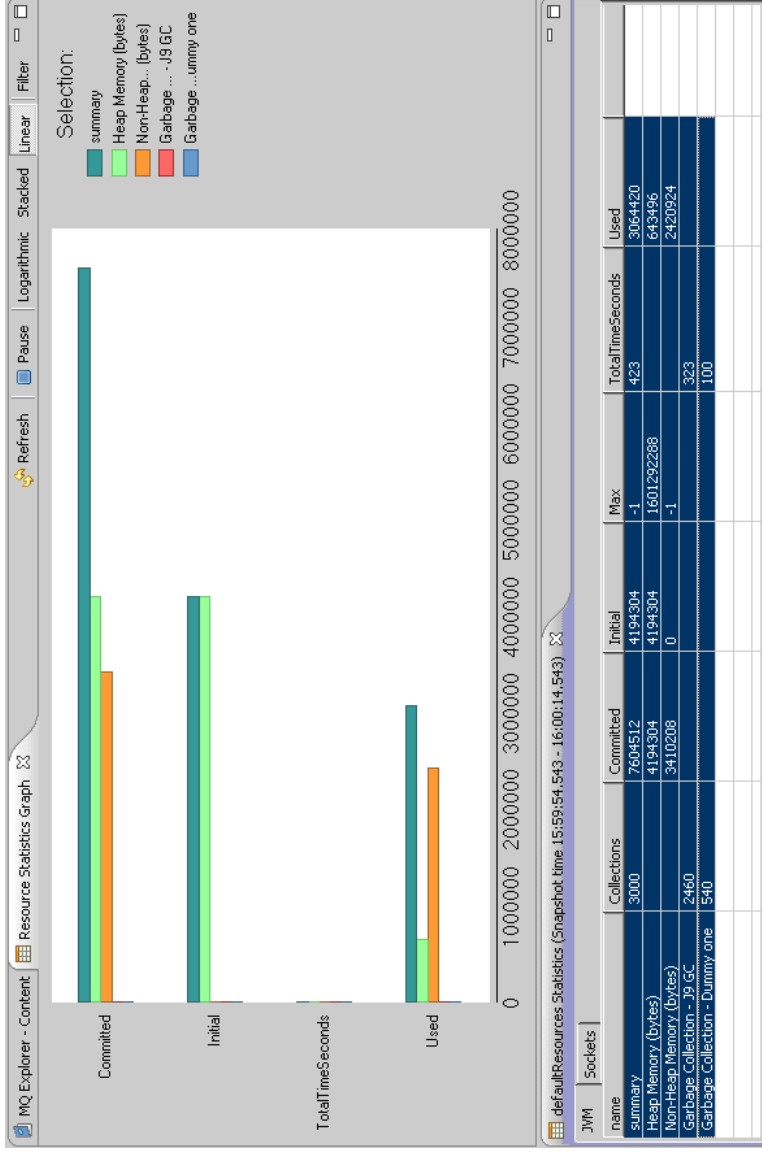
Resource Statistics XML

- Based on existing accounting and statistics framework
  - Sample XML published to
 $SYS/Broker/<broker>/ResourceStatistics/<eg> :

```
<ResourceStatistics brokerLabel="STRESS1" brokerUUID="1e9e4ba1-828a-4f91-ab87-742306e94e5b" executionGroupNam
e="eg.EAS.SOCKET.1" executionGroupUUID="15cccf2f-2401-0000-0080-8e68043b2073" startDate="2009-10-08" startTim
e="10:29:26.198" endDate="2009-10-08" endTime="10:29:46.270">
  <ResourceType name="Sockets">
    <resourceIdentifier name="summary" TotalMessages="854" MinimumMessagesPerMinute="1179" MaximumMessagesPer
Minute="1383" AverageMessagesPerMinute="5328" TotalBytesSent="1066547" MinimumBytesSentPerSecond="11571" Maxi
mumBytesSentPerSecond="41756" AverageBytesSentPerSecond="42334" TotalBytesReceived="282958" MinimumBytesRecei
vedPerSecond="2692" MaximumBytesReceivedPerSecond="11455" AverageBytesReceivedPerSecond="3264" MinimumBytesSe
ntPerMessage="835627" MaximumBytesSentPerMessage="835627" AverageBytesSentPerMessage="2506881" MinimumBytesRe
ceivedPerMessage="54338" MaximumBytesReceivedPerMessage="54338" AverageBytesReceivedPerMessage="163014"/>
    <resourceIdentifier name="localhost.7800" TotalMessages="461" MinimumMessagesPerMinute="1383" MaximumMess
agesPerMinute="1383" AverageMessagesPerMinute="1383" TotalBytesSent="231422" MinimumBytesSentPerSecond="11571
" MaximumBytesSentPerSecond="11571" AverageBytesSentPerSecond="11571" TotalBytesReceived="229117" MinimumByte
sReceivedPerSecond="11455" MaximumBytesReceivedPerSecond="11455" AverageBytesReceivedPerSecond="11455" Minima
mBytesSentPerMessage="502" MaximumBytesSentPerMessage="502" AverageBytesSentPerMessage="502" MinimumBytesRece
ivedPerMessage="497" MaximumBytesReceivedPerMessage="497" AverageBytesReceivedPerMessage="497"/>
    <resourceIdentifier name="localhost.7900" TotalMessages="393" MinimumMessagesPerMinute="1179" MaximumMess
agesPerMinute="1179" AverageMessagesPerMinute="41756" TotalBytesSent="835125" MinimumBytesSentPerSecond="41756
" MaximumBytesSentPerSecond="2692" AverageBytesSentPerSecond="41756" TotalBytesReceived="53841" MinimumBytes
ReceivedPerSecond="2692" MaximumBytesReceivedPerSecond="2692" AverageBytesReceivedPerSecond="2692" MinimumByt
esSentPerMessage="2125" MaximumBytesSentPerMessage="2125" AverageBytesSentPerMessage="2125" MinimumBytesRecei
vedPerMessage="137" MaximumBytesReceivedPerMessage="137" AverageBytesReceivedPerMessage="137"/>
  </ResourceType>
</ResourceStatistics>
```

