

What's New in the WebSphere MQ Family

Session # 8875

Morag Hughson

hughson@uk.ibm.com



What's New in the WebSphere MQ Family - Abstract

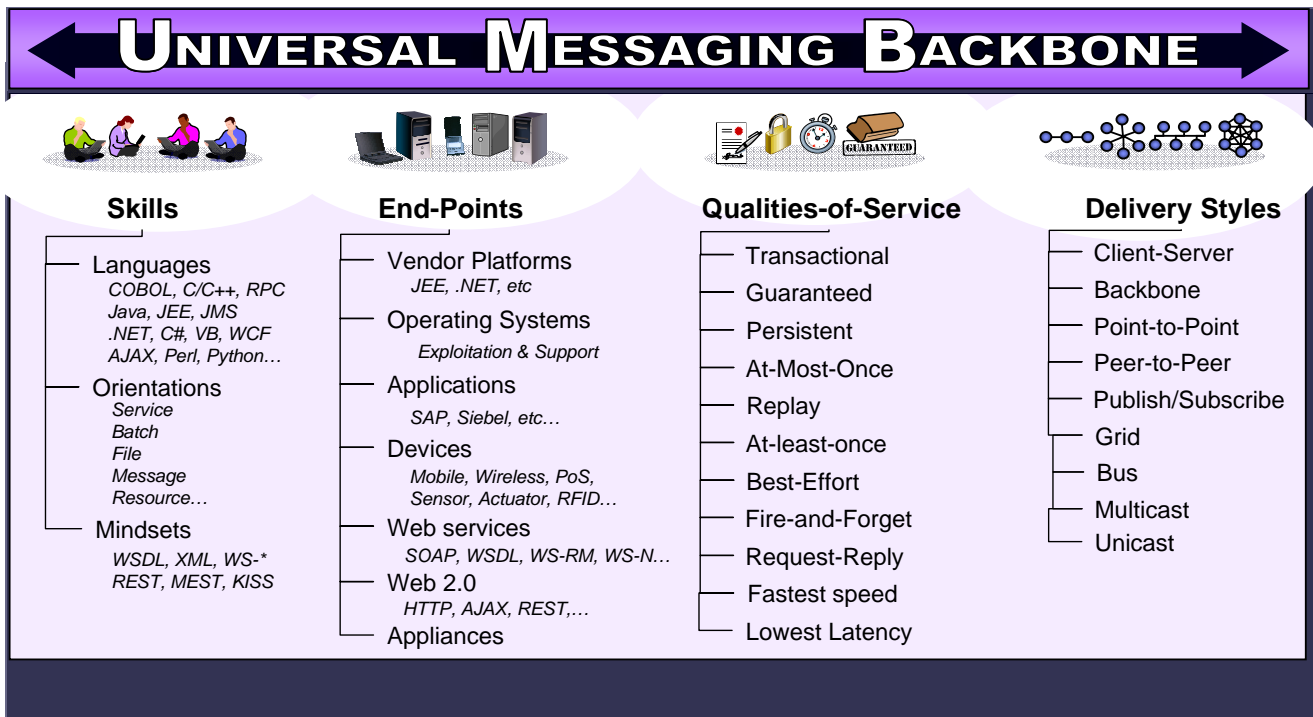
N
O
T
E
S

- WebSphere MQ is the world's leading messaging system used by most of the top fortune 500 companies. Millions of messages and trillions of dollars flow through MQ every day. In spite of this success the product continues to evolve and improve. Constant evolution and development is the bedrock of sustained success and WebSphere MQ is no exception to this process.
- Why not take this opportunity to come along and listen in about the most recent changes and features in the WebSphere MQ product family.
- This session will recap on the themes of MQ 7.0.1 which bring about additional features to relieve resource constraints, reduce total cost of ownership, and enable better use of existing investments.
- Any recent announcements will of course be highlighted in this session, so why not come along and find out the most up to date information about the WebSphere MQ Product family.

IBM's Vision – Universal Messaging Backbone



- Addressing the full spectrum of speed and availability transport requirements



IBM's Vision – Universal Messaging Backbone



- The vision for MQ is that it provides a range of capabilities, making it suitable to be a transport backbone across all environments. While MQ does not necessarily have all of these capabilities today in the core product, this picture drives the selection of features as we continue to enhance it.

N
O
T
E
S

WebSphere MQ V7.0.1 – Major Themes

- Enhanced availability options on Distributed platforms
- Constraint relief on z/OS
- IBM portfolio exploitation, and extension of reach, for V7 features
- Ongoing performance, consumability and serviceability enhancements
- Keeping pace with industry evolution in areas such as platforms and SSL

WebSphere MQ V7.0.1 Rationale

N
O
T
E
S

- WMQ V7.0.1 is designed to be an vehicle for delivery of function earlier than might have been expected with a traditional release cycle (for example with a V7.1 or a V8).
- Some of the function included in this release is a prerequisite on which other products such as Message Broker will build.

WMQ V7.0.1 Content Summary

New Feature	Benefits	Details
Multi-Instance Queue Managers	Increases availability Does not require specialist skills Can help ease system maintenance	Enables automatic failover to a standby Queue Manager instance in the event of an incident or planned outage
Automatic Client Reconnect	Increases availability Simplifies programming	Provides Client-connected applications with automatic detection of failures and reconnects to alternative Queue Managers
Enhanced Governance	Increases visibility of changes Enables SOA Governance	Emits events whenever configuration changes are made or commands are run Service Definition wizard generates WSDL describing MQ apps
Enhanced SSL Security	Simplifies security certificate management	Supports certificate checks with Online Certificate Status Protocol (OCSP) as well as to Certificate Revocation Lists (CRL)
Enhanced .NET support	Increases ease-of-use for .NET developers	Provides IBM Message Service Client for .NET developers Supports use of WebSphere MQ as custom channel within Windows Communication Foundation
Increased 64-bit z/OS exploitation	Increased use of z/OS system resources Provides constraint relief for virtual storage	Extends use of 64-bit storage by Queue Manager enabling more capacity such as number of open queues
z/OS Log Compression	Increased use of z/OS system resources Increased log performance & bandwidth	Compresses message logs produced by persistent messages
z/OS Group Units of Work	Increased resilience	Enables Units of Work to be owned collectively by Queue Sharing Groups so that any Queue Manager in the group can process two-phase transactions from clients
Publish/Subscribe Interfaces	Additional control of pub/sub behaviour Simplified integration for Message Broker	Exit point to dynamically modify routing and content Tools to migrate pub/sub state from MB to MQ

in Anaheim
2011

Installation and Delivery

- **WMQ V7.0.1 is a modification release on the V7 base**
 - Which means limited scope for new objects/attributes
 - Minimises migration aspects
- **On Distributed platforms, it is available in two ways**
 - A fixpack for upgrade from existing V7 installations (which can be backed out)
 - A replacement V7 installation image
 - Customers ordering V7 will now get V7.0.1
 - Single service stream for V7.0.x.y
- **On z/OS, it is available as a modification level release**
 - Migration supported from V6 and from V7.0.0
 - Customers ordering V7 will now get V7.0.1
 - New ZPARM OPMODE option to control whether new function is available

End of Service and Component Deprecation



- **End of Service has been announced for V6**
 - 30 September 2011 (Distributed and zOS)
- **Components being removed in next full release**
 - File Transfer (Windows/Linux utility) – not to be confused with FTE product
 - API exerciser, Windows performance monitor integration
- **C++ API (the Imq classes) to be stabilised**
 - Cannot extend current code while maintaining application compatibility
 - Many C++ applications use the C MQI today.
- **Next full release will not support HP-UX on PA-RISC hardware**
- **SupportPac MA0F (AMI) will inherit EoS for V6**
- **SupportPac MS0E (Admin Wrapper) will inherit EoS for V6**
- **SupportPac MC91 (HA) withdrawn.**
 - Still downloadable, but not on main SupportPac page



End of Service and Component Deprecation



N
O
T
E
S

- End of Support for V6 (and V7.0.0 on z/OS) is end of Sept 2011.
- <http://www.ibm.com/software/support/lifecycle/> contains details.
- The C++ classes will continue to be supported, and kept in step with any necessary compiler version changes, but the function available through those classes will not be enhanced.



Distributed Platforms: Multi-instance Queue Managers



- **Basic failover support without HA coordinator**
 - Faster takeover: fewer moving parts
 - Cheaper: no specialised software or administration skills needed
 - All non-z/OS platforms
- **Queue manager data is held in networked storage**
 - NAS, NFS, GPFS etc so more than one machine sees the queue manager data
 - Improves storage management options: formal support for these even without failover config
- **Multiple (2) instances of a queue manager on different machines**
 - One is “active” instance; other is “standby” instance
 - Active instance “owns” the queue manager’s files and will accept app connections
 - Standby instance does not “own” the queue manager’s files and apps cannot connect
 - If active instance fails, standby performs queue manager restart and becomes active
- **Instances share data, so it’s the SAME queue manager**



