



**IBM**

---

# **MQSeries V5 Features & Functions**

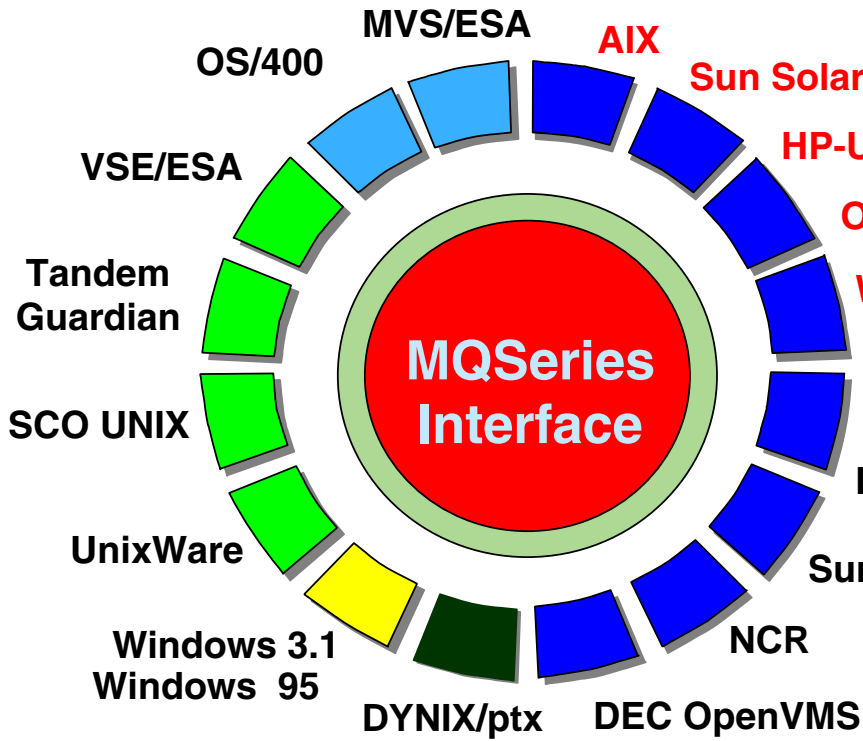
<http://www.software.ibm.com/ts/mqseries>

---

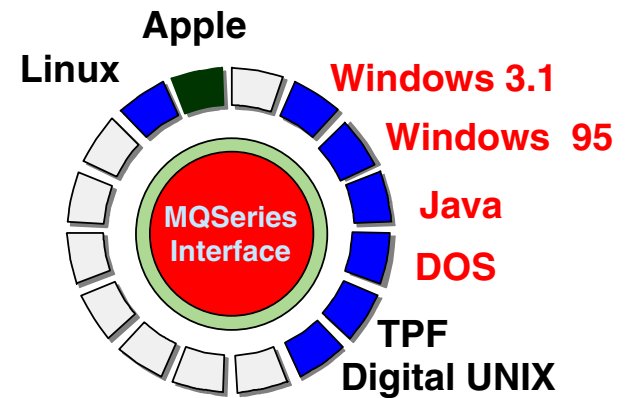
**MQSeries**  
*Commercial Messaging*

# Platform Coverage

## Queue Managers



## Clients only



### Simultaneous availability for key server platforms

- UNIX
  - AIX, HP-UX, Sun Solaris
- PC
  - OS/2, Windows NT

### Clients

- DOS, Windows 3.1, Windows 95, Java
- All server platforms have clients

### Other MQSeries platforms

- May be upgraded in follow-on releases
  - OpenVMS,
  - NCR, SINIX, DC/OSx, SunOS
- N.B. MVS, OS/400, Tandem Guardian platforms follow separate development processes



## Platform Coverage - Notes

---

MQSeries is now available on more than 20 platforms and there are clear issues with keeping each member of the product family as up to date as is appropriate for that platform. One approach to addressing this issue is to develop MQSeries platforms in parallel. This is quite feasible for platforms which are a part of the common code base for MQSeries and has been done for this release on the following platforms:

- AIX
- Windows NT
- Sun Solaris
- HP-UX
- OS/2
- DOS, Windows3.1, Windows 95 and Java clients

The choice of platforms is based upon customer demand and the commercial importance of each platform. Clearly, some platforms which are part of the common code base have been omitted from this release. Again, based on customer demand, these platforms may be brought up to this new functional level.

