WebSphere MQ Everyplace V2.0.2

Contents

Programming reference

If you have installed extra reference plug-ins they appear in this section in the Contents.

The plug-ins and their contents are:

API References

- Java Programming Reference
- C Programming Reference

C Programming Guides

- C Bindings Programming Reference

This section also contains:

JMX Attributes and operations

The topics in this section describe the attributes and operations that each JMX-instrumented MQe resource can access.

Each resource is described here by up to three sub-topics:

1. **Attributes**

   In the table on these pages:

   - The **Attribute Name** is the name returned by the `Attribute class getName()` method or the `MBeanAttributeInfo getName()` method.
   - The **Attribute description** describes the attribute.
   - The **Attribute Type** is the String representation of the attribute data type retrieved using the `MBeanAttributeInfo getType()` method.
   - The **Read/Write** column indicates whether an attribute is read-only (RO), or can also be updated (RW).

2. **Operations**

   These pages show:

   - The **Operation Name**
   - The **Operation description**
   - The **Parameter**. For more details on what each parameter is, see the following sub-topic **Operations parameters**, where present.

   Unless otherwise stated, the return type of all operations is `java.lang.Void`. To ensure that default values are used, leave parameter entries blank on a GUI interface or use a value of null in a programmatic interface.

3. **Operations parameters**

   These pages explain the parameters on the preceding page, showing:

   - **Parameter name**
   - **Parameter type**
   - **Parameter description**

Admin MBean

Attributes

Table 1. Admin MBean attributes

Attribute name	Attribute description	Attribute type	Read/Write
AdminMsgExpiry	The default expiry time (in milliseconds) for sent admin messages	java.lang.Integer	RW
AdminQName	The default name of the admin queue to use for sending admin messages	java.lang.String	RW
CacheInterval	The default time (in milliseconds) to retain cached values (see below)	java.lang.Long	RW
InquireOnConnect	Indicates whether remote queue manager resources are inquired upon and MBeans for these created when new connection definitions are created	java.lang.Boolean	RW
MsgPollInterval	The default time (in milliseconds) to wait for admin reply messages when performing remote admin	java.lang.Long	RW
LocalMsgTimeout	The default length of time (in milliseconds) to check for admin reply messages when performing local admin	java.lang.Long	RW
RemoteMsgTimeout	The default length of time (in milliseconds) to check for admin reply messages when performing remote admin	java.lang.Long	RW

Note:

1. In the current implementation, the values set for these MQeAdminJmx attributes are not persistent between deletion and re-creation of the MBean so must be reset whenever a new MQeAdminJmx MBean is created. This effectively means that remote queue managers must be refreshed so that their children are all visible via JMX.
2. The attribute *CacheInterval* relates to a caching mechanism employed in the MQe JMX interface. Attribute values are cached for this duration and inquires during the cache interval do not result in a refresh of the values. Attribute values are refreshed whenever an update is done (that is to say, the value of one or more of the attributes is changed). In that instance, the cache clock is reset to zero and no refresh will take place until either the *CacheInterval* expires or another update is done. The default value is 0.
3. *InquireOnConnect* is used whenever new connection definitions are created or loaded. If this attribute has a value of true, an attempt will be made to inquire upon the remote queue manager accessed directly by this connection definition. If the inquire is successful, an attempt will be made to create MBeans for all of the remote queue manager child resources. If the attribute value is false, an MBean will be created for the remote queue manager but no MBeans will be created or registered for its child resources. The creation of the children MBeans can be effected when desired by using the MQeRemoteQueueManagerJmx refresh() method. The default setting is false which means that no children MBeans for remote queue managers are created at start-up.
4. *MsgPollInterval* specifies how frequently the admin reply message should be looked for. The lower the number, the more times the reply message will be searched for during the *Local/RemoteMsgTimeout* period. This relates effectively to remote administration — for local administration, there should only need to be a single attempt to retrieve the reply message.
5. *LocalMsgTimeout* and *RemoteMsgTimeout* indicate the length of time that it takes to check for a reply message. If a reply message is not returned within the specified time, then the inquire/update returns in an unknown state. For all local administration, a reply message should always be received. A case

when a reply message may not be received is if remote administration is taking place and the remote queue manager does not have a connection definition back to the originating queue manager. In this case setting the *RemoteMsgTimeout* value to zero may be useful as it is already known that a reply message will not be received. In every case where a reply message is not received an exception will always be thrown. Setting the *RemoteMsgTimeout* to zero does not change this.

6. *LocalMsgTimeout, RemoteMsgTimeout* and *MsgPollInterval* have defaults 10,000ms, 10,000ms and 10ms respectively. Resetting these values takes effect immediately and the new values are in force until they are reset again or the application is terminated.

Operations

There are no operations for this MBean

Queue manager

Attributes

Table 2. Queue Manager attributes

Attribute name	Attribute description	Attribute type	Read/Write
Name	The name of the resource	java.lang.String	RO
Description	An arbitrary string describing the resource	java.lang.String	RW
Aliases	Alternative names for the resource	[Ljava.lang.String;	RW
ChannelAttributeRules	The rule class (or alias) to be associated with the channel attribute	java.lang.String	RW
ChannelTimeout	The time (in milliseconds) after which an outgoing idle channel will be turned off	java.lang.Long	RW
Connections	A list of connections owned by this queue manager	[Ljava.lang.String;	RO
MQBridges	A list of bridges owned by this queue manager	[Ljava.lang.String;	RO
Queues	A list of queues owned by this queue manager	[Ljava.lang.String;	RO
Rule	The rule class (or alias) to be used by this queue manager	java.lang.String	RW
MaxTransThreads	The maximum number of threads that will be spawned to service the transmission needs of the queue manager	java.lang.Integer	RW
Version	The version of MQe hosting the queue manager	[Ljava.lang.String;	RO
CommsListeners	A list of communications listeners owned by this queue manager	[Ljava.lang.String;	RO
BridgeCapable	Indicates whether the queue manager is bridge capable	java.lang.Boolean	RO
MQeClass	The class of the resource	java.lang.String	RO
QMsgStore	The default message store class (or alias) for a queue that determines how messages on that queue are stored	java.lang.String	RW
QAdapter	The default storage adapter class (or alias) for a queue	java.lang.String	RW
QPath	The default path for messages to be stored for queues	java.lang.String	RW