Messsage Broker Administration API (CMP API)

- Programming interface that your applications can use to control brokers and their resources through a remote interface.

- Set of Java classes packaged as a single JAR file

- Full javadoc for the API is available in the manuals and in the broker runtime

- The API can be used to perform the following tasks
  - Deploy BAR files
  - Change the broker configuration properties
  - Create, modify, and delete execution groups
  - Inquire and set the status of the broker and its associated esources, and to be informed if status changes
  - Execution groups
  - Deployed message flows
  - Deployed files used by the message flows (for example, JAR files)
  - View the Administration log
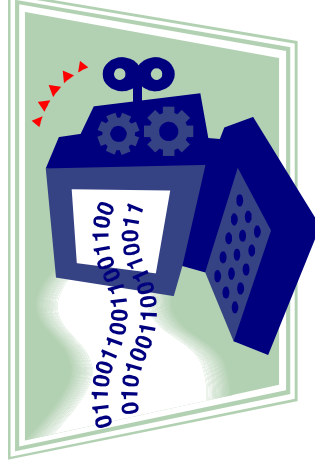
V7 CMP API Changes

- The CMP remains the strategic interface for programmatic administration
  - Now connects directly to the broker!

- Lots more information provided
  - Local broker information
  - Node API
  - Configurable Services
  - Accounting and Statistics Information

- Support for v6.x applications
  - Old package names, class names and method signatures unaffected (although many deprecated methods)
  - Applications will compile without change
  - Applications that do operations that are relevant in v7 should continue to work

- Future
  - Move remaining mqsi commands to use CMP
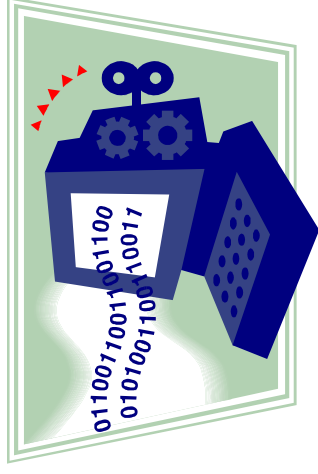
CMPAPI Example

- This simple example connects to a broker on the local machine and deploys a bar file and displays the result

```
import com.ibm.broker.config.proxy.*;
public class DeployBAR {
public static void main(String[] args) {
BrokerConnectionParameters bcp =
        new MQBrokerConnectionParameters("localhost", 2414, "MB7QMGR");

try {

BrokerProxy b = BrokerProxy.getInstance(bcp);
ExecutionGroupProxy eg = b.getExecutionGroupByName("default");
DeployResult dr = eg.deploy("MyBAR.bar", true, 30000);
System.out.println("Result = "+dr.getCompletionCode());

} catch (Exception e) {
e.printStackTrace();
}
}
}
```

CMP API Exerciser

Administrative Security

- Simplified administrative security in V7 allows 3 levels of authorisation for administrative actions:
  - Reading
  - Writing
  - Executing (i.e. starting and stopping)

- On two object types:
  - Broker
  - Execution Group

- Administrative Security is not enabled by default

- Access controlled using MQ queues on the broker's queue manager

- Guidance provided for migration from CM ACLs
  - Though there is not a one-to-one mapping