Note that this style of development is only possible for platforms which are a part of the common code base. Thus, it is not possible to include MVS or OS/400 platforms in this (particular) effort.

### Resource coordination

- New MQI verb ... MQBEGIN

### Distribution Lists

### Partial GET/PUT

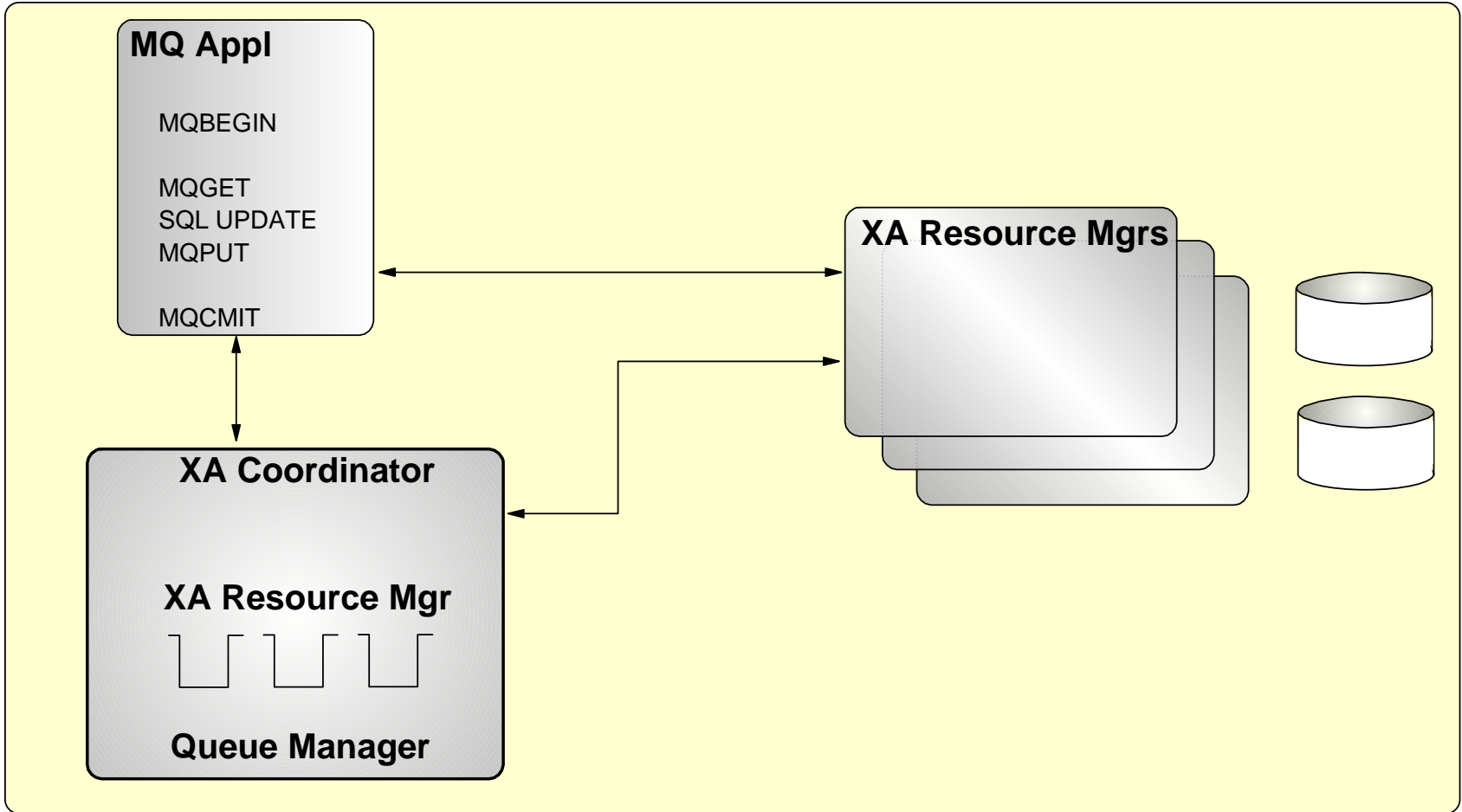
- Large message support
- Small queue support