Operations

Table 3. Queue Manager operations

Operation name	Operation description	Parameter [1]
addAlias [2]	Adds an alias to the queue manager	alias
removeAlias [2]	Removes an alias from the queue manager	alias
addAlias	Adds an array of aliases to the queue manager	aliases
removeAlias	Removes an array of aliases from the queue manager	aliases
createAdminQueue	Creates a new admin queue	queueName, messageStore(optional), adapter(optional), path(optional)
createApplicationQueue	Creates a new application queue	queueName, messageStore(optional), adapter(optional), path(optional)
createHomeServerQueue	Creates a new home server queue	queueName, getFromQMgrName
createAsyncProxyQueue	Creates a new asynchronous proxy queue	queueName, destinationQMgrName, messageStore(optional), adapter(optional), path(optional)
createSyncProxyQueue	Creates a new synchronous proxy queue	queueName, destinationQMgrName
createStoreQueue	Creates a new store queue	queueName, messageStore(optional), adapter(optional), path(optional)
createForwardQueue	Creates a new forward queue	queueName, forwardToQMgrName, messageStore(optional), adapter(optional), path(optional)
createCommsListener	Creates a new communications listener	listenerName, listenerAdapter, listenerPort
createAliasConnection	Creates a new Alias Connection	connectionName
createMQConnection	Creates a new MQ Connection	connectionName
createUdpipConnection	Creates a new Udpip Connection	connectionName, address, port
createTcpipHistoryConnection	Creates a new Tcpip History Connection	connectionName, address, port
createTcpipHttpConnection	Creates a new Tcpip Http Connection	connectionName, address, port
createTcpipLengthConnection	Creates a new Tcpip Length Connection	connectionName, address, port
createIndirectConnection	Creates a new Indirect Connection	connectionName, viaQMName
createMQBridgeQueue	Creates a new MQBridge queue	queueName, destinationQMgrName, bridgeName, proxyName, clientConnectionName
createMQBridge	Creates a new MQBridge	bridgeName
triggerTransmission	Initiate the triggering of any pending messages	

Note:

1. See the following table for more information on parameters
2. This operation is provided to allow compatibility with adapters which can not handle array parameters to operations. A similar operation has also been added for some queues and connections.

Operations parameters

Table 4. Queue manager operation parameters

Parameter name	Parameter type	Parameter description
adapter	java.lang.String	Class name for the adapter to use with the message store - optional
address	java.lang.String	IP address for a connection
alias	java.lang.String	Name of the queue manager alias
aliases	[Ljava.lang.String;	Names of the queue manager aliases
bridgeName	java.lang.String	Name of an MQ bridge
clientConnectionName	java.lang.String	Name of MQ client connection associated with an MQ bridge queue
connectionName	java.lang.String	Name of a connection
destinationQMgrName	java.lang.String	Name of the queue manager that owns a given proxy (remote) queue or a bridge queue
forwardToQMgrName	java.lang.String	Name of the queue manager that messages are forwarded to from a Forward queue
getFromQMgrName	java.lang.String	Name of the queue manager that owns a given home server queue
listenerAdapter	java.lang.String	Listener adapter class
listenerName	java.lang.String	Name of a listener
listenerPort	java.lang.String	Port for a listener to listen on
messageStore	java.lang.String	Class name for the message store optional
path	java.lang.String	Path for the queue store optional
port	java.lang.String	IP Port for a connection
proxyName	java.lang.String	Name of MQ queue manager proxy associated with an MQ bridge queue
queueName	java.lang.String	Name of the queue
viaQMName	java.lang.String	Name of a queue manager to connect via (for an indirect connection)

Note:

1. The return type in each case is of type `java.lang.Void`. Hence, return types have not been included in the table.
2. There may seem to be a discrepancy between the input parameters listed for the operations and the input parameters required for the corresponding MQe operations. This is because the interface design allows the user to input only mandatory parameters at this point. The reason for this is that where the adapter used provides a graphical interface, the inclusion of all optional parameters for each operation would result in a very cluttered interface. Thus, all optional parameters have been omitted in these create operations. Once the resource has been created, they can be specified as updates using `setAttribute()` or `setAttributes()`.
3. Some of these methods may seem unfamiliar to someone who uses the MQe programmatic interface. In particular the methods `createStoreQueue()`, and `createForwardQueue()` do not correspond to MQe standard APIs. The rationale behind these resources is explained in the relevant sections below on Store Queues, Forward Queues and Connections.

Remote queue manager

Attributes

Table 5. Remote queue manager attributes

Attribute name	Attribute description	Attribute type	Read/Write
Accessible	A Boolean value indicating whether the remote queue manager is accessible from the local queue manager or not	java.lang.Boolean	RO
Name	The name of the resource	java.lang.String	RO
Description	An arbitrary string describing the resource	java.lang.String	RW
Aliases	Alternative names for the resource	[Ljava.lang.String;	RW
ChannelAttributeRules	The rule class (or alias) to be associated with the channel attribute	java.lang.String	RW
ChannelTimeout	The time (in milliseconds) after which an outgoing idle channel will be turned off	java.lang.Long	RW
Connections	A list of connections owned by queue manager	[Ljava.lang.String;	RO
MQBridges	A list of Bridges owned by this queue manager	[Ljava.lang.String;	RO
Queues	A list of Queues owned by this queue manager	[Ljava.lang.String;	RO
Rule	The rule class (or alias) to be used by this queue manager	java.lang.String	RW
MaxTransThreads	The maximum number of threads that will be spawned to service the transmission needs of the queue manager	java.lang.Integer	RW
Version	The version of MQe hosting the queue manager	[Ljava.lang.Short;	RO
CommsListeners	A list of communications listeners owned by this queue manager	[Ljava.lang.String;	RO
BridgeCapable	Indicates whether the queue manager is bridge capable	java.lang.Boolean	RO
MQeClass	The class of the resource	java.lang.String	RO
QMsgStore	The default message store class (or alias) for a queue that determines how messages on that queue are stored	java.lang.String	RW
QAdapter	The default storage adapter class (or alias) for a queue	java.lang.String	RW
QPath	The default path for messages to be stored for queues	java.lang.String	RW

Operations

Table 6. Remote queue manager operations

Operation name	Operation description	Parameter [1]
addAlias [2]	Adds an alias to the queue manager	alias

Table 6. Remote queue manager operations (continued)