Security Queues

SYSTEM.BROKER.AUTH
SYSTEM.BROKER.AUTH.<egname>

+inq = Read
+put = Write
+set = Execute

Required Task Authorizations

Tasks	Queue Names SYSTEM.BROKER.AUTH	SYSTEM.BROKER.AUTH.EG
Set broker properties	R+W	
View broker properties	R	
Create or delete configurable services	R+W	
Set configurable services properties	R+W	
View configurable services properties	R	
Create or delete execution groups	R+W	
Rename execution groups	R+W	
List execution groups	R	
Start or stop execution groups	R X^1	X^1
Set execution group properties	R	W
View execution group properties	R	R
Start or stop resource statistics collection	R	X
Report resource statistics	R	R
Deploy	R	W
List message flows and other deployed objects	R	R
Start or stop message flows	R	X
Delete resources from an execution group	R	W

Note: X^1 Execute access is required on the broker or on an individual execution group

http://publib.boulder.ibm.com/infocenter/wmbhelp/v7r0m0/topic/com.ibm.etools.mft.doc/bp43530_.htm

Required Command Authorizations

Command	Queue Names	
	SYSTEM.BROKER.AUTH	SYSTEM.BROKER.AUTH.EG
mqsichangeresourcestats	R	X
mqsicreateexecutiongroup	R+W	
mqsideleteexecutiongroup	R+W	
mqsideploy	R	W
mqsilist	R	R[1]
mqsimode	R (to display) R+W (to change)	
mqsireloadsecurity	R	W
mqsireportresourcestats	R	R
mqsistartmsgflow	R	X
mqsistopmsgflow	R	X

Note: R[1] You require read access on any execution groups for which you wish to display information
http://publib.boulder.ibm.com/infocenter/wmbhelp/v7r0m0/topic/com.ibm.etools.mft.doc/bp43540_.htm

Broker backup/restore

- As Message Broker v7 no longer uses a system database it is a lot easier to perform backup and restore
- Additionally, the Windows registry is no longer used to hold configuration information
- Only file system artifacts now been to be backed up
- New mqsibackupbroker and mqsirestorebroker commands to backup and restore (for DR)
  - Backup and restore a single broker
  - Backup an active broker as long as it's not processing configuration changes
  - Use the backup file that is created to restore a broker in an identical operating environment
    - The operating system must be at the same level, and the broker and queue manager names must be identical.

  mqsibackupbroker MB7BROKER -d c:\MQSI\BACKUP

- On z/OS edit and submit the BIPBUBK job

MQ HA Overview – Initial State
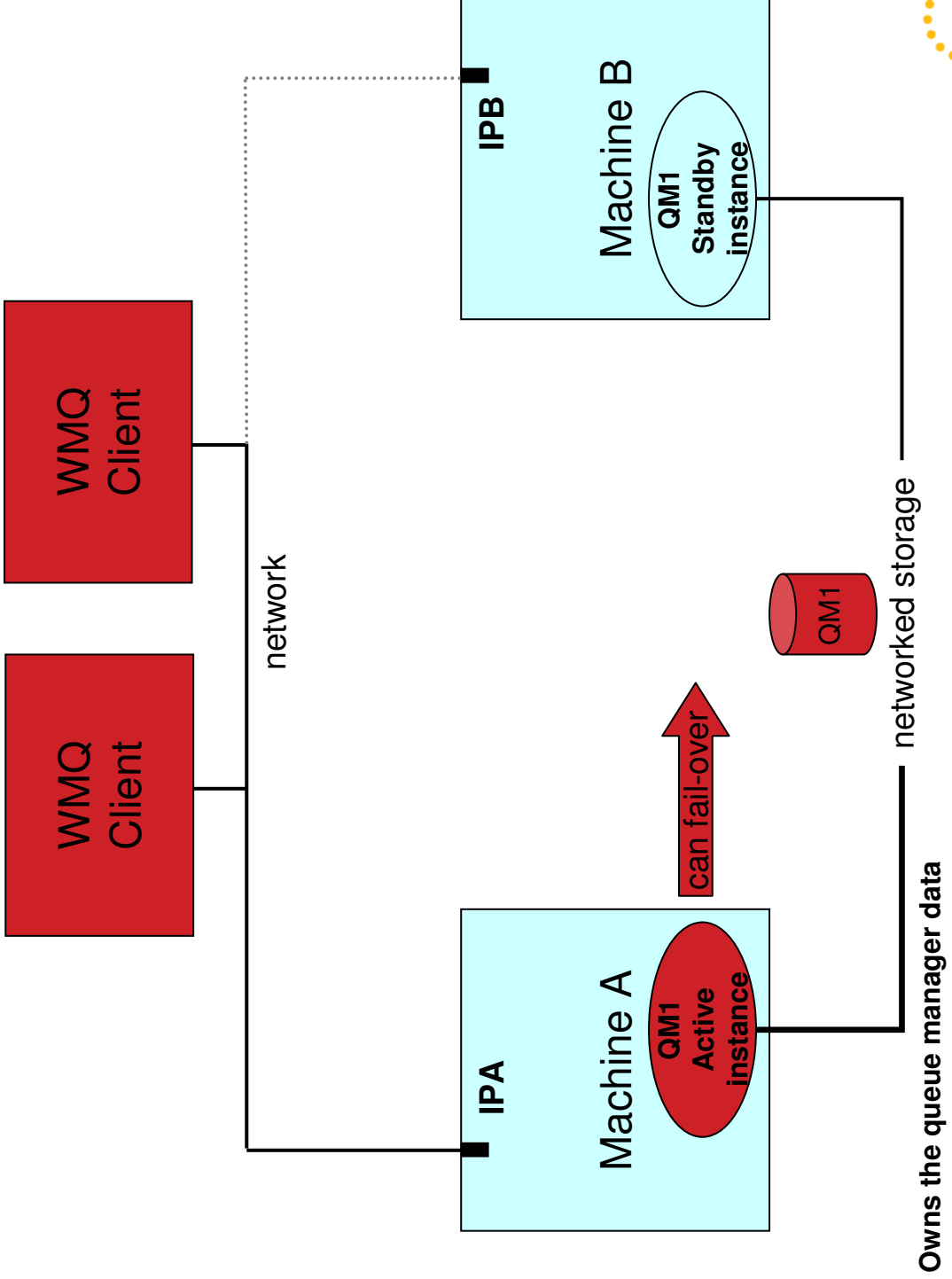
Message Broker - Standalone