### 100 MB messages

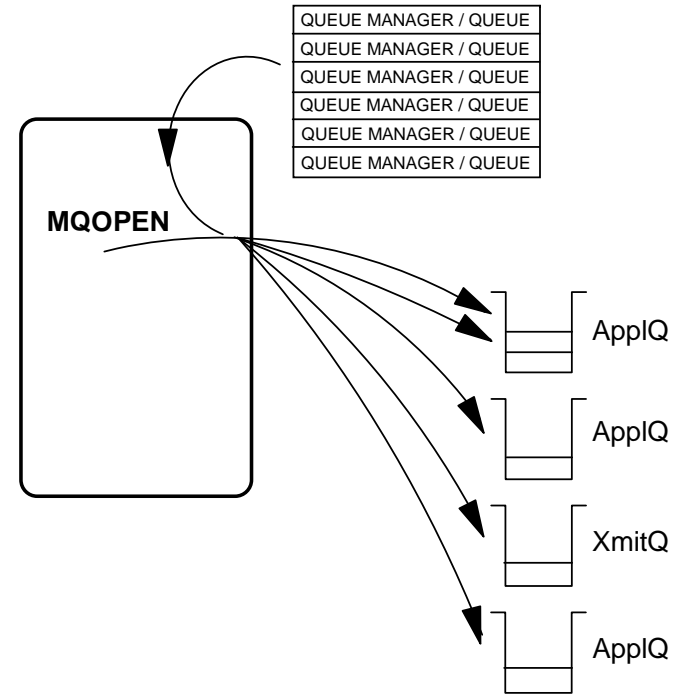
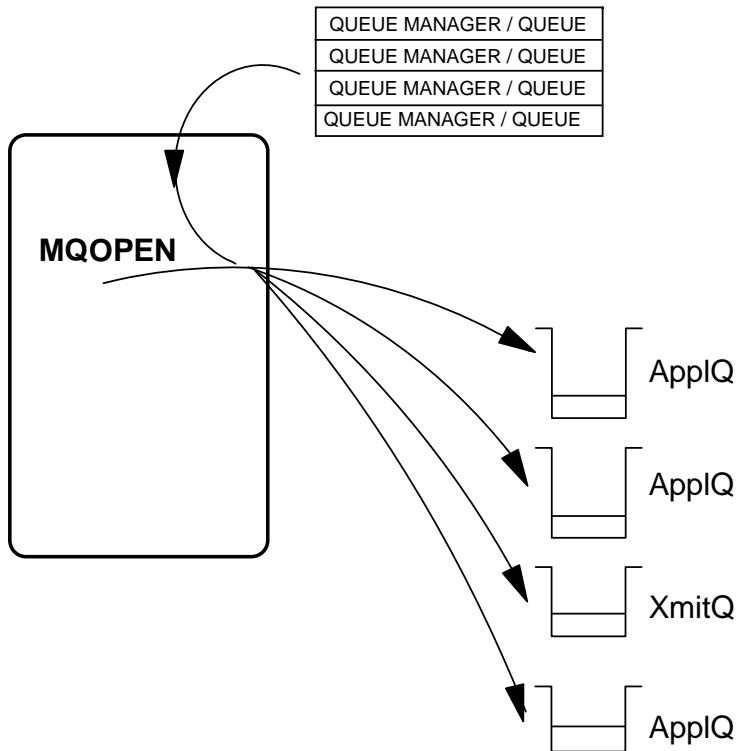
### Integration of important SupportPacs