Operation name	Operation description	Parameter [1]
removeAlias [2]	Removes an alias from the queue manager	alias
addAlias	Adds an array of aliases to the queue manager	aliases
removeAlias	Removes an array of aliases from the queue manager	aliases
createAdminQueue	Creates a new admin queue	queueName, messageStore(optional), adapter(optional), path(optional)
createApplicationQueue	Creates a new application queue	queueName, messageStore(optional), adapter(optional), path(optional)
createHomeServerQueue	Creates a new home server queue.	queueName, getFromQMgrName
createAsyncProxyQueue	Creates a new asynchronous proxy queue	queueName, destinationQMgrName, messageStore(optional), adapter(optional), path(optional)
createSyncProxyQueue	Creates a new synchronous proxy queue	queueName, destinationQMgrName
createStoreQueue	Creates a new store queue	queueName, messageStore(optional), adapter(optional), path(optional)
createForwardQueue	Creates a new forward queue	queueName, forwardToQMgrName, messageStore(optional), adapter(optional), path(optional)
createCommsListener	Creates a new communications listener	listenerName, listenerAdapter, listenerPort
createAliasConnection	Creates a new Alias Connection	connectionName
createMQConnection	Creates a new MQ Connection	connectionName
createUdpipConnection	Creates a new Udpip Connection	connectionName, address, port
createTcpipHistoryConnection	Creates a new Tcpip History Connection	connectionName, address, port
createTcpipLengthConnection	Creates a new Tcpip Length Connection	connectionName, address, port
createTcpipHttpConnection	Creates a new Tcpip Http Connection	connectionName, address, port
createIndirectConnection	Creates a new Indirect Connection	connectionName, viaQMName
createMQBridgeQueue	Creates a new MQBridge queue	queueName, destinationQMgrName, bridgeName, proxyName, clientConnectionName
createMQBridge	Creates a new MQBridge	bridgeName
refresh	Refresh the queue manager resources from the registry	

Note:

1. See Table 7 on page 8 for more information on parameters.

2. This operation is provided to allow compatibility adapters which can not handle array parameters to operations. A similar operation has also been added for some queues and connections.

Operations parameters

Table 7. Remote queue manager operation parameters

Parameter name	Parameter type	Parameter description
adapter	java.lang.String	Class name for the adapter to use with the message store - optional
address	java.lang.String	IP address for a connection
alias	java.lang.String	Name of the queue manager alias
aliases	[Ljava.lang.String;	Names of the queue manager aliases
bridgeName	java.lang.String	Name of an MQ bridge
clientConnectionName	java.lang.String	Name of MQ client connection associated with an MQ bridge queue
connectionName	java.lang.String	Name of a connection
destinationQMgrName	java.lang.String	Name of the queue manager that owns a given proxy (remote) queue or a bridge queue
forwardToQMgrName	java.lang.String	Name of the queue manager that messages are forwarded to from a Forward queue
getFromQMgrName	java.lang.String	Name of the queue manager that owns a given home server queue
listenerAdapter	java.lang.String	Listener adapter class
listenerName	java.lang.String	Name of a listener
listenerPort	java.lang.String	Port for a listener to listen on
messageStore	java.lang.String	Class name for the message store optional
path	java.lang.String	Path for the queue store - optional
port	java.lang.String	IP Port for a connection
proxyName	java.lang.String	Name of MQ queue manager proxy associated with an MQ bridge queue
queueName	java.lang.String	Name of the queue
viaQMName	java.lang.String	Name of a queue manager to connect via (for an indirect connection)

Note:

1. The refresh() operation requires particular comment. When resources are added to or removed from a JMX-enabled queue manager, whether via the JMX interface or from another application, these updates are automatically reflected in the MBeans (that is to say, corresponding MBeans are registered or deregistered). However, when the MBeans corresponding to a queue manager which is remote to the JMX-enabled queue manager (known to it through a direct connection) are updated from another application, these changes are not reflected automatically. In this case, the refresh() operation has to be invoked to update the MBeans in accordance with the current remote queue manager resources.
2. When a direct connection to another Queue Manager is created, a RemoteQueueManager MBean is created in addition to the MBean for the connection definition itself. If the MQeAdminJmx attribute InquireOnConnect is set to true, MBeans for the remote queue manager child resources will be created and registered with the MBeanServer instance at this point. However, if InquireOnConnect is set to false, the child MBeans will not be created. The refresh operation on this MBean will need to be invoked at a later time in order to create the remote queue manager alias MBeans and the MBeans for the child resources when/if required. Note that MBeans for the remote Queue Manager child

resources are only created for queue managers connected to the JMX local queue manager by direct connections no child MBeans are created for queue managers known only through MQ, alias or indirect connections.

Admin queue

Attributes

Table 8. Admin queue attributes

Attribute name	Attribute description	Attribute type	Read/Write
Name	The name of the resource	java.lang.String	RO
Description	An arbitrary string describing the resource	java.lang.String	RW
Active	A Boolean value indicating whether the queue is active or not	java.lang.Boolean	RO
Adapter	The adapter class (or alias) to be used by the queue	java.lang.String	RW
Aliases	Alternative names for the resource	[Ljava.lang.String;	RW
AttributeRule	The attribute class (or alias) associated with the security attributes of the queue	java.lang.String	RW
Authenticator	The authenticator class (or alias) associated with the queue	java.lang.String	RW
Compressor	The compressor class (or alias) associated with the queue	java.lang.String	RW
Cryptor	The cryptor class (or alias) associated with the queue	java.lang.String	RW
CreationDate	The time (in milliseconds since midnight Jan1, 1970 GMT) the queue object was created	java.lang.Long	RO
CurrentDepth	The number of messages currently on the queue	java.lang.Integer	RO
Expiry	The time (in milliseconds) after which messages places on the queue expire.	java.lang.Long	RW
LocalQMgr	The name of the queue manager to own the resource	java.lang.String	RO
MaxDepth	The maximum number of messages that may be placed on the queue	java.lang.Integer	RW
MaxMessageSize	The maximum size of a message that can be placed on the queue	java.lang.Integer	RW
MessageStore	The class (or alias) determines how messages on the queue are stored	java.lang.String	RO
MQeClass	The class of the resource	java.lang.String	RO
Path	The path locating the physical storage for the queue	java.lang.String	RO
Priority	The default priority to be associated with messages on the queue	java.lang.Byte	RW
Rule	The rule class (or alias) to be used by the queue	java.lang.String	RW
TargetRegistry	The registry to be used by the authenticator	java.lang.Byte	RW
TimerInterval	The time (in milliseconds) between attempts to get messages	java.lang.Long	RW

Operations

Table 9. Admin queue operations

Operation name	Operation description	Parameter
addAlias	Adds an alias to this queue	alias
removeAlias	Removes an alias from this queue	alias
addAlias	Adds an array of aliases to this queue	aliases
removeAlias	Removes an array of aliases from this queue	aliases
delete	Deletes this queue	

Operations parameters

Table 10. Admin queue operations parameters

Parameter name	Parameter type	Parameter description
alias	java.lang.String	Name of the resource alias
aliases	[Ljava.lang.String;	Names of the resource aliases

Application queue

This queue type is also referred to *local queue* elsewhere in MQe documentation.