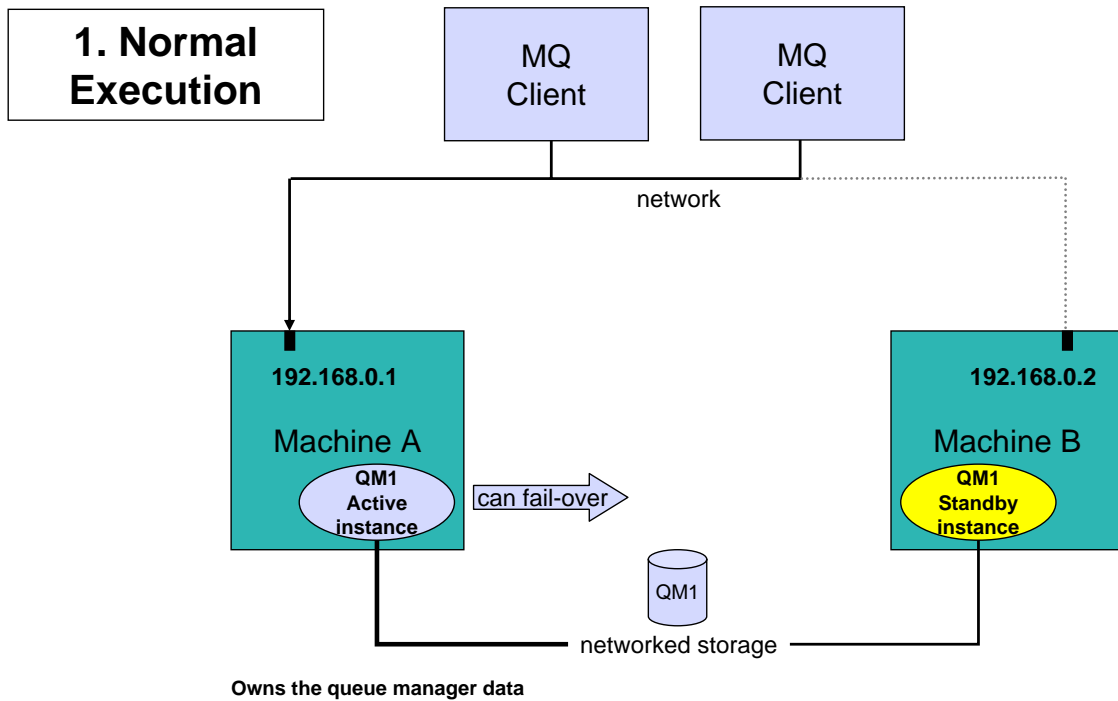
Distributed Platforms: Multi-instance Queue Managers



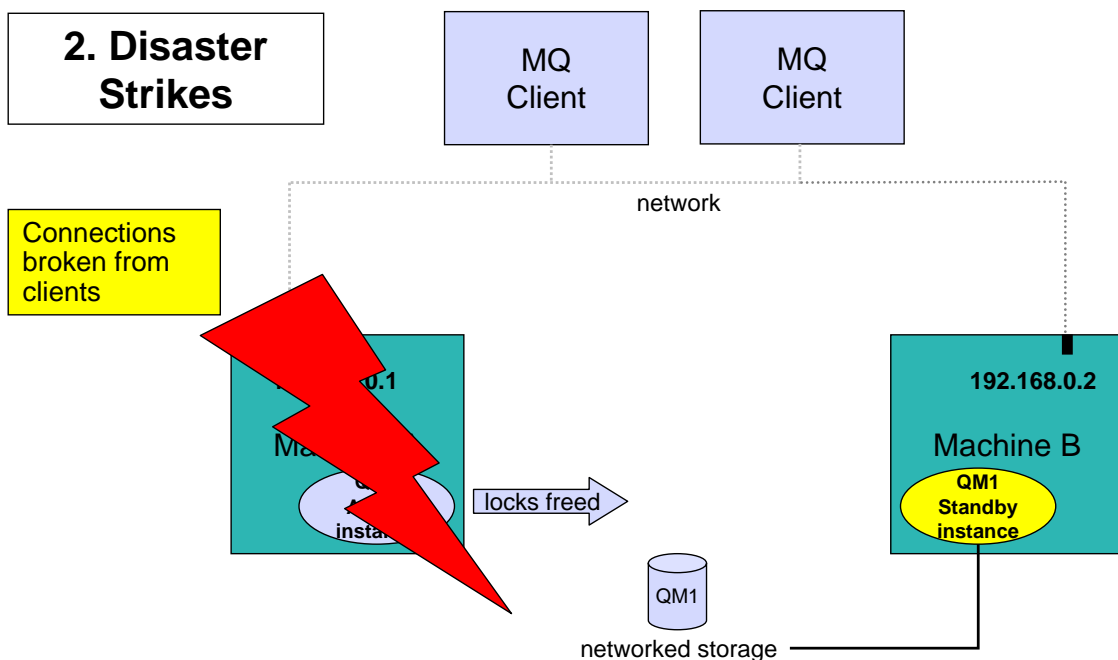
- N**
- “Basic failover”: no interaction with other resources like disks, IP addresses, databases, user applications etc. There is also no sophisticated control over where the queue managers run and move to (eg like a 3-node HACMP setup).
- O**
- Currently no System i: waiting for PTFs for their NFSv4 implementation.
- T**
- Architecturally, this is essentially the same as an existing HACMP/VCS setup, with the data shared between systems. It does not give anything “stronger” in terms of availability – but we do expect the typical takeover time to be significantly less. And it is much simpler to administer.
- E**
- Just as with an HACMP/VCS configuration, the takeover is at heart a restart of the queue manager, so non-persistent messages are discarded, queue manager channels go into retry etc.
- S**
- HACMP/VCS configurations are still a valid choice – but now you have an extra choice. You can continue to use HACMP/VCS solutions using MQ V7.0.1. While we do not (currently) ship, as standard, the start/stop/monitor commands that are currently in MC91, these are simple enough that you should be able to write them anyway. The queue manager is now more sensitive to when its internal processes die, so the MC91 “kill” script that iterated through all process names should not be necessary.



Multi-instance Queue Managers

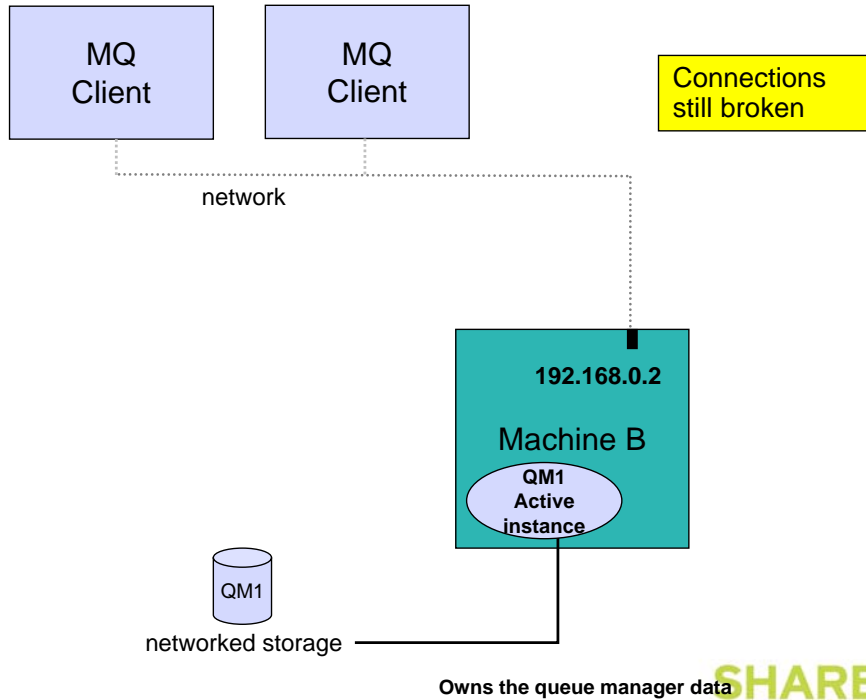


Multi-instance Queue Managers



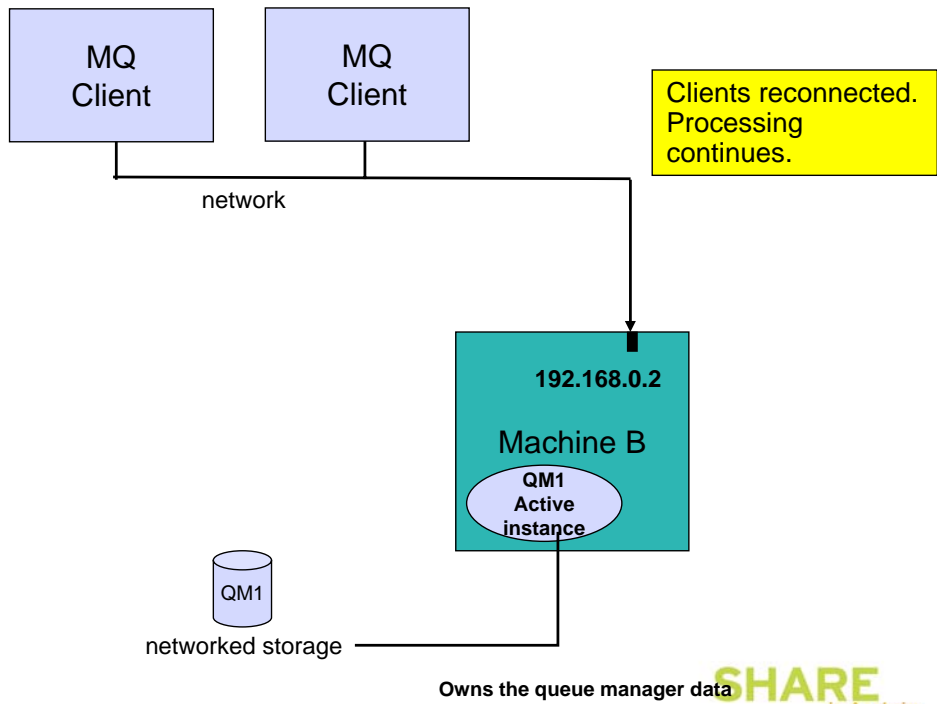
Multi-instance Queue Managers

3. Standby Comes to Life



Multi-instance Queue Managers

4. Recovery Complete



Multi-instance queue managers: Details

- **MQ is NOT becoming an HA coordinator**
 - Generally, if other resources also required, use an HA coordinator such as HACMP
 - Service objects can restart applications with qmgr but limited control
 - Message Broker also integrates with and exploits this MQ function
- **The IP address is not taken over**
 - Channel config needs all possible addresses unless you use external IPAT or intelligent router
 - CONNAME('host1(port1),host2(port2)') syntax extension on all platforms including z/OS
- **Support for networked storage over modern network file system protocols**
 - For example, NFS v4 (not v3)
 - Tool shipped to validate configuration
- **New options for crtmqm/strmqm/endmqm to control operations**
 - Cannot guarantee which instance becomes the primary
- **Removes need for MC91, which will be withdrawn**
 - crtmqm now does equivalent of MC91's hacrtmqm