Machine A

Broker1 Active instance

QM1 Active instance

can fail-over

Machine B

Broker1 Standby instance

QM1 Standby instance

Broker1

QM1

networked storage

networked storage

Owns the queue manager data
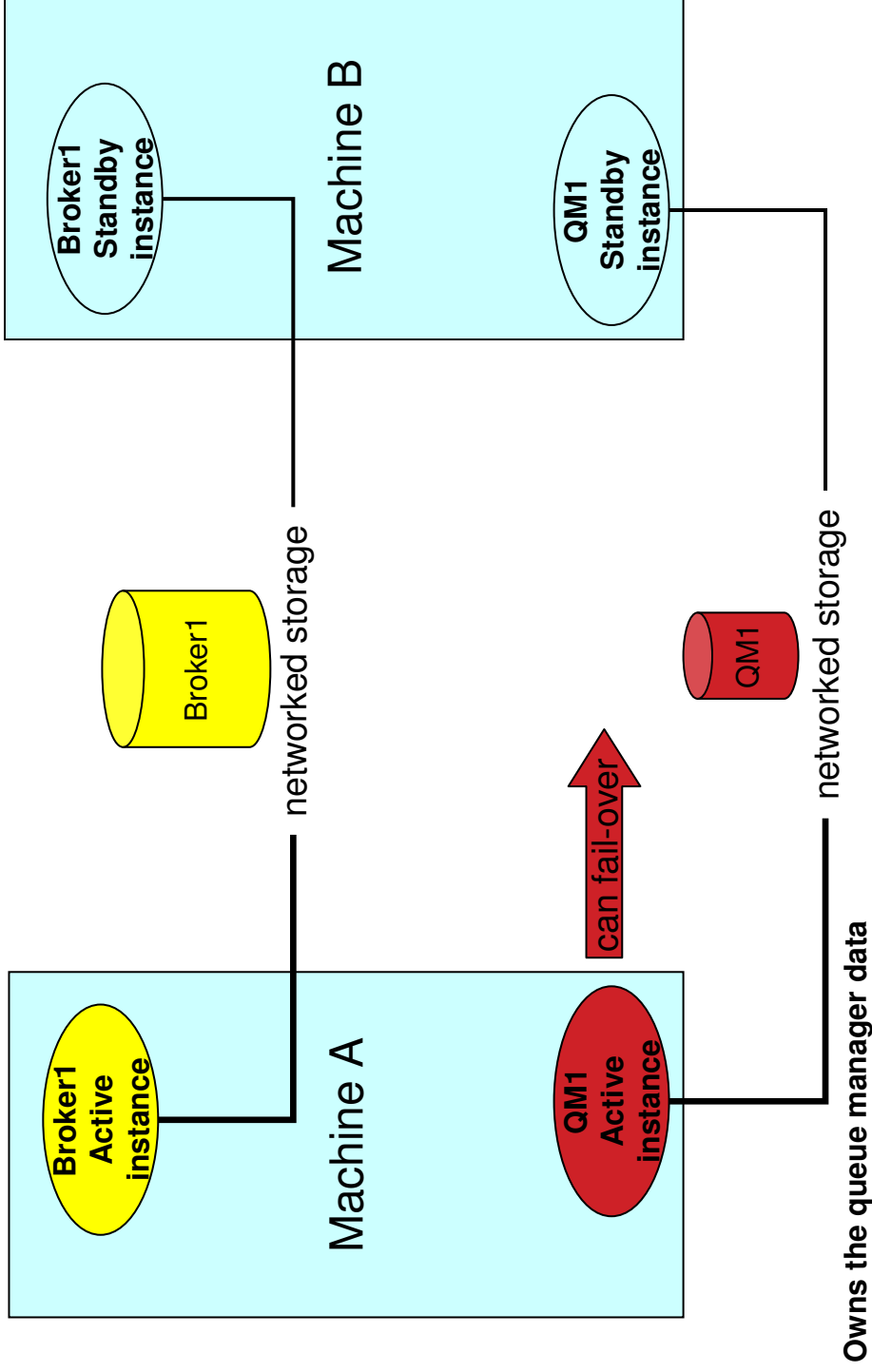
Message Broker – As a Service