Attributes

Table 11. Application queue attributes

Attribute name	Attribute description	Attribute type	Read/Write
Name	The name of the resource	java.lang.String	RO
Description	An arbitrary string describing the resource	java.lang.String	RW
Active	A Boolean value indicating whether the queue is active or not	java.lang.Boolean	RO
Adapter	The adapter class (or alias) to be used by the queue	java.lang.String	RW
Aliases	Alternative names for the resource	[Ljava.lang.String;	RW
AttributeRule	The attribute class (or alias) associated with the security attributes of the queue	java.lang.String	RW
Authenticator	The authenticator class (or alias) associated with the queue	java.lang.String	RW
Compressor	The compressor class (or alias) associated with the queue	java.lang.String	RW
Cryptor	The cryptor class (or alias) associated with the queue	java.lang.String	RW
CreationDate	The time (in milliseconds since midnight Jan1, 1970 GMT) the queue object was created	java.lang.Long	RO
CurrentDepth	The number of messages currently on the queue	java.lang.Integer	RO
Expiry	The time (in milliseconds) after which messages places on the queue expire	java.lang.Long	RW
LocalQMgr	The name of the queue manager to own the resource	java.lang.String	RO

Table 11. Application queue attributes (continued)

Attribute name	Attribute description	Attribute type	Read/Write
MaxDepth	The maximum number of messages that may be placed on the queue	java.lang.Integer	RW
MaxMessageSize	The maximum size of a message that may be placed on the queue	java.lang.Integer	RW
MessageStore	The class (or alias) determines how messages on the queue are stored	java.lang.String	RO
Messages	The message bodies of messages on the queue	[Ljava.lang.String;	RO
MQeClass	The class of the resource	java.lang.String	RO
Path	The path locating the physical storage for the queue	java.lang.String	RO
Priority	The default priority to be associated with messages on the queue	java.lang.Byte	RW
Rule	The rule class (or alias) to be used by the queue	java.lang.String	RW
TargetRegistry	The registry to be used by the authenticator	java.lang.Byte	RW

Operations

Table 12. Application queue operations

Operation name	Operation description	Parameter
addAlias	Adds an alias to this queue	alias
removeAlias	Removes an alias from this queue	alias
addAlias	Adds an array of aliases to this queue	aliases
removeAlias	Removes an array of aliases from this queue	aliases
delete	Deletes this queue	
deleteMessage[1]	Deletes a message from this queue	index
putMessage[1]	Places a message onto the queue	message

Operations parameters

Table 13. Application queue operations parameters

Parameter name	Parameter type	Parameter description
index	java.lang.Integer	The index of the message to be deleted
message	java.lang.String	The text body of the message to be put
alias	java.lang.String	Name of the resource alias
aliases	[Ljava.lang.String;	Names of the resource aliases

Home Server queue

Attributes

Table 14. Home Server queue attributes

Attribute name	Attribute description	Attribute type	Read/Write
Name	The name of the resource	java.lang.String	RO

Table 14. Home Server queue attributes (continued)

Attribute name	Attribute description	Attribute type	Read/Write
Description	An arbitrary string describing the resource	java.lang.String	RW
Active	A Boolean value indicating whether the queue is active or not	java.lang.Boolean	RO
AttributeRule	The attribute class (or alias) associated with the security attributes of the queue	java.lang.String	RW
Authenticator	The authenticator class (or alias) associated with the queue	java.lang.String	RW
Compressor	The compressor class (or alias) associated with the queue	java.lang.String	RW
Cryptor	The cryptor class (or alias) associated with the queue	java.lang.String	RW
CreationDate	The time (in milliseconds since midnight Jan1, 1970 GMT) the queue object was created	java.lang.Long	RO
GetFromQMgr	The name of the queue manager that the home server queue will pull messages from	java.lang.String	RO
LocalQMgr	The name of the queue manager to own the resource	java.lang.String	RO
MQeClass	The class of the resource	java.lang.String	RO
Rule	The rule class (or alias) to be used by the queue	java.lang.String	RW
TargetRegistry	The registry to be used by the authenticator	java.lang.Byte	RW
TimerInterval	The time (in milliseconds) between attempts to get messages	java.lang.Long	RW
Transporter	The class (or alias) that flows messages over the channel to the target queue	java.lang.String	RW

Operations

Table 15. Home Server queue operations

Operation name	Operation description	Parameter
delete	Deletes this queue	

Asynchronous Proxy queue

This queue type is also referred to as *asynchronous remote queue* elsewhere in MQe documentation.

Attributes

Table 16. Asynchronous Proxy queue attributes

Attribute name	Attribute description	Attribute type	Read/Write
Name	The name of the resource	java.lang.String	RO
Description	An arbitrary string describing the resource	java.lang.String	RW
Active	A Boolean value indicating whether the queue is active or not	java.lang.Boolean	RO
Adapter	The adapter class (or alias) to be used by the queue	java.lang.String	RW
Aliases	Alternative names for the resource	[Ljava.lang.String;	RW

Table 16. Asynchronous Proxy queue attributes (continued)

Attribute name	Attribute description	Attribute type	Read/Write
AttributeRule	The attribute class (or alias) associated with the security attributes of the queue	java.lang.String	RW
Authenticator	The authenticator class (or alias) associated with the queue	java.lang.String	RW
Compressor	The compressor class (or alias) associated with the queue	java.lang.String	RW
Cryptor	The cryptor class (or alias) associated with the queue	java.lang.String	RW
CreationDate	The time (in milliseconds since midnight Jan1, 1970 GMT) the queue object was created	java.lang.Long	RO
CurrentDepth	The number of messages currently on the queue	java.lang.Integer	RO
Expiry	The time (in milliseconds) after which messages places on the queue expire	java.lang.Long	RW
DestinationQMgr	The name of the queue manager to own the physical queue	java.lang.String	RO
LocalQMgr	The name of the queue manager to own the resource	java.lang.String	RO
MaxDepth	The maximum number of messages that may be placed on the queue	java.lang.Integer	RW
MaxMessageSize	The maximum size of a message that may be placed on the queue	java.lang.Integer	RW
MessageStore	The class (or alias) determines how messages on the queue are stored	java.lang.String	RO
MQeClass	The class of the resource	java.lang.String	RO
Path	The path locating the physical storage for the queue	java.lang.String	RO
Priority	The default priority to be associated with messages on the queue	java.lang.Byte	RW
Rule	The rule class (or alias) to be used by the queue	java.lang.String	RW
TargetRegistry	The registry to be used by the authenticator	java.lang.Byte	RW
Transporter	The class (or alias) that flows messages over the channel to the target queue	java.lang.String	RW

Operations

Table 17. Asynchronous Proxy queue operations

Operation name	Operation description	Parameter
addAlias	Adds an alias to this queue	alias
removeAlias	Removes an alias from this queue	alias
addAlias	Adds an array of aliases to this queue	aliases
removeAlias	Removes an array of aliases from this queue	aliases
delete	Deletes this queue	
putMessage	Places a message onto the queue	message

Operations parameters

Table 18. Asynchronous Proxy queue operations parameters

Parameter name	Parameter type	Parameter description
message	java.lang.String	The text body of the message to be put
alias	java.lang.String	Name of the resource alias
aliases	[Ljava.lang.String;	Names of the resource aliases

Synchronous Proxy queue

This queue type is also referred to as *synchronous remote queue* elsewhere in MQe documentation.