Multi-instance queue managers: Details

N
O
T
E
S

- QMgr Service Objects have no enforced or implied sequence which make them difficult to use for dependency management in these scenarios. They also force the started applications to run with mqm authority (unless sudo-ish things are done).
- Failover time depends on 2 things:
 - a) How quickly the standby notices the active has gone. This will be near-instantaneous when just the qmgr fails or is explicitly ended. Will probably be longer, based on filesystem timeouts, when the whole box fails.
 - b) The amount of work that has to be done during restart, typically dependent on any long-running transactions that need to be replayed or backed out.
- From the announcement letter:
 - Unix: For multi-instance queue managers, you will need a networked storage device (such as a NAS). The storage must be accessed by a network file system protocol which is Posix-compliant and supports lease-based locking. NFS v4 and IBM GPFS both satisfy this requirement. Earlier versions of NFS do not satisfy this requirement and must not be used with multi-instance queue managers.
 - Windows: A networked storage device accessed by the Common Internet File System (CIFS) protocol used by Microsoft Windows networks is required.
- MC91 is being moved to the “withdrawn” page for SupportPacs so it is still available, but it is no longer as prominent for HA capabilities. The regular commands like crtmqm now do the necessary work previously done by hacrtmqm (although in a different way); there is a new addmqinf command that is the equivalent of halinkmqm. The scripts continue to work with V7.0.1 both for existing/upgraded qmgrs, and for newly created ones. But they are not recommended.

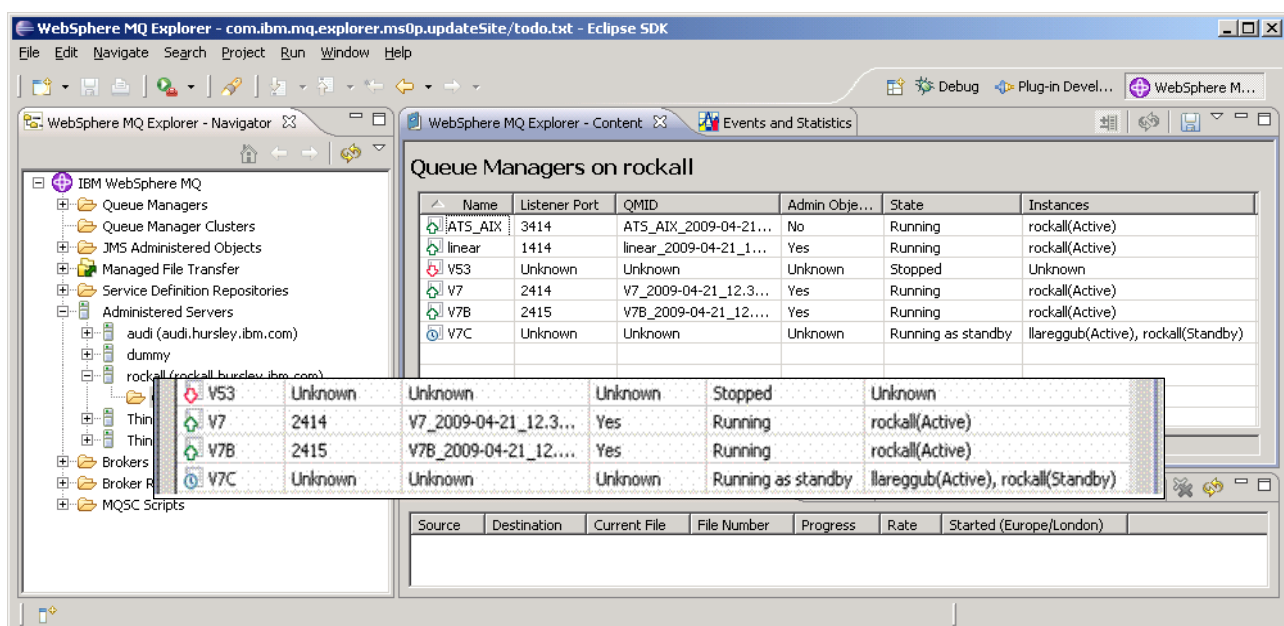
Multi-instance queue managers: How it looks

- Enhanced dspmq
- New option for dspmq to output English-only text
 - Useful for programmable parsing

```
$ hostname
rockall
$ dspmq -x
QMNAME(V7)          STATUS(Running)
  INSTANCE(rockall)  MODE(Active)
QMNAME(V7B)         STATUS(Running)
  INSTANCE(rockall)  MODE(Active)
QMNAME(V7C)         STATUS(Running as standby)
  INSTANCE(llareggub) MODE(Active)
  INSTANCE(rockall)  MODE(Standby)
```

Multi-instance queue managers: How it looks

- As a graphical example, SupportPac MS0P V7.0.1

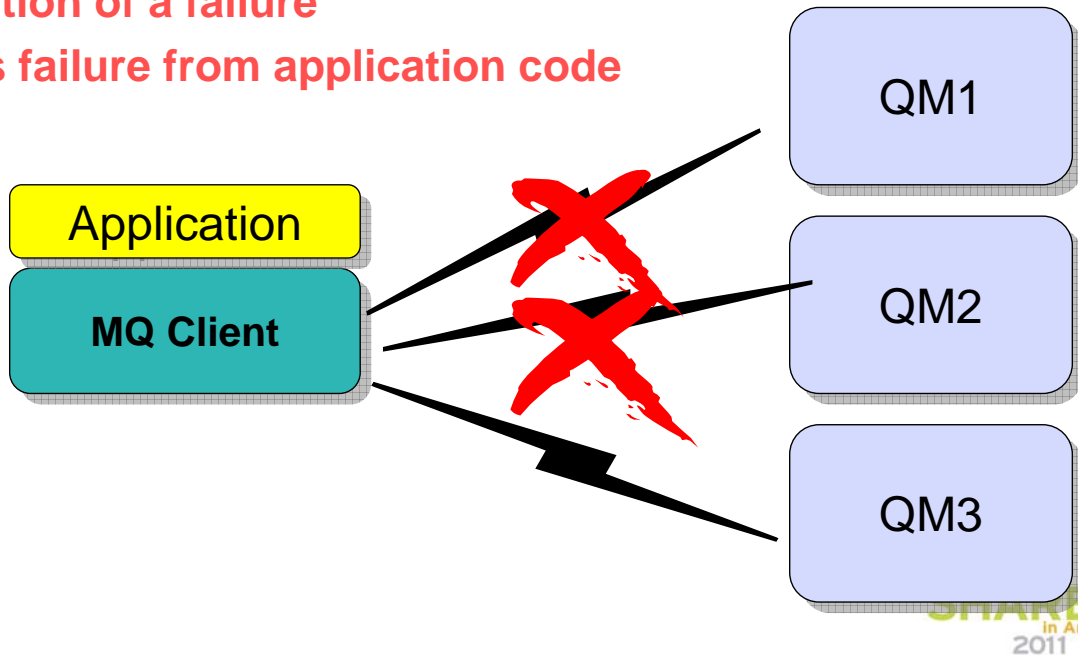


The screenshot shows the WebSphere MQ Explorer interface. The main window displays a table titled "Queue Managers on rockall". The table has columns for Name, Listener Port, QMID, Admin Obj..., State, and Instances. The data is as follows:

Name	Listener Port	QMID	Admin Obj...	State	Instances
ATS_AIX	3414	ATS_AIX_2009-04-21...	No	Running	rockall(Active)
linear	1414	linear_2009-04-21_1...	Yes	Running	rockall(Active)
V53	Unknown	Unknown	Unknown	Stopped	Unknown
V7	2414	V7_2009-04-21_12.3...	Yes	Running	rockall(Active)
V7B	2415	V7B_2009-04-21_12....	Yes	Running	rockall(Active)
V7C	Unknown	Unknown	Unknown	Running as standby	llareggub(Active), rockall(Standby)

Automatic Client Reconnection

- Client library provides necessary reconnection logic on detection of a failure
- Hides failure from application code



Automatic Client Reconnection

- Tries to hide queue manager failures by restoring current state automatically
 - Re-opens queues and other qmgr objects, re-establishes subscriptions
 - For example, if MQPUT returns error, client reruns MQCONN/MQOPEN/MQPUT internally
 - Not all MQI is seamless, but majority repaired transparently
- Uses the list of addresses in CONNAME to find queue manager
 - MQSERVER environment variable also understands list
 - MQSERVER=SYSTEM.DEF.SVRCONN/TCP/host1(1414),host2(1414)
- Can reconnect to the same or different Queue Manager
- Reconnection interval is backed off exponentially on each unsuccessful retry
 - Total timeout is configurable – default 30 minutes.
- Enabled in application code or ini file
 - Event Handler callback shows reconnection is happening if app cares

Automatic Client Reconnection: Details

- **Tries to keep dynamic queues with same name**
 - So replies may not be missed
- **Not all MQI is seamless, but majority repaired transparently**
 - eg a browse cursor would revert to the top of the queue, non-persistent messages will have been lost during restart, non-durable subscriptions may miss some messages, in-flight transactions backed out, hObj values maintained
- **Some MQI options will fail if you have reconnection enabled**
 - Using MQGMO_LOGICAL_ORDER, MQGET gives MQRC_RECONNECT_INCOMPATIBLE
- **Initially just in MQI and JMS – not the other OO classes**
 - Requires both client and server to be V7.0.1 level with SHARECNV>0
 - Server can be z/OS

Client Reconnection

N

- A TDQ is not deleted immediately when a connection fails, so a reconnecting client should find it if it lost connection due to a network failure. If the queue manager has failed over to the standby instance, the original TDQ will have been deleted but a new one will be created automatically by the client using the same name. So depending on the nature and timing of the failure, reply messages may still be accessible.