- C++ support
- Windows 95 32-bit client
- Java support

## Resource Coordination:

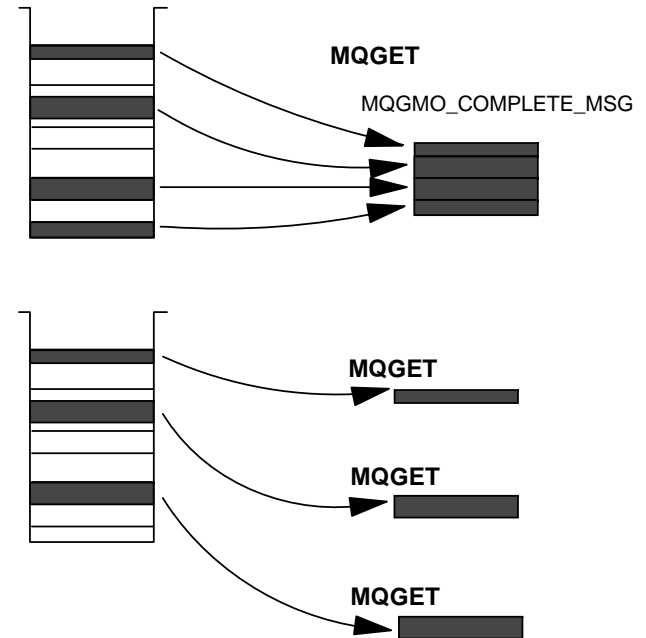
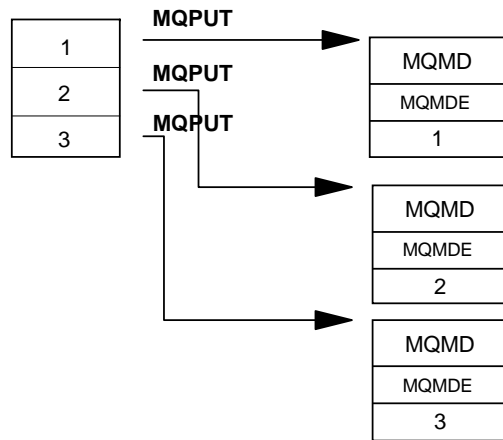
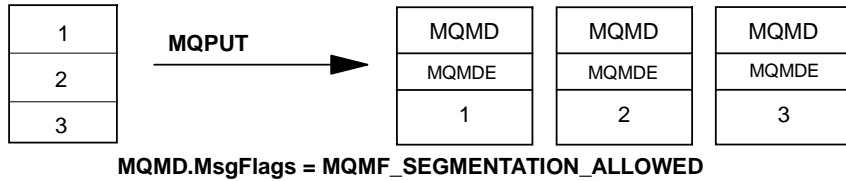


## Intelligent Distribution Lists:



**One copy per XmitQ**

## Partial PUT/GET







## Major New Functions - Notes

---

When MQSeries is used in conjunction with other (X/Open XA) Resource Managers it is possible to coordinate updates to MQSeries resources and updates to those other resources. Before MQSeries V5, it was necessary to have a Transaction Manager present in order to achieve this coordination. MQSeries V5 now provides Resource Coordination support to enable coordinated updates without the use of a Transaction Monitor. Note that this DOES NOT mean that MQSeries is evolving into a Transaction Monitor, MQSeries is only providing the resource coordination function - a small piece of the total functions provided by a Transaction Monitor.

Distribution Lists provides the capability to place messages on multiple target queues with a single MQPUT command. The MQOPEN command now accepts a list of queue names as input rather than a single name. The Object Handle returned now refers to a list of destinations rather than a single destination. Then a single MQPUT will place the same message on these multiple destinations.

The operation of Distribution Lists attempts to optimise the distribution of messages by controlling when multiple copies of messages are created; if more than one destination in a distribution list resolves to the same target transmission queue then the processing will place only ONE copy of the message on the transmission queue. At the target system, the message is then copied to the multiple target destinations.

In order to cater for situations where there are mismatches between the size of messages and the message capacity of the target queue, there are two new facilities introduced. Firstly, the maximum size of messages is increased from 4MB to 100MB. Secondly, the new Partial Get/Put facility allows messages to be segmented when it is too large to be placed on the target queue. This segmentation is automatic and results in multiple messages being placed on the target queue, each message representing a segment of



## Major New Functions - Notes

---

the original message When the message is taken from the queue, the Queue Manager will automatically recompose the message back to its original form. This segmentation is available for any queue size - it does not have to apply only to very large messages. The segmentation mechanism is exposed and so applications may place segmented messages directly onto a queue and may remove individual segments of a message from a queue.