Attributes

Table 19. Synchronous Proxy queue attributes

Attribute name	Attribute description	Attribute type	Read/Write
Name	The name of the resource	java.lang.String	RO
Description	An arbitrary string describing the resource	java.lang.String	RW
Active	A Boolean value indicating whether the queue is active or not	java.lang.Boolean	RO
Aliases	Alternative names for the resource	[Ljava.lang.String;	RW
AttributeRule	The attribute class (or alias) associated with the security attributes of the queue	java.lang.String	RW
Authenticator	The authenticator class (or alias) associated with the queue	java.lang.String	RW
Compressor	The compressor class (or alias) associated with the queue	java.lang.String	RW
Cryptor	The cryptor class (or alias) associated with the queue	java.lang.String	RW
CreationDate	The time (in milliseconds since midnight Jan1, 1970 GMT) the queue object was created	java.lang.Long	RO
DestinationQMgr	The name of the queue manager to own the physical queue	java.lang.String	RO
LocalQMgr	The name of the queue manager to own the resource	java.lang.String	RO
MaxMessageSize	The maximum size of a message that may be placed on the queue	java.lang.Integer	RW
MQeClass	The class of the resource	java.lang.String	RO
Rule	The rule class (or alias) to be used by the queue	java.lang.String	RW
TargetRegistry	The registry to be used by the authenticator	java.lang.Byte	RW
Transporter	The class (or alias) that flows messages over the channel to the target queue	java.lang.String	RW

Operations

Table 20. Synchronous Proxy queue operations

Operation name	Operation description	Parameter
addAlias	Adds an alias to this queue	alias

Table 20. Synchronous Proxy queue operations (continued)

Operation name	Operation description	Parameter
removeAlias	Removes an alias from this queue	alias
addAlias	Adds an array of aliases to this queue	aliases
removeAlias	Removes an array of aliases from this queue	aliases
delete	Deletes this queue	
putMessage	Places a message onto the queue	message

Operations parameters

Table 21. Synchronous Proxy queue operations parameters

Parameter name	Parameter type	Parameter description
message	java.lang.String	The text body of the message to be put
alias	java.lang.String	Name of the resource alias
aliases	[Ljava.lang.String;	Names of the resource aliases

Store queue

These two queue types (`store` and `forward`) require some explanation.

An MQe JMX `store queue` MBean maps onto an MQe queue of type `MQeStoreAndForwardQueue` but with the functionality of that queue somewhat curtailed for ease of use:

- An `MQeStoreAndForwardQueue` has the ability to store messages for a list of target queue managers (`DestinationQMgrs`) and also has the ability to forward messages to one specified `ForwardToQMgr`.
- However, the MQe JMX implementation has split this dual messaging functionality into two, so that our `store queues` retain the ability to **store** messages for a list of target queue managers, but do not have a `ForwardToQMgr`.
- The **forwarding** functionality of the `MQeStoreAndForwardQueue` is retained in our `forward queue` MBean.

Attributes

Table 22. Store queue attributes

Attribute name	Attribute description	Attribute type	Read/Write
Name	The name of the resource.	java.lang.String	RO
Description	An arbitrary string describing the resource	java.lang.String	RW
Active	A Boolean value indicating whether the queue is active or not	java.lang.Boolean	RO
Adapter	The adapter class (or alias) to be used by the queue	java.lang.String	RW
AttributeRule	The attribute class (or alias) associated with the security attributes of the queue	java.lang.String	RW
Authenticator	The authenticator class (or alias) associated with the queue	java.lang.String	RW
Compressor	The compressor class (or alias) associated with the queue	java.lang.String	RW
Cryptor	The cryptor class (or alias) associated with the queue	java.lang.String	RW
CreationDate	The time (in milliseconds since midnight Jan1, 1970 GMT) the queue object was created	java.lang.Long	RO

Table 22. Store queue attributes (continued)

Attribute name	Attribute description	Attribute type	Read/Write
CurrentDepth	The number of messages currently on the queue	java.lang.Integer	RO
DestinationQMgrs	The queue manager destinations for which a store (or forward) queue will hold messages	[Ljava.lang.String;	RW
Expiry	The time (in milliseconds) after which messages places on the queue expire	java.lang.Long	RW
LocalQMgr	The name of the queue manager to own the resource	java.lang.String	RO
MaxDepth	The maximum number of messages that may be placed on the queue	java.lang.Integer	RW
MaxMessageSize	The maximum size of message that may be placed on the queue	java.lang.Integer	RW
MessageStore	The class (or alias) determines how messages on the queue are stored	java.lang.String	RO
MQeClass	The class of the resource	java.lang.String	RO
Path	The path locating the physical storage for the queue	java.lang.String	RO
Priority	The default priority to be associated with messages on the queue	java.lang.Byte	RW
Rule	The rule class (or alias) to be used by the queue	java.lang.String	RW
TargetRegistry	The registry to be used by the authenticator	java.lang.Byte	RW
Transporter	The class (or alias) that flows messages over the channel to the target queue	java.lang.String	RW

Operations

Table 23. Store queue operations

Operation name	Operation description	Parameter
addDestinationQMgr [1]	Adds a destinationQMgr to the queues DestinationQMgrs List	DestinationQMgr
removeDestinationQMgr [1]	Removes a destinationQMgr from the queues DestinationQMgrs List	DestinationQMgr
addDestinationQMgrs	Adds an array of destinationQMgrs to the queues DestinationQMgrs List	DestinationQMgrs
removeDestinationQMgrs	Removes an array of destinationQMgrs from the queues DestinationQMgrs List	DestinationQMgrs
delete	Deletes this queue	

Note:

1. This operation is provided to allow compatibility with adapters which cannot handle array parameters to operations.

Operations parameters

Table 24. Store queue operations parameters

Parameter name	Parameter type	Parameter Description
DestinationQMgr	java.lang.String	Destination queue manager name to be added or removed
DestinationQMgrs	[Ljava.lang.String;	Destination queue manager names to be added or removed

Forward queue

These two queue types (`store` and `forward`) require some explanation.

An MQe JMX `store queue` MBean maps onto an MQe queue of type `MQeStoreAndForwardQueue` but with the functionality of that queue somewhat curtailed for ease of use:

- An `MQeStoreAndForwardQueue` has the ability to store messages for a list of target queue managers (`DestinationQMgrs`) and also has the ability to forward messages to one specified `ForwardToQMgr`.
- However, the MQe JMX implementation has split this dual messaging functionality into two, so that our `store queues` retain the ability to **store** messages for a list of target queue managers, but do not have a `ForwardToQMgr`.
- The **forwarding** functionality of the `MQeStoreAndForwardQueue` is retained in our `forward queue` MBean.