Machine B

QM1
Standby
instance

Machine A

QM1
Active
instance

Broker1
Active
instance

Broker1

QM1

networked storage

networked storage

can fail-over

Owns the queue manager data
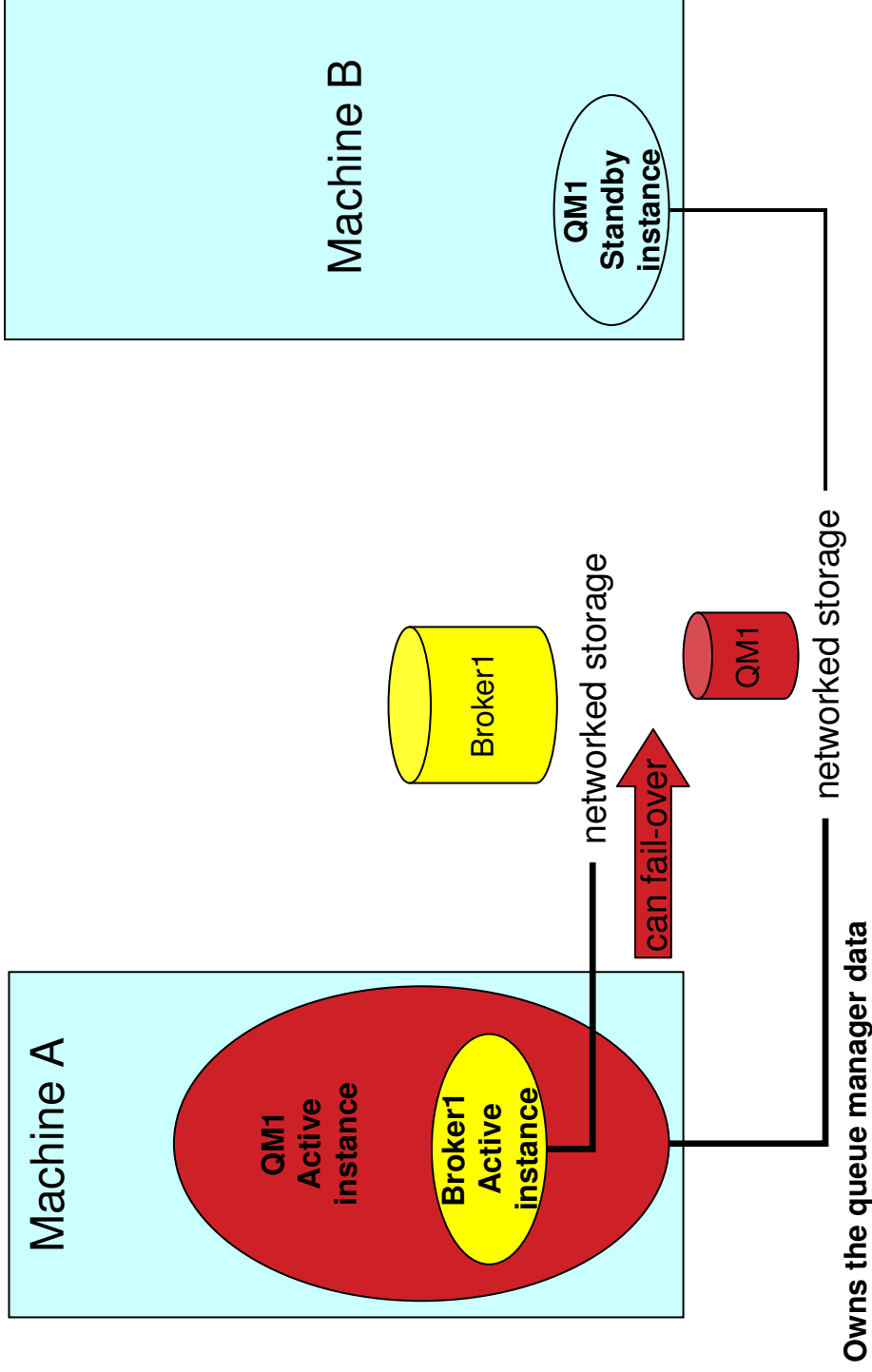
Migrating to V7

