

**Name** create-jmsdest – creates a JMS physical destination

**Synopsis** create-jmsdest [--help]  
--desttype *dest\_type*  
[--property (*name=value*)[:*name=value*]\*]  
[--target *target*]  
*dest\_name*

**Description** The create-jmsdest subcommand creates a Java Message Service (JMS) physical destination. Typically, you use the create-jms-resource subcommand to create a JMS destination resource that has a Name property that specifies the physical destination. The physical destination is created automatically when you run an application that uses the destination resource. Use the create-jmsdest subcommand if you want to create a physical destination with non-default property settings.

This subcommand is supported in remote mode only. Remote asadmin subcommands require a running domain administration server (DAS).

**Options** --help  
-?

Displays the help text for the subcommand.

--desttype

The type of the JMS destination. Valid values are topic and queue.

--property

Optional attribute name/value pairs for configuring the physical destination. You can specify the following properties for a physical destination.

maxNumMsgs

The maximum number of unconsumed messages permitted for the destination. A value of -1 denotes an unlimited number of messages. The default value is -1. For the dead message queue, the default value is 1000.

If the limitBehavior property is set to FLOW\_CONTROL, it is possible for the specified message limit to be exceeded because the broker cannot react quickly enough to stop the flow of incoming messages. In such cases, the value specified for maxNumMsgs serves as merely a hint for the broker rather than a strictly enforced limit.

maxBytesPerMsg

The maximum size, in bytes, of any single message. Rejection of a persistent message is reported to the producing client with an exception; no notification is sent for non-persistent messages.

The value may be expressed in bytes, kilobytes, or megabytes, using the following suffixes:

b Bytes

k Kilobytes (1024 bytes)

m     Megabytes (1024 x 1024 = 1,048,576 bytes)

A value with no suffix is expressed in bytes; a value of -1 denotes an unlimited message size. The default value is -1.

#### maxTotalMsgBytes

The maximum total memory, in bytes, for unconsumed messages. The default value is -1. The syntax is the same as for `maxBytesPerMsg`. For the dead message queue, the default value is 10m.

#### limitBehavior

The behavior of the message queue broker when the memory-limit threshold is reached. Valid values are as follows.

##### REJECT\_NEWEST

Reject newest messages and notify the producing client with an exception only if the message is persistent. This is the default value.

##### FLOW\_CONTROL

Slow the rate at which message producers send messages.

##### REMOVE\_OLDEST

Throw out the oldest messages.

##### REMOVE\_LOW\_PRIORITY

Throw out the lowest-priority messages according to age, with no notification to the producing client.

If the value is `REMOVE_OLDEST` or `REMOVE_LOW_PRIORITY` and the `useDMQ` property is set to `true`, excess messages are moved to the dead message queue. For the dead message queue itself, the default limit behavior is `REMOVE_OLDEST`, and the value cannot be set to `FLOW_CONTROL`.

#### maxNumProducers

The maximum number of message producers for the destination. When this limit is reached, no new producers can be created. A value of -1 denotes an unlimited number of producers. The default value is 100. This property does not apply to the dead message queue.

#### consumerFlowLimit

The maximum number of messages that can be delivered to a consumer in a single batch. A value of -1 denotes an unlimited number of messages. The default value is 1000. The client runtime can override this limit by specifying a lower value on the connection factory object.

In load-balanced queue delivery, this is the initial number of queued messages routed to active consumers before load balancing begins.

**useDMQ**

If set to `true`, dead messages go to the dead message queue. If set to `false`, dead messages are discarded. The default value is `true`.

**validateXMLSchemaEnabled**

If set to `true`, XML schema validation is enabled for the destination. The default value is `false`.

When XML validation is enabled, the Message Queue client runtime will attempt to validate an XML message against the specified XSDs (or against the DTD, if no XSD is specified) before sending it to the broker. If the specified schema cannot be located or the message cannot be validated, the message is not sent, and an exception is thrown.

This property should be set when a destination is inactive: that is, when it has no consumers or producers and when there are no messages in the destination. Otherwise the producer must reconnect.

**XMLSchemaURIList**

A space-separated list of XML schema document (XSD) URI strings. The URIs point to the location of one or more XSDs to use for XML schema validation, if `validateXMLSchemaEnabled` is set to `true`. The default value is `null`.

Use double quotes around this value if multiple URIs are specified, as in the following example:

```
"http://foo/flap.xsd http://test.com/test.xsd"
```

If this property is not set or `null` and XML validation is enabled, XML validation is performed using a DTD specified in the XML document. If an XSD is changed as a result of changing application requirements, all client applications that produce XML messages based on the changed XSD must reconnect to the broker.

To modify the value of these properties, you can use the *as-install/mq/bin/imqcmd* command. See “Physical Destination Property Reference” in *Sun GlassFish Message Queue 4.4 Administration Guide* for more information.

**--target**

Creates the physical destination only for the specified target. Although the `create-jmsdest` subcommand is related to resources, a physical destination is created using the JMS Service (JMS Broker), which is part of the configuration. A JMS Broker is configured in the `config` section of `domain.xml`. Valid values are as follows:

**server**

Creates the physical destination for the default server instance. This is the default value.

***configuration-name***

Creates the physical destination in the specified configuration.

***cluster-name***

Creates the physical destination for every server instance in the specified cluster.

*instance-name*

Creates the physical destination for the specified server instance.

**Operands** *dest\_name*

A unique identifier for the JMS destination to be created.

**Examples** **EXAMPLE 1** Creating a JMS physical destination

The following subcommand creates a JMS physical queue named `PhysicalQueue` with non-default property values.

```
asadmin> create-jmsdest --desttype queue
--property maxNumMsgs=1000:maxBytesPerMsg=5k PhysicalQueue
Command create-jmsdest executed successfully.
```

**Exit Status** 0 subcommand executed successfully

1 error in executing the subcommand

**See Also** `create-jms-resource(1)`, `delete-jmsdest(1)`, `list-jmsdest(1)`, `flush-jmsdest(1)`

[asadmin\(1M\)](#)