Attributes

Table 25. Forward queue attributes

Attribute name	Attribute description	Attribute type	Read/Write
Name	The name of the resource	java.lang.String	RO
Description	An arbitrary string describing the resource	java.lang.String	RW
Active	A Boolean value indicating whether the queue is active or not	java.lang.Boolean	RO
Adapter	The adapter class (or alias) to be used by the queue	java.lang.String	RW
AttributeRule	The attribute class (or alias) associated with the security attributes of the queue	java.lang.String	RW
Authenticator	The authenticator class (or alias) associated with the queue	java.lang.String	RW
Compressor	The compressor class (or alias) associated with the queue	java.lang.String	RW
Cryptor	The cryptor class (or alias) associated with the queue	java.lang.String	RW
CreationDate	The time (in milliseconds since midnight Jan1, 1970 GMT) the queue object was created	java.lang.Long	RO
CurrentDepth	The number of messages currently on the queue	java.lang.Integer	RO
DestinationQMgrs	The queue manager destinations for which a forward (or store) queue will hold messages	[Ljava.lang.String;	RW
Expiry	The time (in milliseconds) after which messages places on the queue expire	java.lang.Long	RW
ForwardToQMgr	The name of the next queue manager that will receive the messages for a forward queue	java.lang.String	RO
LocalQMgr	The name of the queue manager to own the resource	java.lang.String	RO

Table 25. Forward queue attributes (continued)

Attribute name	Attribute description	Attribute type	Read/Write
MaxDepth	The maximum number of messages that may be placed on the queue	java.lang.Integer	RW
MaxMessageSize	The maximum size of a message that may be placed on the queue	java.lang.Integer	RW
MessageStore	The class (or alias) determines how messages on the queue are stored	java.lang.String	RO
MQeClass	The class of the resource	java.lang.String	RO
Path	The path locating the physical storage for the queue	java.lang.String	RO
Priority	The default priority to be associated with messages on the queue	java.lang.Byte	RW
Rule	The rule class (or alias) to be used by the queue	java.lang.String	RW
TargetRegistry	The registry to be used by the authenticator	java.lang.Byte	RW
Transporter	The class (or alias) that flows messages over the channel to the target queue	java.lang.String	RW

Operations

Table 26. Forward queue operations

Operation name	Operation description	Parameter
addDestinationQMgr	Adds a destinationQMgr to the queues DestinationQMgrs List	DestinationQMgr
removeDestinationQMgr	Removes a destinationQMgr from the queues DestinationQMgrs List	DestinationQMgr
addDestinationQMgrs	Adds an array of destinationQMgrs to the queues DestinationQMgrs List	DestinationQMgrs
removeDestinationQMgrs	Removes an array of destinationQMgrs from the queues DestinationQMgrs List	DestinationQMgrs
delete	Deletes this queue	

Operations parameters

Table 27. Forward queue operations parameters

Parameter name	Parameter type	Parameter Description
DestinationQMgr	java.lang.String	Destination queue manager name to be added or removed
DestinationQMgrs	[Ljava.lang.String;	Destination queue manager names to be added or removed

Communications Listener

Attributes

Table 28. Communications Listener attributes

Attribute name	Attribute description	Attribute type	Read/Write
Name	The name of the resource	java.lang.String	RO
Description	An arbitrary string describing the resource	java.lang.String	RW

Table 28. Communications Listener attributes (continued)

Attribute name	Attribute description	Attribute type	Read/Write
Running	A Boolean value indicating if the listener is running	java.lang.Boolean	RO
Adapter	The class (or alias) of the communications protocol adapter	java.lang.String	RO
ChannelTimeout	The time (in milliseconds) after which an idle incoming connection will be timed out	java.lang.Long	RW
CurrentChannels	The number of channels currently open on the communications listener	java.lang.Integer	RO
MaxChannels	The maximum number of channels allowed for the communications listener	java.lang.Integer	RW
LocalQMgr	The name of the queue manager to own the resource	java.lang.String	RO
MQeClass	The class of the resource	java.lang.String	RO
Port	The IP port number used by the communications listener to service incoming connection requests	java.lang.String	RO

Operations

Table 29. Communications Listener operations

Operation name	Operation description	Parameter
stop	Starts this listener	
start	Stops this listener	
delete	Deletes this listener	

MQ/Alias connection

- MQ connections are used to define MQ queue managers. The only parameter needed to create one is the connection definition name.
- Alias connections are used as another way to add aliases to a local queue manager. The only parameter that is needed to create one is the connection definition name.

Both these connections may also be known as no-op connections.

Although there are two separate MQeQueueManagerJmx methods for creating MQ and Alias connections, both types of connection share a domain name:
`com.ibm.WMQe_`*`<OwningQMName>`*`_MQConnections:name=`*`<ConnectionName>`*.

This is because they are identical in practice.

Attributes

Table 30. MQ/Alias Connection attributes

Attribute name	Attribute description	Attribute type	Read/Write
Name	The name of the resource	java.lang.String	RO
Description	An arbitrary string describing the resource	java.lang.String	RW
Aliases	Alternative names for the resource	[Ljava.lang.String;	RW

Table 30. MQ/Alias Connection attributes (continued)

Attribute name	Attribute description	Attribute type	Read/Write
LocalQMgr	The name of the queue manager to own the resource	java.lang.String	RO

Operations

Table 31. MQ/Alias Connection operations

Operation name	Operation description	Parameter
addAlias	Adds an alias to this resource	alias
removeAlias	Removes an alias from this resource	alias
addAlias	Adds an array of aliases to this resource	aliases
removeAlias	Removes an array of aliases from this resource	aliases
delete	Deletes this resource	

Operations parameters

Table 32. MQ/Alias connection operations parameters

Parameter name	Parameter type	Parameter description
alias	java.lang.String	Name of the resource alias
aliases	[Ljava.lang.String;	Names of the resource aliases

Direct connection

To create a direct connection, the parameters *adapter*, *port* and *address* are all valid.

- The *port* and *address* values are required.
- The *adapter* is assigned a default value according to the type of direct connection created.

There are several types of connections which fall under this category and which share the same attributes and operations. These are currently:

- Udpip connection, TcpipLength connection, TcpipHttp connection, TcpipHistory connection.
- An instance of each type of connection and its corresponding MBean is created using a type-specific API in the QueueManager MBean (for example `createUdpipConnection()`).

Once created, the type of connection can be distinguished by the value of the Adapter attribute.

For the sake of convenience, these connection types are grouped together in this section under the heading DirectConnection.