In order to make MQSeries functions available in a timely manner, a lot of functions/facilities were made available via the SupportPac mechanism. Many of the facilities previously available as SupportPacs are now available as a part of the base product.



## Internet Support

---

**Java clients**

**Internet Gateways**

**(Web based administration)**

**HTML publications**



## Distributed Messaging

---

### Performance

- Fast messages ... CSD for some platforms
- Internal channel architecture
- Trusted bindings for MCAs
- Channel batch interval

### Novell SPX support

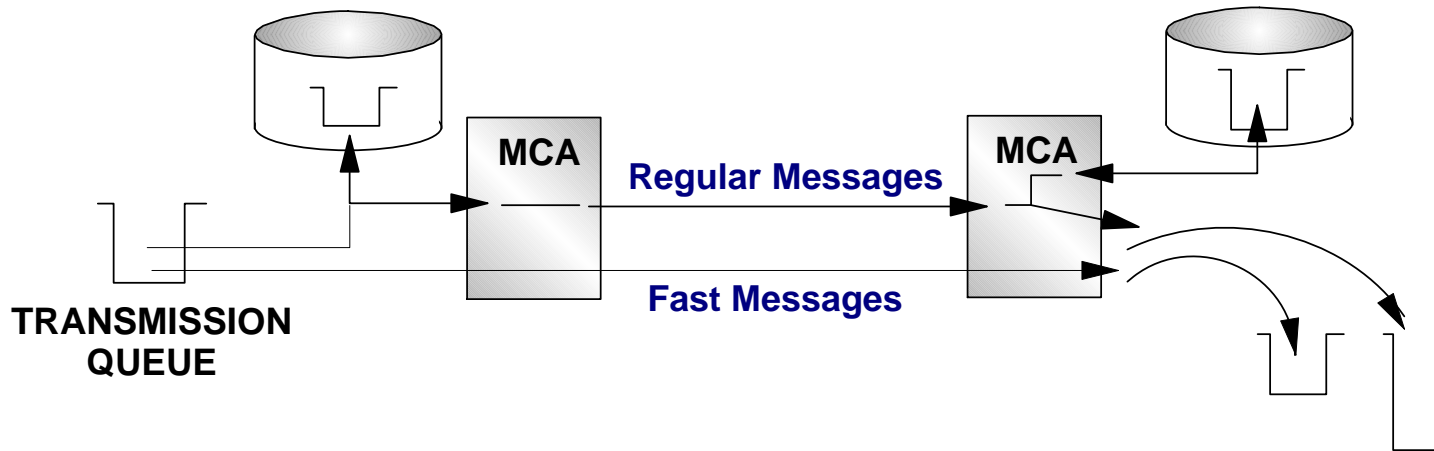
### Dynamic channel definition

- Receiver and MQI server channels

### Chained MCA exits

### Save channel status over restart

### Remove process definition for triggered channels



## No commit processing for *non-persistent* messages

- Reduced processing requirement for non-persistent messages
- Faster throughput...messages are available immediately to applications

## Persistent messages benefit as well

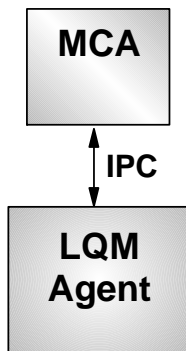
- Persistent messages may share the channel
- Standard commit processing

## Fast Messages are *optional*

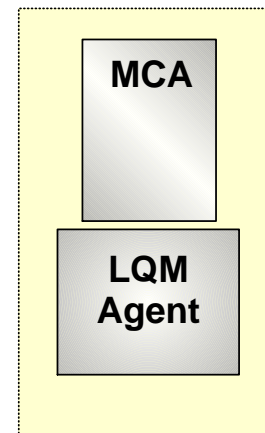
# Trusted Bindings

---

Regular Channel



Channel using Trusted Binding



## Single process

- For both MCA and Local Queue Manager Agent
- No IPC overhead for MCA MQI commands