O

- MQCNO flags: RECONNECT, RECONNECT_Q_MGR, RECONNECT_DISABLED
- Options not supported: MQPMO_LOGICAL_ORDER, MQGMO_LOGICAL_ORDER

T

- MQPUT of PERSISTENT message outside of syncpoint
 - May return MQRC_CALL_INTERRUPTED
- MQSTAT may return only a 'subset' of information
 - Reconnection may have occurred during a message sequence
 - Can be used to query reconnection status

E

- Failure 'during' transaction
 - Transaction is 'doomed' – final MQCMIT will always cause a rollback
 - MQRC_BACKED_OUT returned to application
 - Non-transactional operations are allowed to complete

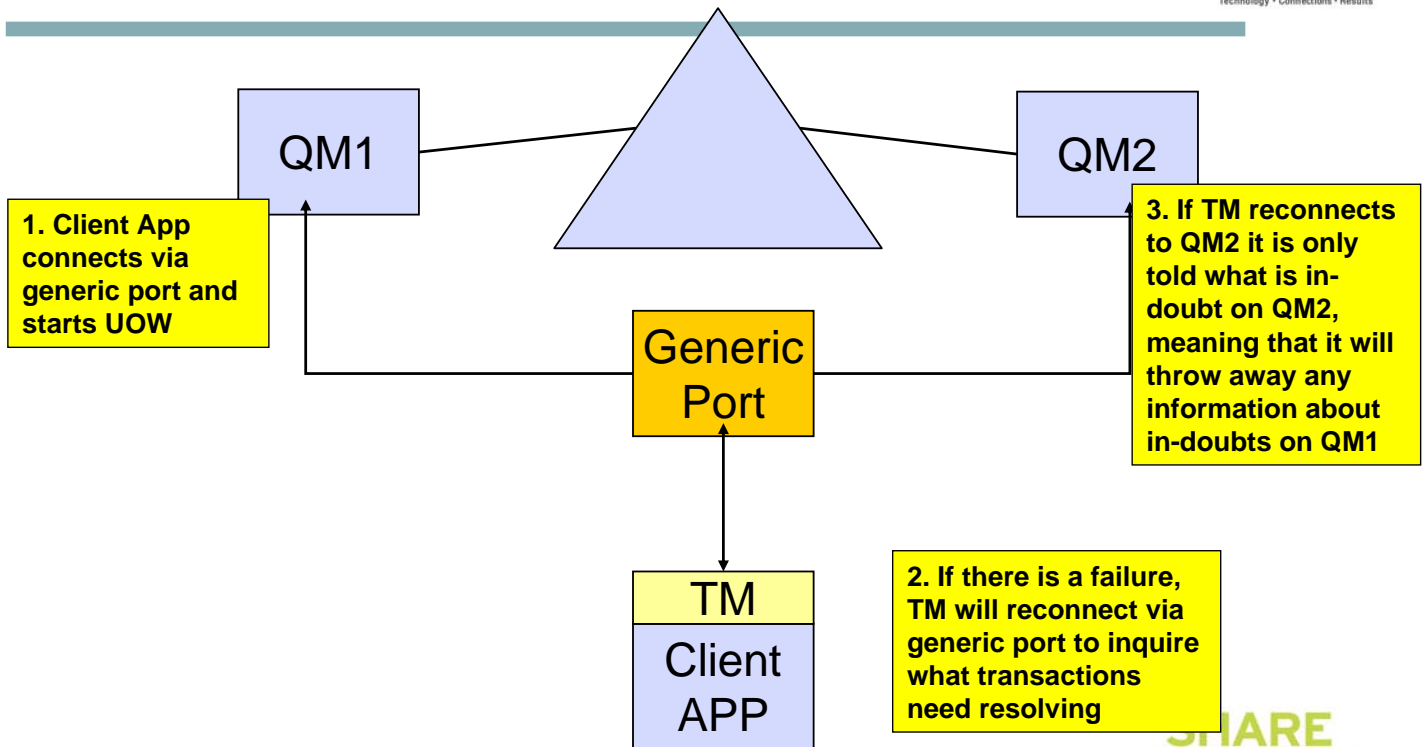
S

- XA and reconnection are mutually exclusive
 - The XA interface is too restrictive to guarantee data integrity between multiple resource managers if a connection to MQ is re-established silently

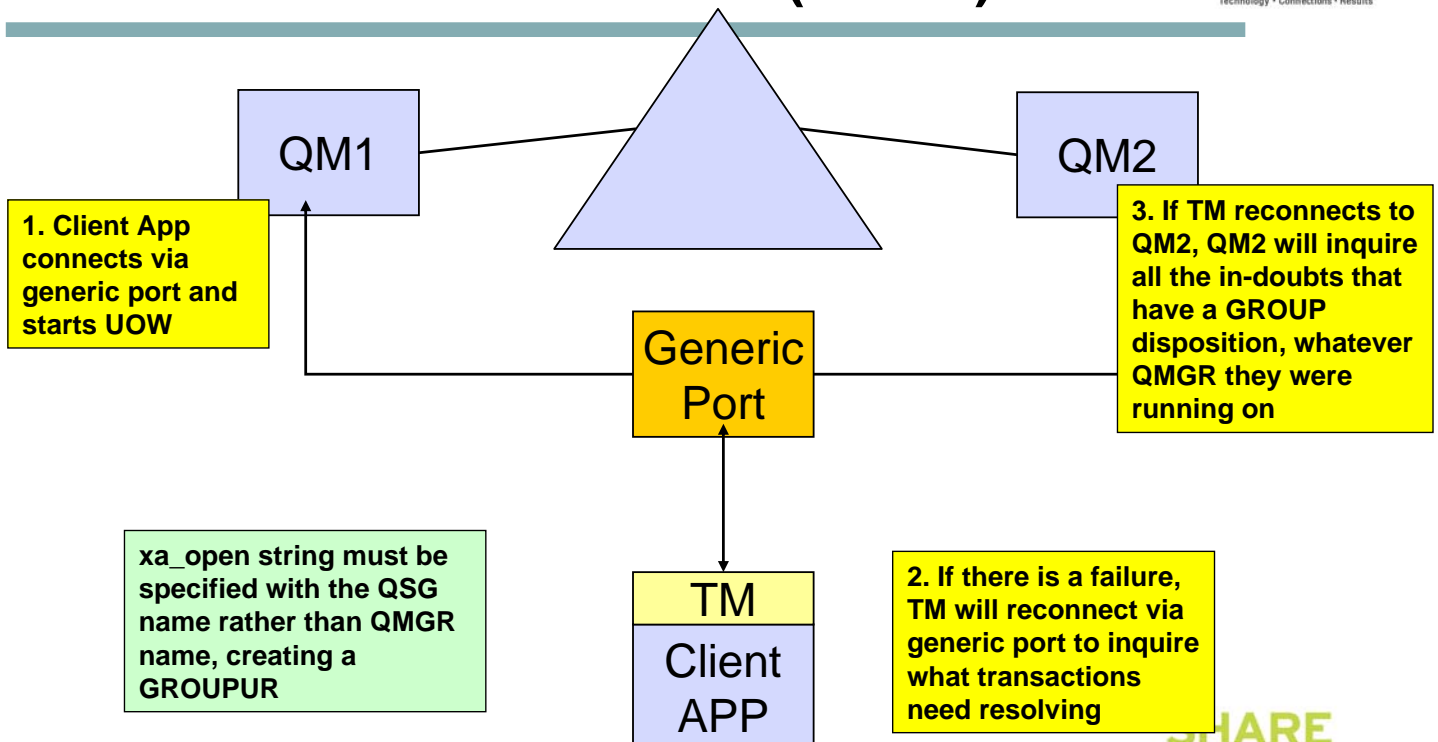
Group-level Units of Recovery for z/OS

- **A client's two-phase/global transaction can now be owned by a QSG**
 - Instead of by individual queue managers
- **These in-doubt transactions can be resolved on any QMGR in the QSG.**
 - Required for support 2-phase commit resolution while connected to the QSG
- **Requires use of the Extended Transactional Client**
 - For example, from WAS
 - Configure the WAS client connection with the QSG name rather than the QMGR name

GROUPUR: The Problem



GROUPUR: The Solution (V7.0.1)

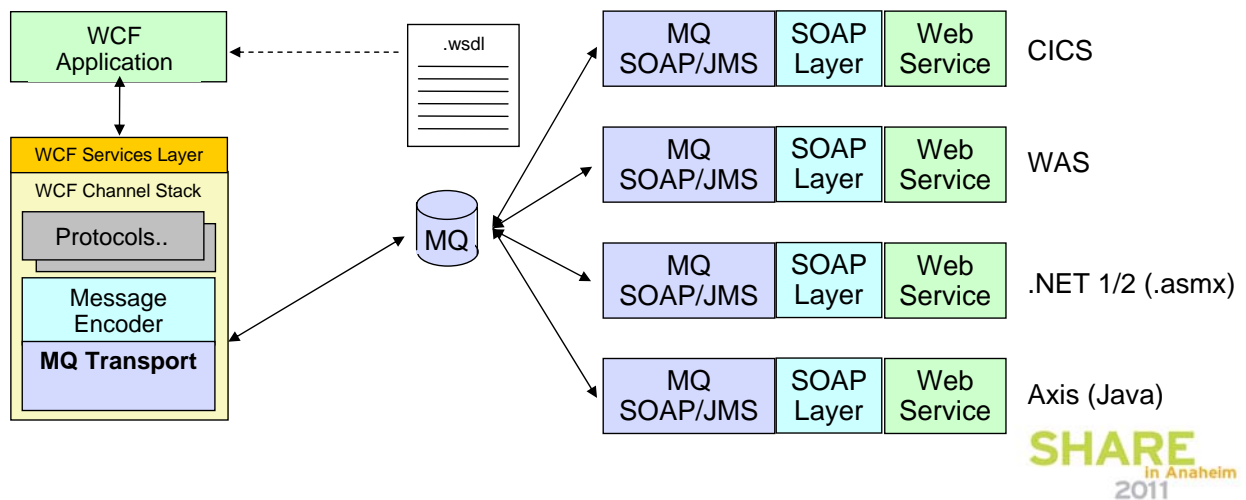


Windows Communication Foundation

- **Windows Communication Foundation (WCF)**
 - Underpins .NET Web Services and Messaging
 - Built-in Transports – e.g. MSMQ, HTTP(S), Named Pipes, TCP/IP, etc.
 - Transports can be extended with “custom channels”
 - Allows alternative transports (like MQ) to be slotted into WCF seamlessly
- **WCF support now included in product**
 - Previously an Alphaworks prototype
- **Built on XMS .Net classes**
 - Now shipped as part of the product instead of SupportPac
 - Internally, it exploits V7 API, in same way as JMS implementation

WCF: Usage within MQ

- Interface to web services hosted over MQ using SOAP/JMS
- Extending to other application servers with adoption of W3C SOAP/JMS standard (BEA, Sonic, TIBCO, Axis)



WCF: Usage within MQ

N
O
T
E
S

- What is being provided here is the first phase of features and architectural possibilities for use of WCF.
- Over time we would expect to see more features, such as better transactional support and additional security configuration. There are also options such as exploiting a WSDL definition of an application that are being considered.
- But this component provides the core piece that is needed, based on highest priority customer requirements.
- The XMS .Net implementation has been redesigned to follow the same model internally as the V7 JMS classes. These classes are now included in the WMQ product instead of being a SupportPac. They are installed automatically on upgrade to 7.0.1 if the Java Messaging and Web Services component was previously installed.
- The C and C++ XMS classes are, for now, still a separate SupportPac.