- Message Broker V7 supports coexistence
  - Install v7 alongside your previous version
  - However, must use V7 tools for v7 brokers, v6.x tools for v6.x brokers/ConfigMgrs

- Before migration
  - Move to WebSphere MQ V7.0.1
  - If you want to use Pub/Sub, run *migmbbrk*

- Migration
  - Direct migration (*mqsimigratecomponents*) supported from V6 and V6.1
  - Rollback option available (any post-migration configuration changes not reflected)
  - Load up existing artefacts in the V7 toolkit

- After migration
  - Graphical tools automatically show local brokers
  - If you wish to manage remote brokers, connect to them (IP/Port/QMgr)
  - Remove CM and DB if no longer required

Configurable Services

- Allows separation of flow design from the details about external services
  - eg: SMTP server or a JMS provider
- Flow developer configures nodes with the configurable service names
- Broker administrator creates and configures the configurable service with appropriate values
- Restart the execution group for changes to take effect
- Configure using Message Broker Explorer or using the commands
  - mqsicreateconfigurableservice – create a new configurable service
    mqsicreateconfigurableservice <brokerName> -c <cs name> -o <cs name>
    mqsicreateconfigurableservice <brokerName> -c <cs name> -o <cs name> -n <property>,<property2> -v <value>,value2>
  - mqsichangeproperties – change an existing configurable service
    mqsichangeproperties <brokerName> -c <cs type> -o <cs name> -n <property> -v <value>
  - mqsireportproperties – report available configurable services and their attributes
    mqsireportproperties <brokerName> -c <cs type> -o <cs name> -r
  - mqsideleteconfigurableservice – delete a configurable service
    mqsideleteconfigurableservice <brokerName> -c <cs name> -o <cs name>
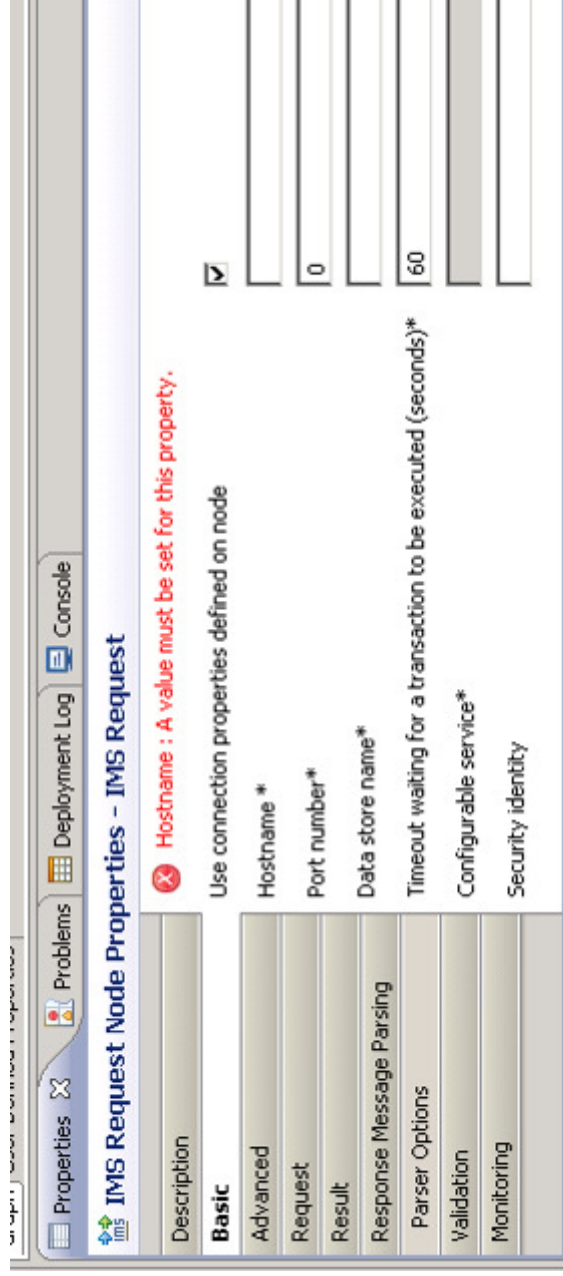
Configurable Services