## Distributed Messaging - Notes

---

There have been many improvements made to the distributed messaging component of the Queue Manager. Performance issues have been addressed in the following ways:

- For non-persistent messages, both the processing required has been reduced and the throughput improved by optionally (via a channel definition) removing non-persistent messages from the existing commit processing performed by message channel agents. This serves to reduce the overall processing required for a message and to make the message available to receiving applications before the end of a batch.
- The mechanism used for processing messages and assuring once/once-only message delivery has been improved to remove the use of an MQSeries queue as a log mechanism. A more efficient mechanism is now used.
- An MCA may now be tightly integrated into the Queue Manager by use of Trusted Binding support
- End of batch processing is expensive (in processing terms) and is often made more expensive when a batch is 'prematurely' terminated by a transmission queue becoming empty. BatchInterval allows a batch to be kept open for a minimum period of time, reducing the overheads of repeated commit processing

Support for Novell networks today is only possible via some form of emulation over the Novell protocols. This deficiency is addressed by direct support of Novell SPX. In order to reduce some of the system management overheads associated with channels, receiver and SVRCONN channels may now be automatically defined on receipt of an inbound attach request. This mechanism may also be used to populate a new Queue Manager with the required channel definitions.

## Distributed Messaging - Notes...

---

In order to accommodate multiple MCA exits of the same type, it is now possible to chain exits. Thus, multiple message, send or receive exits may be used by a single MCA without the need to merge the exits into a single entity. The current status of channels is retained across a Queue Manager restart. Thus, if a channel is disabled or (more importantly) in RETRY state then this is restored when the Queue Manager is restarted.

Process definitions - for triggered channels - play a very limited role. The triggering rules have been relaxed for the channel initiator so that a process definition is no longer a requirement for triggered MCAs. Note that a Process definition may still be used if desired.



### General

- Application Programming
  - IBM COBOL support
  - PL/1 support ... OS/2, windows NT and AIX
  - C++ support
  - threaded application support
  - improved control of MsgId/CorrelId
- Reference messages ... for distributed messaging
- CICS/Encina support for *all* platforms
- DCE directory support for remote queues

### Security

- DCE exits for
  - authentication of remote systems
  - message integrity
  - message privacy
- Windows NT security domain integration
- Controlled loading of optional code



## Additional Support - Notes

---

For application programming, there is the addition of new programming language bindings. IBM COBOL is supported and PL/1 support is included - due to customer demand on certain platforms. The support for threaded applications is made more consistent across the platforms, enabling a connect from each thread within a program. Note that MQSeries connection and object handles cannot be shared by multiple threads. The existing support for C++, available via SupportPacs is now made a part of the base product.

Reference message support is intended for applications which pass large message to application on remote Queue Managers. The requirement to place the message data on an MQSeries queue is removed; a reference to the message data is placed on the queue and the appropriate message data is retrieved (by a supplied MCA Message Exit) when the message is transferred to the target system. This avoids the need to remove the message data from one form of disk storage to another.

The support for CICS and Encina transaction monitors is made available for all platforms that have both MQSeries implementations and CICS and/or Encina implementations. In particular, support for CICS and Encina on the Sun Solaris platforms is included.

For security, support for DCE authentication, integrity and encryption is provided via MCA exits. Thus, it is possible to provide secure communications between servers and between clients and servers. The DCE functions are provided via the gssapi and both source code and object code are provided, allowing the exits to be modified, as required. For the windows NT environment, there are some extensions to the way that security domains are supported to improve the integration between MQSeries and the windows NT operating system.

### Administration and Management

- MQSC
  - syntax improvements
  - interaction improvements
- Stop receiver channels
- MQI server channel status
- Licence management

### RAS

- FFST improvements
- New trace functions
- Log dump utility

### Installation

- Automatic creation of default objects

## Additional Support - Notes

---

The MQSeries commands have been extended to improve the user interaction with this facility:

- There were several areas where the defaults provided by the commands were not intuitive. These defaults have been addressed. (For instance, the default for DISPLAY QUEUE is now ALL)
- The way that data is displayed is optimized so that as much data as possible is displayed in a window as possible; instead of a single column of information, there are 2 columns wherever possible.