Simplified Administration

- **Remote runmqsc no longer requires default qmgr**
 - New optional “-m” flag to specify a local qmgr when “-w” also used
 - Useful in active/active HA configurations where no default qmgr may exist
- **Can set dsp permissions for any user on SYSTEM.AUTH.DATA.QUEUE**
 - Making it easier to have “read-only” administration
 - No more Authorisation Failure Events if someone tries to access this queue
 - Also being retro-fitted to V6 via APAR IZ52608: expected in 6.0.2.8

Publish/Subscribe Enhancements

- **Option to discover if no subscribers (user or proxy) during MQPUT/PUT1**
 - MQPMO_WARN_IF_NO_SUBS_MATCHED
 - MQRC_NO_SUBS_MATCHED
- **Will guarantee that no one has received the publication**
 - But does NOT guarantee that anyone will definitely receive the publication
 - For example, it is not returned if the target queue is full
- **Publish Exit**
 - When a publication is made, this exit is invoked for each valid subscriber
 - Runs “inside” the queue manager
 - Can change routing destination
 - Can change contents of message
 - Can change contents of message descriptor
 - Can inhibit publication

Publish/Subscribe Enhancements for Message Broker



- **Message Broker v7 now exploits and integrate with WMQ's pub/sub engine**
 - Instead of having an independent implementation
 - Easily connects MB's wide range of connectivity and format support to WMQ backbone
 - Common topic domain
- **V7.0.1 assists with migration for current users of MB's pub/sub**
 - Both tools and documentation provided
 - Extracts lists of topics and subscriptions to automatically populate WMQ objects
 - Help also provided with authorisations
 - Purely administrative so that developers of message flows will not see any change
- **MB still used for selection based on message content**
 - WMQ does the filtering of subscribers based on topic and message properties
 - MB parses message content to do further level of filtering

SHARE
in Anaheim
2011

Publish/Subscribe Enhancements



- N**
- Full Message Broker integration and exploitation for the administration of pub/sub requires a post-V6.1 Message Broker.
- O**
- In preparation for that, WMQ is providing tools and documentation to help migration. State is read from an existing Message Broker (or Event Broker) and – as far as possible – replicated in WMQ's definitions. Partly because of the different security models that exist, not everything can be precisely moved, but these are called out during the migration process.
- T**
- The migration tools are not on z/OS initially, but will be enabled via PTF soon after V7.0.1 is available.
- E**
- The MB application development and “runtime” aspects are essentially unchanged – existing message flows continue to work with no indication that WMQ is now doing some of the work that MB used to do.
- S**
- This will join WMQ and MB into a common pub/sub domain, with a common topic space and a common security model.

SHARE
in Anaheim
2011

Security & Monitoring

- **Command and Configuration Events for Distributed platforms**
 - Matching already-available z/OS function
 - Successful commands – MQSC or PCF – recorded as event messages
 - Configuration changes report “before” and “after” definitions of objects
 - Provides a record of who changed what and when

```
display qmgr event
AMQ8408: Display Queue Manager details.
QMNAME(V7) CMDEV(ENABLED) CONFIGEV(ENABLED) ...
```

- **SSL OCSP Support**
 - Now commonly used as alternative to LDAP-based CRLs
 - Simpler to manage as no need to have an LDAP server
 - Can use details provided in inbound certificate
 - Also can be configured within queue manager or by application code

Security & Monitoring

- N**
- Command events do not currently record the security commands like setmqaut. They also do not record other command line programs such as strmqm.
 - The formats of these events are common across z/OS and Distributed platforms.
 - Many monitoring tools will already understand the formats; they may just have to be pointed at the appropriate EVENT queue.
 - Remember that the event queues can be redefined as aliases to topics, so events can be delivered to multiple consumers via pub/sub.
 - As on z/OS, only SUCCESSFUL commands are reported. Failures due to security problems will be recorded in the usual way via Authorisation events. Command Events can be configured to not record DISPLAY/Inquire commands.
 - Use of OCSP can be driven purely from information found within the incoming certificate.
 - Alternatively it can be configured on queue manager using new AUTHINFO subtype and SSLCRLNL definitions (Distributed platforms only)
 - Configurable in CCDT using new AUTHINFO subtype (all platforms)
 - Client applications can directly specify OCSP servers using the MQAIR structure in the MQI.
- O**
- T**
- E**
- S**

z/OS Constraint Relief

- **In V7.0, the queue manager started to exploit 64-bit addressing**
 - New Pub/Sub features
- **In V7.0.1 more Queue Manager storage moves to 64-bit**
 - 64-bit Queue Indices
 - 64-bit Lock Manager
 - 64-bit Security Profile Cache
- **Can have more open queues, more messages on indexed queues etc**
- **Small message storage changed**
 - One message per page instead of fitting multiples into single page
 - May increase DASD use, especially if messages not quickly retrieved
 - But typical workloads show improved performance and reduced CPU

z/OS Constraint Relief

- N**
- In V7.0, MQ took the first step towards having a 64-bit implementation on z/OS. New elements for the publish/subscribe components were written to use this addressing.
- O**
- In V7.0.1, the next step is to move some more of the queue manager's internal tables using the new infrastructure. This is invisible to users, except it frees space for the other internal tables which can now be larger. For example, it should now be possible to have more messages on indexed queues or have more open queues without running out of storage.
- T**
- The overall effects will of course be heavily dependent on workload.
- E**
- The way messages are stored on disk has changed to put small messages (<2K) in their own page instead of trying to fit as many messages as possible into a page. While this has the potential to increase DASD requirements (more "wasted" space per message), measurements have shown improvements in throughput, and reduced steady-state pageset, because the page becomes available for reuse faster. The CPU costs can also go down because there is less processing
- S**
- working out where to store the message.

z/OS Log Compression

- **Can increase the throughput possible for persistent messages**
- **May reduce the size of your logs**
 - Dependent on your message content
 - Useful if you are DASD constrained.
- **RLE (run-length encoding) of “insert” log records for private queue messages**
 - Will not compress shared queue log records
 - SMF 115 records updated to show compression rates achieved etc
- **Controlled via zPARM option at queue manager level.**
 - COMPLOG(NONE) or COMPLOG(RLE) in CSQ6LOGP
 - Can also be viewed/controlled via DISPLAY LOG / SET LOG

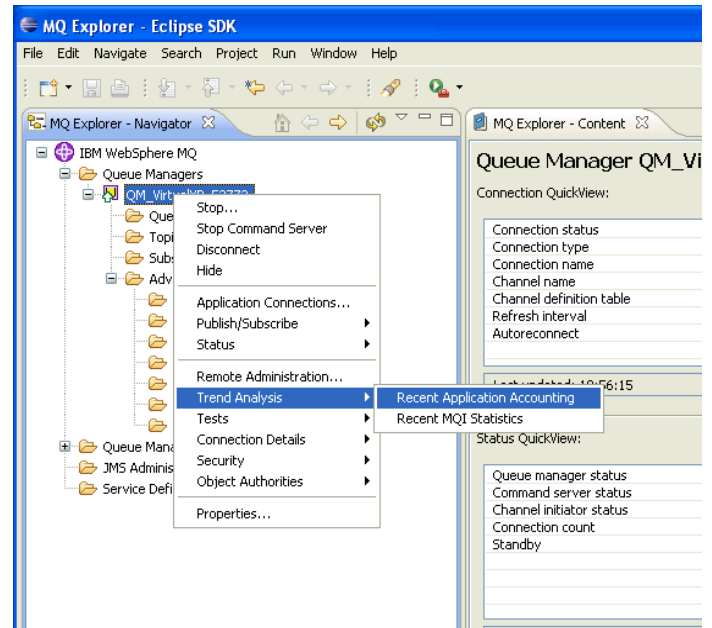
z/OS Log Compression

N
O
T
E
S

- For log compression, this improves the I/O processing, reducing the bandwidth required to log persistent messages. Clearly there is some CPU cost associated with the compression (and, if required during recovery, decompression). But this may be offset by the I/O savings – and for some customers, it’s the I/O usage that was most critical in limiting their workload.
- Compression is done using the same run-length encoding algorithm that is an option on channels.