- Extensive list available

  - Aggregation
  - CICSConnection
  - Collector
  - CORBA
  - EmailServer
  - EISProviders
  - FtpServer
  - IMSConnect
  - JavaClassLoader

  - JDBCProviders
  - JDEdwardsConnection
  - JMSProviders
  - MonitoringProfiles
  - PeopleSoftConnection
  - PolicySets
  - PolicySet Bindings
  - Resequence
  - SAPConnection

  - SecurityProfiles
  - Service Registries
  - SiebelConnection
  - SMTP
  - TCPIPClient
  - TCPIPServer
  - Timer
  - UserDefined

Properties | Problems | Deployment Log | Console

IMS Request Node Properties – IMS Request

Description	ⓧ Hostname : A value must be set for this property.
Basic	Use connection properties defined on node ☑
Advanced	Hostname *
Request	Port number *
Result	Data store name *
Response Message Parsing	Timeout waiting for a transaction to be executed (seconds)* 60
Parser Options	Configurable service*
Validation	Security identity
Monitoring	

Configurable Services - example

mqsicreateconfigurableservice MB7BROKER -c JDBCProviders -o DB2EXTRA -n connectionUrlFormat
-v "jdbc:db2://[serverName]:[portNumber]/[databaseName]:user=[user];password=[password];"

mqsichangeproperties MB7BROKER -c JDBCProviders -o DB2EXTRA -n maxConnectionPoolSize -v 20

mqsireportproperties MB7BROKER -c JDBCProviders -o DB2EXTRA –r

```
JDBCProviders
  DB2EXTRA
    connectionUrlFormat='jdbc:db2://[serverName]:[portNumber]/[databaseName]:user=[user];password=[password];'
    connectionUrlFormatAttr1=''
    connectionUrlFormatAttr2=''
    connectionUrlFormatAttr3=''
    connectionUrlFormatAttr4=''
    connectionUrlFormatAttr5=''
    databaseName='default_Database_Name'
    databaseType='default_Database_Type'
    databaseVersion='default_Database_Version'
    description='default_Description'
    environmentParms='default_none'
    jarsURL='default_Path'
    maxConnectionPoolSize='20'
    portNumber='default_Port_Number'
    securityIdentity='default_User@default_Server'
    serverName='default_Database_Server_Name'
    type4DatasourceClassName='default_Type_Four_Datasource_Class_Name'
    type4DriverClassName='default_Type_Four_Driver_Class_Name'
```

BIP8071I: Successful command completion.

mqsideleteconfigurableservice MB7BROKER -c JDBCProviders -o DB2EXTRA

Configurable Services

Configurable Service

Configurable Service
Create a new Configurable Service and set its attributes

*Name
*Type   Aggregation
Template   Template

Key	Value
queuePrefix	
timeoutSeconds	

Add Property   Delete Property

mqsicreateconfigurableservice MB7BROKER -c Aggregation -o -n "\"queuePrefix\", \"timeoutSeconds\"" -v "\"\", \""

Finish   Cancel

MQ Explorer - Navigator

MB7BROKER
default
loopFlow
migrate
SecurityPEPNodeSampleExecutionGroup
SOAPNodesSample
twineball
UNITTest
WSHOST
Configurable Services
Aggregation/agg1
Aggregation/Template
CICSConnection/CICSConnectionTemplate
CORBA/CORBATemplate
Collector/Template
EISProviders/JDEdwards
EISProviders/PeopleSoft
EISProviders/SAP
EISProviders/Siebel
EISProvid...
EmailServ...
JDBCProv...

MQ Explorer - Content

Configurable Service CICSConnectionTemplate

Properties QuickView:

Name	CICSConnectionTemplate
Type	CICSConnection
cicsServer	
clientApplid	
clientQualifier	
connectionTimeoutSecs	30
gatewayURL	local:
requestTimeoutSecs	120
securityIdentity	