In order to allow receiver channels to be more efficiently terminated, there is a heartbeat mechanism introduced to help the receiving channel to notify the sending side that the channel should be terminated.

To aid failure diagnosis, there have been changes to both FFST and trace mechanisms. FFSTs have been enhanced where necessary to provide more appropriate error feedback. Trace has been enhanced to provide additional information on entry and exit from various MQSeries components and to provide more information within these components.



# IBM Software Server Integration

---

**OS/2, AIX, Windows NT**

**Look and feel, OOB experience**

- Enhanced installation
- Eagle versioning

**RAS, Service**

**(Web browser based administration)**

**HTML publications**



## IBM Software Server integration - Notes

---

MQSeries V5 is a part of the IBM Software Server Family, providing a common set of functions and facilities for all products across the range of Software Server products. For MQSeries V5, this is applicable to AIX, OS/2 and Windows NT. The principal functions provided are:

- Common look and feel and 'Out of the Box' (OOB) experience, particularly for product installation
- Common mechanisms for failure diagnosis and for providing information to IBM service
- Web based functions:
  - ▶ Administration facilities for starting/stopping Queue Managers, defining and displaying MQSeries resources definitions.  
Note that this function requires a specific level of Java support, which is not yet at a production level and so this item will ship at a later date.
  - ▶ Publications  
All MQSeries publications are now available as HTML, viewble via any Web browser



## Customer Requirements Addressed

---

- R0383: Cross platform IVP
- R0025: Strengthen the IVP
  
- R0077: PL/1 Support for Mayflower
- R0488: PL/1 support for MQSeries OS/2
- R0098: C++ support on OS/2
- R0820: Solaris C++ support
  
- R0663: Log dump utility to support both linear and cyclic logs
  
- R0736: User-defined Report options in message header
  
- R0113: DCE support for security and directory
- R0613: Queue manager to Queue Manager authentication
- R0693: Header encryption
- R0676: DCE security services in MQSeries for HP-UX

## Customer Requirements Addressed...

---

- R0464: Channel synch via IP address insufficient
- R0316: IP address dependency at MQ recovery
- R0484: Channel performance for low transaction rates
- R0597: Ability to display channel status for clients
- R0664: AIX: display negotiated batch size for channel
- R0723: Dynamic channel generation (Partial)
- R0807: Batchsize based on time window
- R0308: Mover Performance
- R0455: Performance improvements (Partial)
- R0634: Improving communication response time with MQSeries
- R0832: Channel treats non-persistent msgs as recoverable
- R0849: Clean up RUNMQLSR when entering ENDMQM
- R0850: Clear AMQXSSVN when RUNMQLSR process killed
- R0598: Command needed to stop a channel listener
- R0691: Command to shut down listener processes





## Customer Requirements Addressed...

---

- R0411: IPX support
- R0280: Mayflower needs Netware support as in SSI products
- R0594: MQ client support for windows NT and windows 95
  
- R0166: MQSeries to be external syncpoint manager
- R0292: MQM to act as resource coordinator
- R0361: DB synchronisation
- R0574: Unit of work coordination for XA resource managers
- R0575: Unit of work coordination for XA resource managers
- R0196: Unit of work synch w/ DB2/2, not CICS OS/2
  
- R0760: MQ Tuxedo V5 support for HP-UX, AIX
  
- R0167: Support for 'Distribution Lists'
- R0168: Intelligent distribution
- R0724: Distribution lists
- R0176: Enable 1 MQPUT to result in a PUT to >1 queue

## Customer Requirements Addressed...

---

- R0229: MQSeries message sequencing capability
- R0410: Send series of messages as if one
- R0227: Multi segment messages
- R0169: User data segmentation
- R0253: Support messages longer than 4MB
- R0601: Eliminate the 4 meg message size limit
- R0650: Elimination of 4mb maximum message size
- R0729: Message size big enough for multi media objects
- R0601: Eliminate the 4 meg message size limit
- R0017: Message size maximum of 4MB is too small
- R0118: Practically unlimited message size

Total: 52