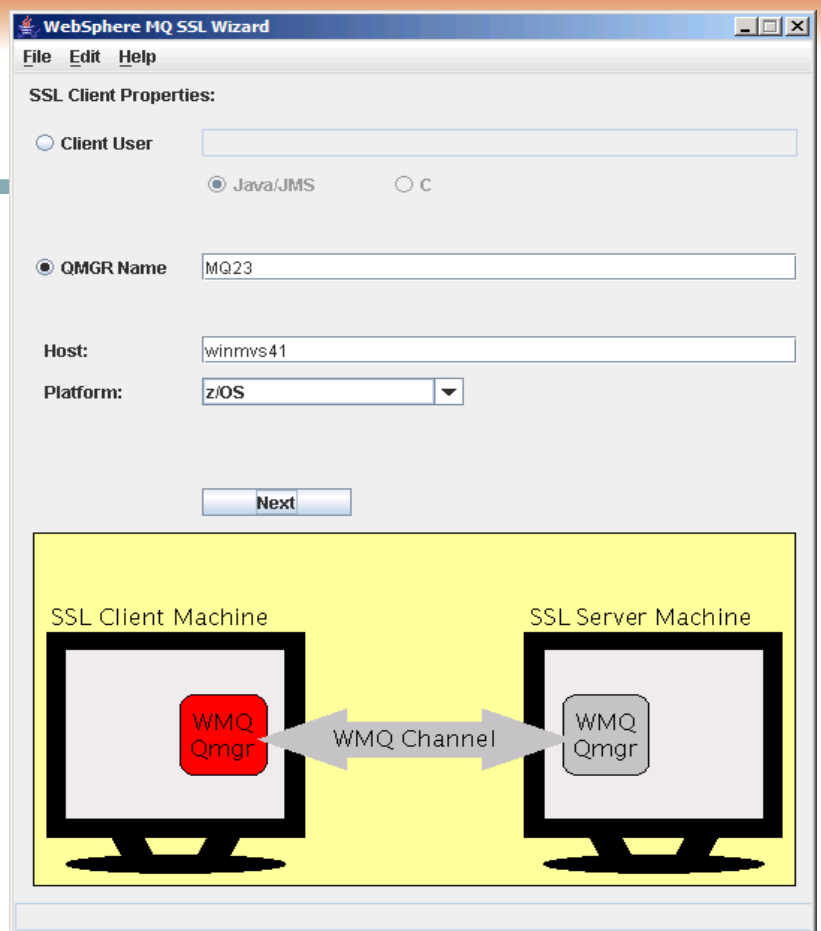
SupportPac MP08: Performance Monitor

- **Initial release of Tivoli function in a new form**
 - Performance Monitor agent for WebSphere MQ for Windows
 - Performance Monitor plug-in for MQ Explorer for Windows and Linux
- **Invoke monitoring from content menus:**
 - On a queue manager
 - On a queue
 - On a channel
- **Opens and populates chart and table views**
- **Can filter chart view and export from table view**



Security SupportPacs

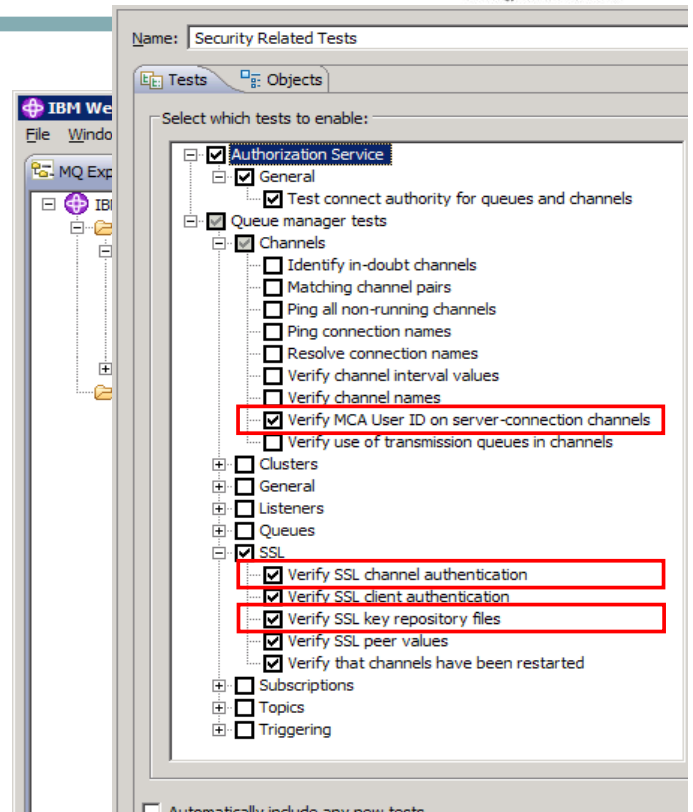
- **SSL Wizard**
 - Eases administration of SSL channels
 - Steps users through configuring channels
 - Recent enhancements for additional options
 - Out now as SupportPac MO04



Security SupportPacs

- **Security checker**

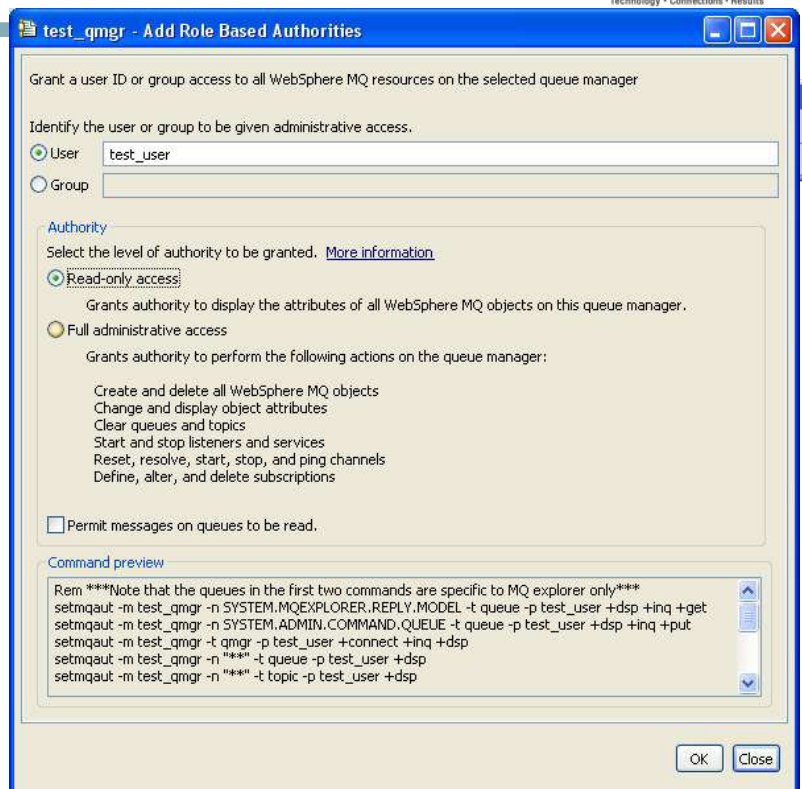
- Finds common security configuration pitfalls
- Reports exposures with suggestions on fix
- Example checks
 - OAM not configured
 - Unprotected inbound channels
 - Wrong permissions set on key database
 - Unexpected filesystem permissions
- Part of SupportPac MO05 now available – now built-in to 7.0.1.3



Security SupportPacs

- **Further configuration assistance**

- Enhanced MQ Explorer authorisation plugin shows command line for scripts
- Wizard to create administrative roles – in supportPac (not in 7.0.1.3)
- Part of SupportPac MO05 now available



New Exit Interface in MQ – Pre-connect

- **Very similar to API Exit**
 - Has parameters which match the MQCONNX verb parameters
- **Does what API BEFORE CONN exit cannot do**
 - Change to QM Name
 - Change to any MQCONNX input options (MQCNO structure)
- **Allows “CCDT” equivalent processing to be provided using some other store**
 - E.g. Look up in LDAP for centralised client channel definitions
 - Provide client connection channel details by:-
 - Filling in MQCD structure hung off MQCNO
 - Providing a set of MQCD structures in the ppMQCDArrayPtr
 - Dropping through without either of above to use CCDT/MQSERVER etc
- **If application codes MQCONN, a default MQCNO is passed to the exit**
 - Latest version of MQCNO is always provided to exit, regardless of application use

Interface details

NOTES

```

/*****
/* MQ_PRECONNECT_EXIT Function -- Preconnect Exit          */
/*****
typedef void MQENTRY MQ_PRECONNECT_EXIT (
    PMQNX  pExitParms,      /* IO: Exit parameter structure*/
    PMQCHAR pQMgrName,     /* IO: Name of queue manager */
    PPMQCNO ppConnectOpts, /* IO: Options that control the*/
                        /* action of MQCONNX */
    PMQLONG pCompCode,     /* OC: Completion code */
    PMQLONG pReason);      /* OR: Reason code qualifying */
                        /* CompCode */

#define MQXT_PRECONNECT_EXIT      23
#define MQXR_PRECONNECT          31

/*****
/* MQNX Structure -- PreConnect Exit options          */
/*****
typedef struct tagMQNX MQNX;
typedef MQNX MQPOINTER PMQNX;

struct tagMQNX {
    MQCHAR4  StrucId;      /* Structure identifier */
    MQLONG   Version;     /* Structure version number */
    MQLONG   ExitId;      /* Type of exit */
    MQLONG   ExitReason;  /* Reason for invoking exit */
    MQLONG   ExitResponse; /* Response from exit */
    MQLONG   ExitResponse2; /* Secondary resp from exit */
    MQLONG   Feedback;    /* Feedback */
    MQLONG   ExitDataLength; /* Length of exit data */
    PMQCHAR  pExitDataPtr; /* Address of exit data */
    MQPTR    pExitUserAreaPtr; /* Address of exit user area */
    PPMQCNO  ppMQCDArrayPtr; /* Address of pointers */
                        /* referencing MQCDs */
    MQLONG   MQCDArrayCount; /* Count of MQCDs referenced */
    MQLONG   MaxMQCDVersion; /* Maximum MQCD version */
                        /* requested */
};

```


New Exit Implementation

- **Looks up LDAP for client connection details**
 - Equivalent function that the CCDT provides
 - centrally managed
- **LDAP Schema provided**
- **SupportPac MA98 WebSphere MQ Connection EndPoint Lookup Exit**