Attributes

Table 33. Direct Connection attributes

Attribute name	Attribute description	Attribute type	Read/Write
Name	The name of the resource	java.lang.String	RO
Description	An arbitrary string describing the resource	java.lang.String	RW
Aliases	Alternative names for the resource	[Ljava.lang.String;	RW
LocalQMgr	The name of the queue manager to own the resource	java.lang.String	RO

Table 33. Direct Connection attributes (continued)

Attribute name	Attribute description	Attribute type	Read/Write
Adapter	The class (or alias) of the communications protocol adapter	java.lang.String	RW
Address	The numeric or string IP address of the machine hosting the remote queue manager	java.lang.String	RW
Channel	The channel class (or alias) to be used in the connection	java.lang.String	RW
Persist [1]	Whether the adapter should be persistent or not	java.lang.Boolean	RW
Port	The IP port number used by the remote queue manager to service incoming requests	java.lang.String	RW
Servlet	Servlet options	java.lang.String	RW

Note:

1. To avoid confusion about how the attribute `Persist` relates to the options
   - `MQeCommunicationsAdapter.MQe_Adapter_PERSIST`
   - `MQeCommunicationsAdapter.MQe_Adapter_NOPERSIST`

   use the following equivalences:
   - Setting `Persist` to `true` is equivalent to setting `MQeCommunicationsAdapter.MQe_Adapter_PERSIST` to `true`.
   - Setting `Persist` to `false` is equivalent to setting `MQeCommunicationsAdapter.MQe_Adapter_NOPERSIST` to `true`.

Operations

Table 34. Direct Connection operations

Operation name	Operation description	Parameter
addAlias	Adds an alias to this resource	alias
removeAlias	Removes an alias from this resource	alias
addAlias	Adds an array of aliases to this resource	aliases
removeAlias	Removes an array of aliases from this resource	aliases
delete	Deletes this resource	

Operations parameters

Table 35. Direct connection operations parameters

Parameter name	Parameter type	Parameter description
alias	java.lang.String	Name of the resource alias
aliases	[Ljava.lang.String;	Names of the resource aliases

Indirect connection

These are connections that use an intermediate queue manager to get to the final destination queue manger.

Indirect connections require that the `viaQMName` parameter is set to the name of the intermediate queue manager.

The only parameters for creating a connection of this type are the `connectionName` and the `viaQMName`.

Attributes

Table 36. Indirect Connection attributes

Attribute name	Attribute description	Attribute type	Read/Write
Name	The name of the resource.	java.lang.String	RO
Description	An arbitrary string describing the resource.	java.lang.String	RW
Aliases	Alternative names for the resource.	[Ljava.lang.String;	RW
LocalQMgr	The name of the queue manager to own the resource	java.lang.String	RO
ViaQMName	The name of the queue manager to be used as the ViaQM	java.lang.String	RW

Operations

Table 37. Indirect Connection operations

Operation name	Operation description	Parameter
addAlias	Adds an alias to this resource	alias
removeAlias	Removes an alias from this resource	alias
addAlias	Adds an array of aliases to this resource	aliases
removeAlias	Removes an array of aliases from this resource	aliases
delete	Deletes this resource	

Operations parameters

Table 38. Indorect connection operations parameters

Parameter name	Parameter type	Parameter description
alias	java.lang.String	Name of the resource alias
aliases	[Ljava.lang.String;	Names of the resource aliases

MQ Bridge queue

Attributes

Table 39. MQ Bridge queue attributes

Attribute name	Attribute description	Attribute type	Read/Write
Name	The name of the resource	java.lang.String	RO
Description	An arbitrary string describing the resource	java.lang.String	RW
Active	A Boolean value indicating whether the queue is active or not	java.lang.Boolean	RO
Aliases	Alternative names for the resource	[Ljava.lang.String;	RW
BridgeName	The bridge object that handles the target MQ queue	java.lang.String	RW

Table 39. MQ Bridge queue attributes (continued)

Attribute name	Attribute description	Attribute type	Read/Write
ClientConnection	The name of the client connection associated with the queue	java.lang.String	RW
CreationDate	The time (in milliseconds since midnight Jan1, 1970 GMT) the queue object was created	java.lang.Long	RO
Expiry	The time (in milliseconds) after which messages places on the queue expire	java.lang.Long	RW
DestinationQMgr	The name of the queue manager to own the physical queue	java.lang.String	RO
LocalQMgr	The name of the queue manager to own the resource	java.lang.String	RO
MaxIdleTime	The maximum time (in seconds) that the MQ bridge queue can hold onto an idle connection before it is returned to the connection pool	java.lang.Integer	RW
MaxMessageSize	The maximum size of a message that may be placed on the queue	java.lang.Integer	RW
MQQueueManagerProxy	The target MQ QueueManager associated with the queue	java.lang.String	RW
MQRemoteQueueName	The actual queue name of the remote MQ queue	java.lang.String	RW
MQeClass	The class of the resource	java.lang.String	RO
Rule	The rule class (or alias) to be used by the queue	java.lang.String	RW
Transformer	The transformer class (or alias) converting the message from MQe to MQ format	java.lang.String	RW

Operations

Table 40. MQ Bridge queue operations

Operation name	Operation description	Parameter
addAlias	Adds an alias to this queue	alias
removeAlias	Removes an alias from this queue	alias
addAlias	Adds an array of aliases to this resource	aliases
removeAlias	Removes an array of aliases from this resource	aliases
delete	Deletes this queue	
putMessage	Places a message onto the queue	message

Operations parameters

Table 41. MQ Bridge queue operations parameters

Parameter name	Parameter type	Parameter description
message	java.lang.String	The text body of the message to be put

Table 41. MQ Bridge queue operations parameters (continued)

Parameter name	Parameter type	Parameter description
alias	java.lang.String	Name of the resource alias
aliases	[Ljava.lang.String;	Names of the resource aliases

MQ Bridge

A bridge resource is part of a hierarchy which takes the following form:

- an `MQ Bridge` instance can have one or more `MQ QueueManager Proxy` children.
  - these can have `MQ Client Connection` children
    - which can have `MQ Listener` children

Note: Some bridge resources, when their attributes are changed, will only reflect these changes when the resource has been stopped: for example, this is the case with the `SyncQName` attribute of the `MQ Client Connection`. For more information on this subject, see the Java Programming Reference.