Local Error Logs

- Key information point for an administrator to monitor
- Message Broker components use the local error log to record information about major activities
- Actual local error logs vary by platform
  - Windows – Windows Event Log (Application View)
  - Unix/Linux – syslog
  - z/OS – JOBLOG & system console log
- When an error occurs, check the local error log first
- Often requested by support
  - Windows
    - The event log fills up so check the size is sufficient or that circular logging is enabled
  - Unix/Linux
    - Make sure you configure the syslog daemon

stdout/stderr

- Useful place to look for errors / debugging
  - Always worth checking for exceptions if problems are occurring
- Each major component redirects its stdout/stderr streams to files
  - Windows
    - *Admin Agent (7.0.0.2)*
      C:\Documents and Settings\All Users\Application Data\IBM\MQSI\components\<brkName>\console.txt
    - *Execution group*
      C:\Documents and Settings\All Users\Application Data\IBM\MQSI\components\<brkName>\<egUUID>\console.txt
  - Linux/Unix
    - *Admin Agent (7.0.0.2)*
      /var/mqsi/components/<brkName>/stdout & stderr
    - *Execution group*
      /var/mqsi/components/<brkName>/<egUUID>/stdout & stderr
  - z/OS
    - *STDOUT / STDERR DD cards in the joblog for both the main broker address space and for any execution groups*
- Can be useful for flow developers who use Java and code system.out.println statements for debugging

Coredump

- In the unlikely event that Message Broker encounters a problem that results in a coredump you need to be aware of where to look for dumps

  - Windows
    - BIP2111 error message (message broker internal error).
    - The error message contains the path to the MiniDump file in your errors directory

  - Linux/UNIX
    - BIP2060 error message (execution group terminated unexpectedly)
    - Look in the directory where the broker was started, or in the service user ID's home directory, to find the core dump file
    - Check your ulimits
      - We recommend an unlimited hard & soft limit for corefile size
      - Ensure you have enough disk space

SVC dump (z/OS)

- Message Broker on z/OS should always produce an SVC dump

- Dump dataset is written based on the system defined setup
  - Use the "display dump" command to display the naming options
  - BIP2060 error message (execution group ended unexpectedly) from the main Broker Address Space.
  - Message should be accompanied by one of the following messages and dump
    - *IEF450I message in the syslog, or component's joblog, showing an abend code followed by a reason code, for example:*

      `IEF450I MQ83BRK DEFAULT - ABEND=S2C1 U0000 REASON=000000C4`
    - *Look in the system's dump dataset hlq for the dump dataset, or search the syslog for the appropriate IEA611I message to find out the dump dataset name.*

- In extreme cases you may see a coredump instead
  - In these cases you will see an IEA993I message in the syslog
  - Look in the started task user's directory for the coredump.pid file, as specified in the syslog:

    `IEA993I SYSMDUMP TAKEN TO coredump.00500319`

- If a dump is not produced then look for a reason why in the JOBLOG and system console log
  - Check both as errors are not always repeated
  - A dump might have been suppressed by DAE

Summary

- V7 is a significant release
- Many improvements for the administrator
  - No Configuration Manager
  - Message Broker Explorer
  - Administration Queue
  - Administration Log
  - Resource Statistics
  - No system database
  - Multi-instance Brokers (HA)
  - Publish/subscribe converged with WMQ
- Expect lots more to come!

- Remember existing administration options

- If you encounter problems always remember to look in the logs

The rest of the week ......

	Monday	Tuesday	Wednesday	Thursday	Friday
08:00			More than a buzzword: Extending the reach of your MQ messaging with Web 2.0	Batch, local, remote, and traditional MVS - file processing in Message Broker	Lyn's Story Time - Avoiding the MQ Problems Others have Hit
09:30		WebSphere MQ 101: Introduction to the world's leading messaging provider	The Do's and Don'ts of Queue Manager Performance	So, what else can I do? - MQ API beyond the basics	MQ Project Planning Session
11:00		MQ Publish/Subscribe	The Do's and Don'ts of Message Broker Performance	Diagnosing problems for Message Broker	What's new for the MQ Family and Message Broker
12:15	MQ Freebies! Top 5 SupportPacs	The doctor is in. Hands-on lab and lots of help with the MQ family		Using the WMQ V7 Verbs in CICS Programs	
01:30	Diagnosing problems for MQ	WebSphere Message Broker 101: The Swiss army knife for application integration	The Dark Side of Monitoring MQ - SMF 115 and 116 record reading and interpretation	Getting your MQ JMS applications running, with or without WAS	
03:00	Keeping your eye on it all - Queue Manager Monitoring & Auditing	The MQ API for dummies - the basics	Under the hood of Message Broker on z/OS - WLM, SMF and more	Message Broker Patterns - Generate applications in an instant	
04:30	Message Broker administration for dummies	All About WebSphere MQ File Transfer Edition	For your eyes only - WebSphere MQ Advanced Message Security	Keeping your MQ service up and running - Queue Manager clustering	
06:00			Free MQ! - MQ Clients and what you can do with them	MQ Q-Box - Open Microphone to ask the experts questions	