New Exit Implementation

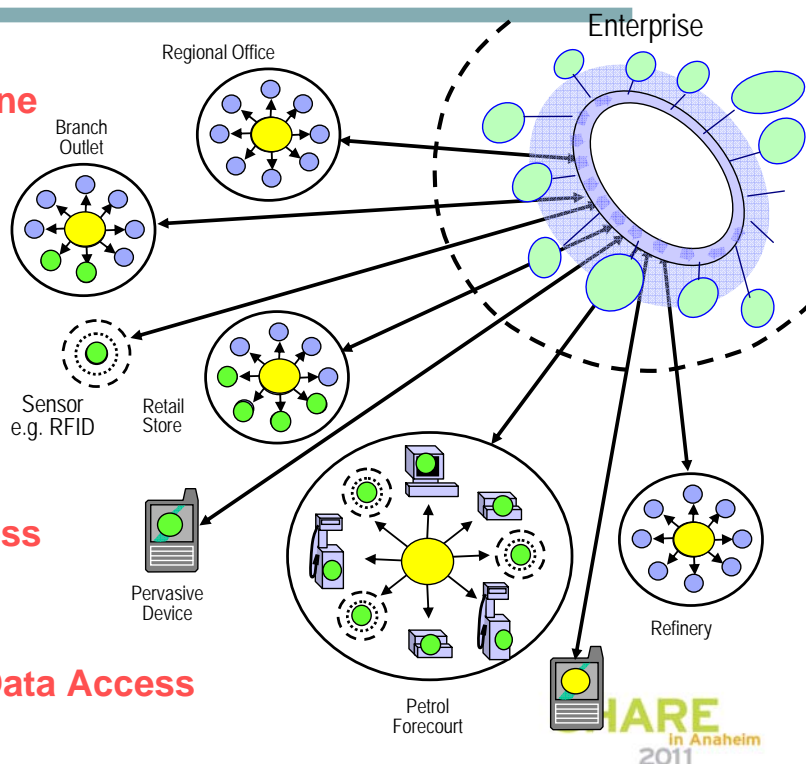
N
O
T
E
S

- A new support Pac MA98 WebSphere MQ Connection EndPoint Lookup Exit, has been developed using this interface.
 - <http://www.ibm.com/support/docview.wss?rs=171&uid=swg24028729>
- It provides the same functionality that the CCDT does, with the difference that the information looked up in the CCDT is kept in LDAP and so can be centrally managed, thus avoiding some of the downsides of the CCDT file method.
- The support pac provides an implementation of the exit as well as the LDAP Schema for use with the exit.

A Universal Messaging Backbone for a Smarter Planet

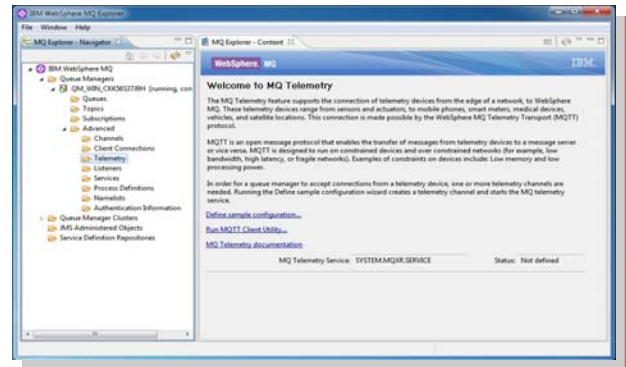


- **Extend the reach of the MQ Universal Messaging Backbone by providing appropriate messaging technology for Interconnectivity for a Smarter Planet**
- **Provide zero-administration capability where appropriate**
- **Provide Enterprise Data Access for new deployments**
- **Provide extended Business Data Access for Enterprise Systems**



WebSphere MQ Telemetry

- **New WMQ feature**
 - Separate purchasable extension
 - Initially available on Windows and Linux
 - Administration using WMQ Explorer
- **Fully integrated / interoperable with WMQ**
 - Publishers and subscribers can exchange messages with MQI and JMS applications
- **Telemetry channels enable MQTT connections to queue manager**
 - Supports MQTTv3 protocol
- **Ships with reference Java (for MIDP upwards) and C clients**
- **Includes WebSphere MQ Telemetry daemon**
 - An advanced client which also accepts connections from other clients
 - Provides ability to store and forward, network local devices.
- **MQTT Protocol**
 - Publish/subscribe messaging paradigm
 - Minimise the on-the-wire footprint.
 - Built for low bandwidth, high latency, unreliable, high cost networks
 - Expect client applications to have very limited processing resources available
 - Provide traditional messaging qualities of service where environment allows
 - Publish the protocol for ease of adoption by device vendors and third-parties



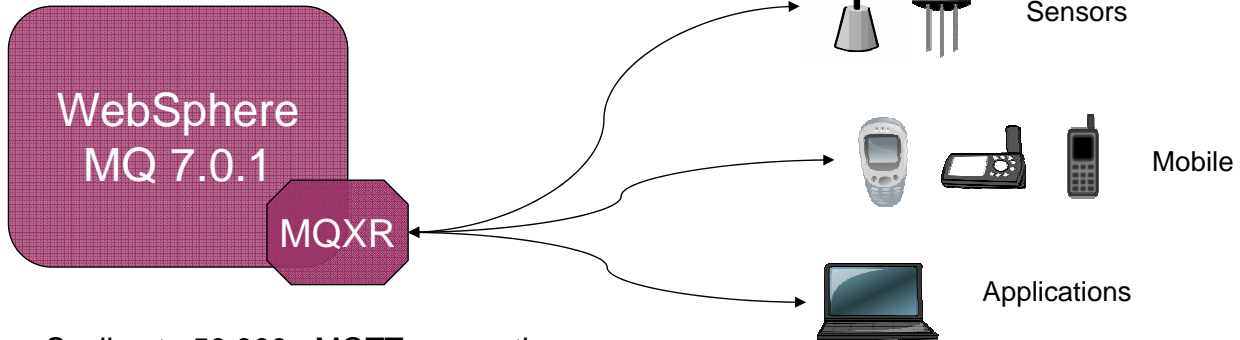
WebSphere MQ Telemetry

N
O
T
E
S

- WebSphere MQ Telemetry is a new separately purchasable feature which is an extension to the Queue Manager extension supporting mass connectivity for smart devices to the enterprise.
- It enables the queue manager to accept connections that use the MQTT protocol; a lightweight, public, low bandwidth messaging protocol for scenarios where enterprise messaging clients are too big or bandwidth intensive. The MQTT protocol has been established for at least 10 years now.
- The MQTT protocol is ideally suited to fragile or expensive networks where devices are only sometimes connected devices. Satellite phones are a good example of this. It is build with constrained bandwidth networks in mind, with a 2 byte fixed header. It is aimed at Niche platforms, for example, tiny sensors, personal devices or edge/small servers.
- The WebSphere MQ Telemetry feature offers mass scalability, accepting greater than 50,000 clients per queue manager. It operates in the same Publish/Subscribe Messaging Domain as the queue manager allowing telemetry publishers and subscribers to exchange messages with MQI and JMS applications.

Topology example: “simple” clients

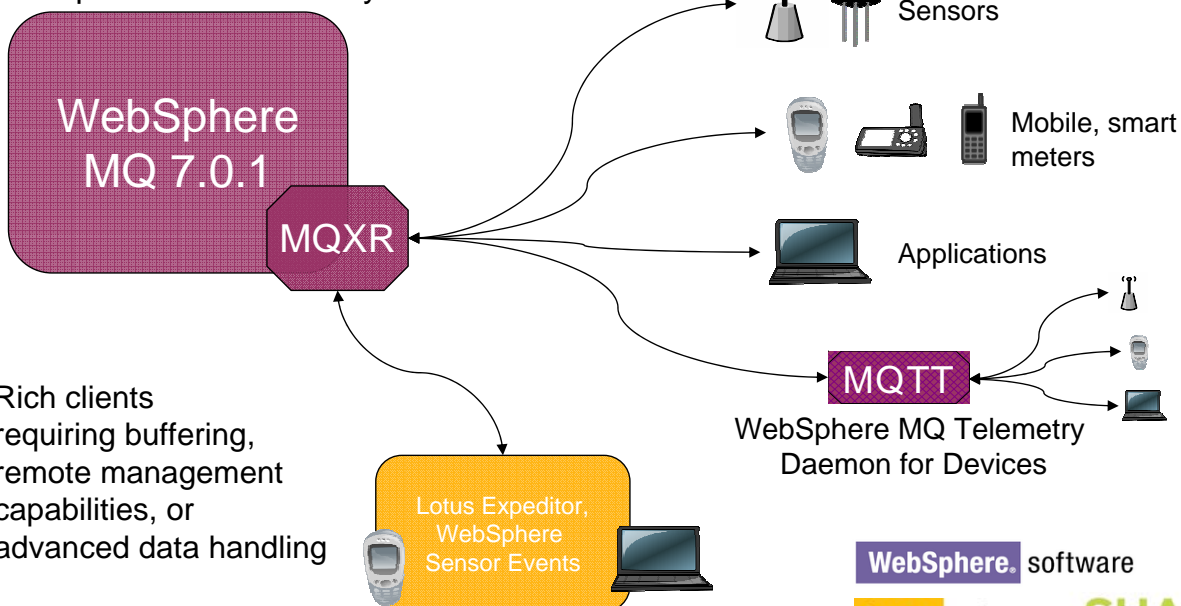
WebSphere MQ Telemetry



Scaling to 50,000+ MQTT connections to a single queue manager

Topology example: “advanced” clients

WebSphere MQ Telemetry



WebSphere MQ Advanced Message Security



- **End-to-End Message Security - Secures application data even before it is passed to MQ**
- **Extension to base MQ – No changes to existing applications**

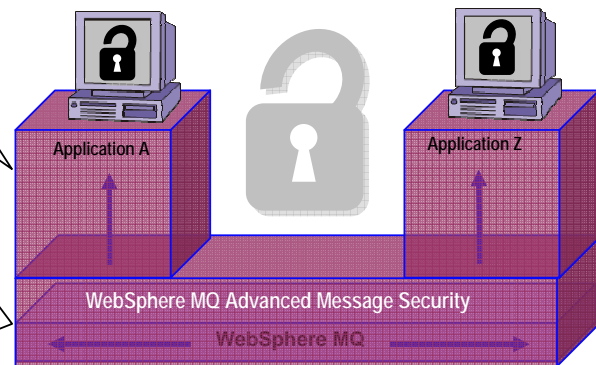
WebSphere MQ standard security:

- Message data can be encrypted in transport (SSL) but not when it resides in the queues
- Authentication is based on Operating System identifier of local process

WebSphere MQ Advanced Message Security

Supplements WebSphere MQ's security features:

- + Assurance that messages have not been altered in transit
- + Assurance that messages originated from the expected source
- + Assurance that messages can only be viewed by intended recipient(s)
- + Administered using queue based policies created from the WMQ Explorer or command line tooling.