Attributes

Table 42. MQ Bridge attributes

Attribute name	Attribute description	Attribute type	Read/Write
Name	The name of the resource	java.lang.String	RO
Description	An arbitrary string describing the resource	java.lang.String	RW
AdministeredObjectClass	The class name (or alias) used to realize the resource	java.lang.String	RO
Children	The list of child objects	[Ljava.lang.String;	RO
DefaultTransformer	The default transformer class (or alias) used for message conversion	java.lang.String	RW
HeartBeatInterval	The heartbeat pulse interval in minutes	java.lang.Integer	RW
LocalQMgr	The name of the queue manager to own the resource	java.lang.String	RO
RunState	An integer value representing the running state of the resource	java.lang.Integer	RO
StartupRuleClass	The class name (or alias) of the rule used to start the resource	java.lang.String	RW

Operations

Table 43. MQ Bridge operations

Operation name	Operation description	Parameter
start	Starts this MQBridge	affectChildren
stop	Stops this MQBridge	affectChildren
delete	Deletes this MQBridge	affectChildren
createMQQMgrProxy	Creates a new MQ QueueManager proxy	proxyName

Operations parameters

Table 44. MQ Bridge operations parameters

Parameter name	Parameter type	Parameter description
affectChildren	java.lang.Boolean	A Boolean value indicating whether actions on this resource should affect child objects. [1]
proxyName	java.lang.String	Name of the MQ queue manager proxy

Note:

1. Some adaptors may have defaults. These defaults may differ from the MQe defaults. For example the Sun RI HtmlAdaptorServer defaults all boolean values to true.

MQ Queue Manager Proxy

Attributes

Table 45. MQ Queue Manager Proxy attributes

Attribute name	Attribute description	Attribute type	Read/Write
Name	The name of the resource	java.lang.String	RO
Description	An arbitrary string describing the resource	java.lang.String	RW
AdministeredObjectClass	The class name (or alias) used to realize the resource	java.lang.String	RO
BridgeName	Identifies the name of the bridge	java.lang.String	RO
Children	The list of child objects	[Ljava.lang.String;	RO
HostName	The IP address of the target MQ queue manager	java.lang.String	RW
LocalQMgr	The name of the queue manager to own the resource	java.lang.String	RO
RunState	An integer value representing the running state of the resource	java.lang.Integer	RO
StartupRuleClass	The class name (or alias) of the rule used to start the resource	java.lang.String	RW

Operations

Table 46. MQ Queue Manager Proxy operations

Operation name	Operation description	Parameter
start	Starts this MQ Queue Manager Proxy	affectChildren
stop	Stops this MQ Queue Manager Proxy	affectChildren
delete	Deletes this MQ Queue Manager Proxy	affectChildren
createClientConnection	Creates a new MQ Client Connection	clientConnectionName

Operations parameters

Table 47. MQ Queue Manager Proxy operations parameters

Parameter name	Parameter type	Parameter description
affectChildren	java.lang.Boolean	A Boolean value indicating whether actions on this resource should affect child objects

Table 47. MQ Queue Manager Proxy operations parameters (continued)

Parameter name	Parameter type	Parameter description
clientConnectionName	java.lang.String	Name of the MQ client connection

MQ Client Connection

Attributes

Table 48. MQ Client Connection attributes

Attribute name	Attribute description	Attribute type	Read/Write
Name	The name of the resource	java.lang.String	RO
Description	An arbitrary string describing the resource	java.lang.String	RW
AdapterClass	The bridge adapter class (or alias) used to move messages from MQe to the target MQ queue	java.lang.String	RW
AdministeredObjectClass	The class name (or alias) used to realize the resource	java.lang.String	RO
BridgeName	Identifies the name of the bridge	java.lang.String	RO
CCSID	The CCSID property used by MQ	java.lang.Integer	RW
Children	The list of child objects	[Ljava.lang.String;	RO
LocalQMgr	The name of the queue manager to own the resource	java.lang.String	RO
MQPassword	The password used with the MQUserID	java.lang.String	RW
MQQMgrProxyName	Identifies the name of the MQ Proxy	java.lang.String	RO
MQUserID	The user ID used by MQ	java.lang.String	RW
MaxConnectionIdleTime	The time (in minutes) after which an idle connection to MQ is discarded and the resources returned to the pool	java.lang.Integer	RW
Port	The IP port number used by the target MQ queue manager	java.lang.String	RW
ReceiveExit	The receive exit specified at the remote end of the MQ client channel	java.lang.String	RW
RunState	An integer value representing the running state of the resource	java.lang.Integer	RO
SecurityExit	The security exit specified at the remote end of the MQ client channel	java.lang.String	RW
SendExit	The send exit specified at the remote end of the MQ client channel	java.lang.String	RW
StartupRuleClass	The class (or alias) of the rule used to start the resource	java.lang.String	RW

Table 48. MQ Client Connection attributes (continued)

Attribute name	Attribute description	Attribute type	Read/Write
SyncQName	The name of the synchronization queue on the MQ queue manager used by the MQBridge	java.lang.String	RW
SyncQPurgeInterval	The time (in minutes) between successive purges of the sync queue	java.lang.Integer	RW
SyncQPurgerRulesClass	The rule class (or alias) used when a message on the sync queue indicates a failure of MQ to confim a message	java.lang.String	RW

Operations

Table 49. MQ Client Connection operations

Operation name	Operation description	Parameter
start	Starts this Client Connection	affectChildren
stop	Stops this Client Connection	affectChildren
delete	Deletes this Client Connection	affectChildren
createListener	Creates a new MQ listener	listenerName

Operations parameters

Parameter name	Parameter type	Parameter description
affectChildren	java.lang.Boolean	A Boolean value indicating whether actions on this resource should affect child objects
listenerName	java.lang.String	Name of the MQ listener

MQ Listener

Attributes

Table 50. MQ Listener attributes

Attribute name	Attribute description	Attribute type	Read/Write
Name	The name of the resource	java.lang.String	RO
Description	An arbitrary string describing the resource	java.lang.String	RW
AdministeredObjectClass	The class name (or alias) used to realize the resource	java.lang.String	RO
BridgeName	Identifies the name of the bridge	java.lang.String	RO
ClientConnectionName	Identifies the name of the Client Connection	java.lang.String	RO
DeadLetterQName	The MQ queue used to hold messages that cannot be delivered from MQ to MQe	java.lang.String	RW
FlowsPerCommit	The number of messages flowed after which the MQ sync queue (if used) is cleared	java.lang.Integer	RW

Table 50. MQ Listener attributes (continued)

Attribute name	Attribute description	Attribute type	Read/Write
ListenerStateStoreAdapter	The specification of permanent storage used to hold state information as messages are moved from MQ to MQe	java.lang.String	RW
LocalQMgr	The name of the queue manager to own the resource	java.lang.String	RO
MQQMgrProxyName	Identifies the name of the MQ Proxy	java.lang.String	RO
RunState	An integer value representing the running state of the resource	java.lang.Integer	RO
StartupRuleClass	The class (or alias) of the rule used to start the resource	java.lang.String	RW
TransformerClass	The class (or alias) of the actual transformer used by the MQBridge	java.lang.String	RW
UndeliveredMessageRuleClass	The rule class (or alias) determining the action to be taken when a message cannot be delivered from MQ to MQe	java.lang.String	RW

Operations

Table 51. MQ Listener operations

Operation name	Operation description	Parameter
start	Starts this MQ Listener	
stop	Stops this MQ Listener	
delete	Deletes this MQ Listener	