*Securing the data and
the applications*

SHARE
in Anaheim
2011

WebSphere MQ Advanced Message Security



- N**
- A common requirement for end to end security is to help comply with regulatory requirements. For instance, in the financial sector, the Payment Card Industry (PCI) mandates the secure handling of payment details etc. Or in large WMQ networks preventing against internal message injection.
- O**
- WMQ Advanced Message Security can address these requirements by enabling message contents to be digitally signed and/or encrypted, using certificates and asymmetric cryptography, from the point they leave the sending application until they are received by the destination application, including while at rest on queues.
- T**
- No applications changes are required to use this new functionality. Its use is controlled by policies created using either the WMQ Explorer GUI or command line tools. These enable fine-grained control over which queues should be protected and how.
- E**
- S**

SHARE
in Anaheim
2011

WebSphere MQ Advanced Message Security: What is it?

- **New product that further enhances MQ security processing**

WebSphere MQ standard security:

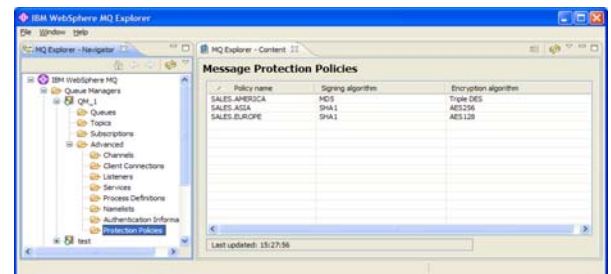
- Message data can be encrypted in transport (SSL) but not when it resides in the queues
- Authentication is based on Operating System identifier of local process

WebSphere MQ Advanced Message Security

Supplements WebSphere MQ's security features:

- + Assurance that messages have not been altered in transit
- + Assurance that messages originated from the expected source
- + Assurance that messages can only be viewed by intended recipient(s)
- + Administered using queue based policies created from the WMQ Explorer or command line tooling.

- **No pre-reqs**
- **No application or object changes required**
- **Works with MQ standard security – SSL, OAM & SAF**
- **Policies define the protection required for messages on queues.**



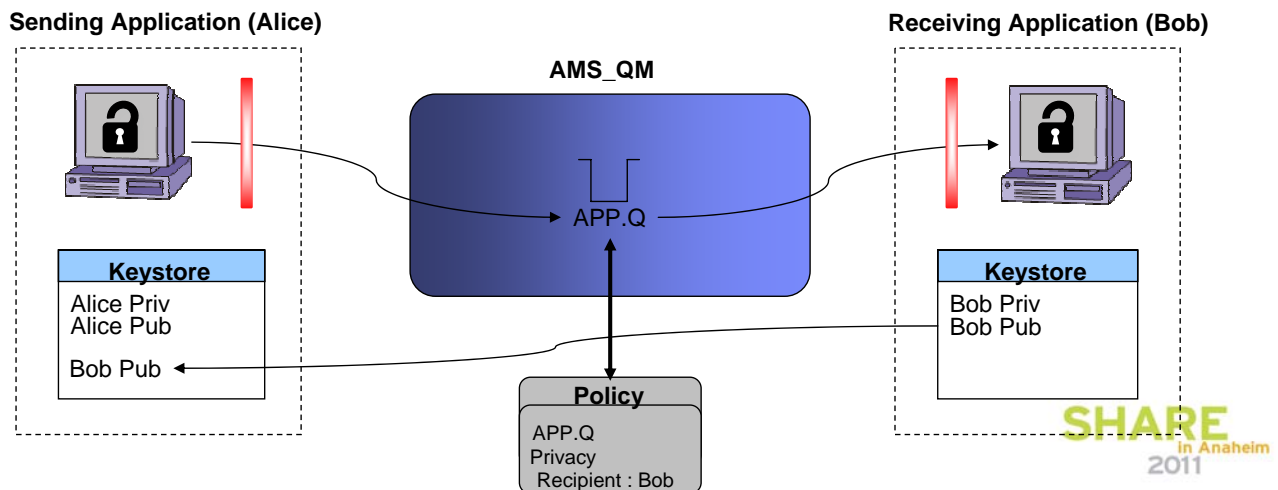
MQ AMS – What is it?

N
O
T
E
S

- WebSphere MQ Advanced Message Security (AMS) is a new (separately priced) product that further enhances WebSphere MQ security processing. It is designed to assist with requirements such as PCI DSS compliance by enabling secure message transfers at application level.
- It provides assurance that messages have not been altered in transit; that messages originated from the expected source, by validating the sender when processing messages; and that messages can only be viewed by intended recipient(s) which is useful when sending confidential information.
- MQ AMS has no prerequisite products (apart from MQ) and has a significantly simplified installation and configuration compared to its predecessor product (WebSphere MQ Extended Security Edition). You can be up and running in minutes.
- MQ AMS works in conjunction with SSL, so you can choose to use either or both depending on your requirements. It also works in conjunction with the WebSphere MQ authorisation model (OAM and SAF).
- No changes are required to WebSphere MQ applications. It works with local applications and clients, including Java. Works for the Point to point messaging domain only, but not for Publish/Subscribe. Support is provided for WebSphere MQ V6 and V7 (requires minimum of MQ V6.0.2.8 or MQ V7.0.1.2); on distributed platforms excluding iSeries. Support is also provided for z/OS for V6 MQI verbs only (i.e. no asynchronous consume, message properties etc). Also no changes are required to existing object definitions.
- Administratively controlled policies are used to define which queues are protected and how.

Advanced Message Security – Key Elements

- Message Interceptors
- Protection policy for the application queue
- Public / private key pairs
- Recipient's public key



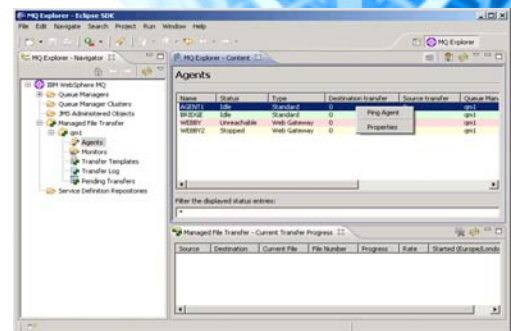
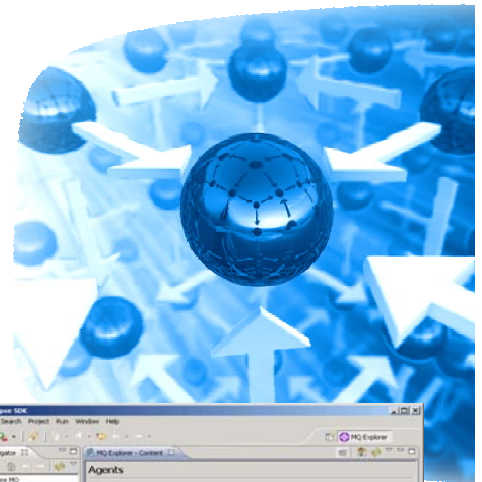
Key Elements

- N
O
T
E
S
- This shows the main moving parts within WebSphere MQ Advanced Message Security.
 - Firstly, message interceptors are used within the sending and receiving applications to detect when end to end security is required and the level of signing and encryption required.
 - The interceptors determine the required behaviour by interrogating the policy defined for the queue the application is to use and will describe the services required (if the message should be digitally signed and/or encrypted or neither), the algorithms used and the identity of allowed senders and receivers if required. Policies are created by using either the WebSphere MQ Explorer or command line tools.
 - Digital certificates are used to sign and encrypt the data as required, a typical set-up being shown for the local key/trust stores.

What's New in WebSphere MQ FTE v7.0.3



- Web transfers and RESTful interface
- Conversion of payload between messages and files
- Ability to start an agent as a Microsoft Windows Service
- Enhanced display of agent status
- JEE Database logger
- Enhanced Integration with WMB V7.0.0.1
- Capability to encrypt file data using MQ AMS V7.0.1.
- Additional platform support



What's New in FTE v7.0.3 – Notes



- N**
- O**
- T**
- E**
- S**
- A Web Gateway component that provides a web-based interface for users that wish to exchange file data with a WMQ FTE network. It uses a RESTful API for sending files into and receiving files from a WMQ FTE network.
 - Convert payloads between file and messages (AKA “file to message” or “message to file”)
 - A Windows Service to automatically start the WMQ FTE agent and WMQ FTE database logger components at Windows system start-up or after a failure, increasing the availability of these components.
 - View deployed WMQFTE Agent in the WebSphere MQ Explorer GUI including their current status.
 - Database logger component (which previously ran as a stand-alone operating system process) can now be deployed and managed as a JEE application, simplifying Web Gateway deployments and re-using JEE administration skills.
 - Tight integration between FTE and WebSphere Message Broker for File Processing. This enables ESB capabilities to be applied to file data such as the ability to parse and transform files and process into messages, files, events, service requests etc.
 - Encrypt file data in the payloads of messages resting on WebSphere MQ queues using WebSphere Advanced Message Security v7.0.1.
 - Support for new platforms including Solaris on x86 hardware.