**JMS Overview**

Java Message Service (JMS) is a Java framework specification for messaging between applications. This specification was developed to supply a uniform messaging interface among enterprise applications.

Using a message service allows you to integrate the applications within an enterprise. For example, you may have several applications: one for customer relations, one for product inventory, and another for raw materials tracking. Each application is crucial to the operation of the enterprise, but even more crucial is communication between the applications to ensure the smooth flow of business processes. Message-oriented-middleware (MOM) creates a common communication protocol between these applications and allows you to easily integrate new and existing applications in your enterprise computing environment.

JMS provides 2 delivery modes for messages

* PERSISTENT
* NON\_PERSISTENT

EMS adds the 3rd delivery mode

PERSISTENT

NON\_PERSISTENT

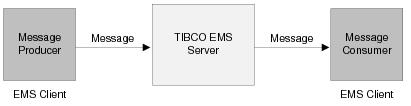
RELIABLE\_DELIVERY

EMS extends *MapMessage*and*StreamMessage*body types of JMS which allow EMS to exchange messages with TIBCO RV and ActiveEnterprise formats.

**JMS Message Models**

The TIBCO EMS server acts as an intermediary for the message and manages its delivery to the correct destination. The server also provides enterprise-class functionality such as fault-tolerance, message routing, and communication with other messaging systems, such as TIBCO FTL, TIBCO Rendezvous, and TIBCO SmartSockets.

[Figure 1](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.003.htm" \l "2829844', '');" \o "JMS Message Models) illustrates an application producing a message, sending it by way of the server, and a different application receiving the message.

*Figure 1**Message Delivery*

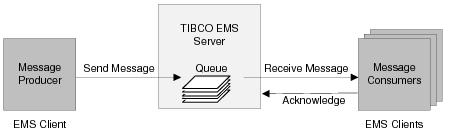
JMS supports these messaging models:

|  |  |
| --- | --- |
| • | [Point-to-Point](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.003.htm" \l "2829849', '');" \o "JMS Message Models) (queues) |
| • | [Publish and Subscribe](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.003.htm" \l "3688093', '');" \o "JMS Message Models) (topics) |

**Point-to-Point**

Point-to-point messaging has one producer and one consumer per message. This style of messaging uses a *queue* to store messages until they are received. The message producer sends the message to the queue; the message consumer retrieves messages from the queue and sends acknowledgment that the message was received.

More than one producer can send messages to the same queue, and more than one consumer can retrieve messages from the same queue. The queue can be configured to be exclusive, if desired. If the queue is exclusive, then all queue messages can only be retrieved by the first consumer specified for the queue**. Exclusive queues are useful when you want only one application to receive messages for a specific queue.** If the queue is not exclusive, any number of receivers can retrieve messages from the queue. **Non-exclusive queues are useful for balancing the load of incoming messages across multiple receivers**.

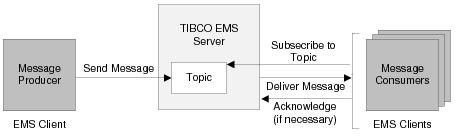
*Figure 2**Point-to-point messages*

**Publish and Subscribe**

In a publish and subscribe message system, producers address messages to a *topic*. In this model, the producer is known as a *publisher* and the consumer is known as a *subscriber*.

Many publishers can publish to the same topic, and a message from a single publisher can be received by many subscribers. Subscribers subscribe to topics, and all messages published to the topic are received by all subscribers to the topic. **This type of message protocol is also known as *broadcast* messaging because messages are sent over the network and received by all interested subscribers, similar to how radio or television signals are broadcast and received.**

[Figure 3](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.003.htm" \l "2829873', '');" \o "JMS Message Models) illustrates publish and subscribe messaging. Each message consumer subscribes to a topic. When a message is published to that topic, all subscribed consumers receive the message.

*Figure 3**Publish and subscribe messages*

**Durable Subscribers for Topics**

By default, subscribers only receive messages when they are active. If messages arrive on the topic when the subscriber is not available, the subscriber does not receive those messages.

The EMS APIs allow you to create durable subscribers to ensure that messages are received, even if the message consumer is not currently running. **Messages for durable subscriptions are stored on the server as long as durable subscribers exist for the topic, or until the message expiration time for the message has been reached, or until the storage limit has been reached for the topic.** Durable subscribers can receive messages from a durable subscription even if the subscriber was not available when the message was originally delivered.

When an application restarts and recreates a durable subscriber with the same ID, all messages stored on the server for that topic are delivered to the durable subscriber.

See [Creating a Message Consumer](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.095.htm" \l "4999850', '');" \o "Creating a Message Consumer) for details on how to create durable subscribers.

**Shared Subscriptions for Topics**

Shared subscriptions allow an application to share the work of receiving messages on a topic among multiple message consumers. When multiple consumers share a subscription, only one consumer in the group receives each new message. This is similar in function to a queue; however, there are no restrictions placed on the type of consumers to the topic, meaning that a topic can have a mix of shared and not shared, durable and non-durable consumers. When a message is published to the topic, the same message goes to all the matching subscriptions.

Shared subscriptions are created with a specific name, and optionally a client ID. Consumers sharing the subscription specify this name when subscribing to the topic. If the shared subscription type is durable, it persists an continues to accumulate messages until deleted. If the shared subscription type is non-durable, it persists only so long as subscribers exist.

For example, the topic foo might have the following subscriptions:

|  |  |
| --- | --- |
| • | not shared, non-durable subscription |
| • | not shared, durable subscription |

|  |  |
| --- | --- |
| • | shared, non-durable subscription called mySharedSub with three shared consumers |
| • | shared, durable subscription called myDurableSharedSub with two shared consumers |

If a message is received on foo, each of the above four subscriptions receive that same message. For the shared subscriptions mySharedSub and myDurableSharedSub, the message is delivered to only one if its respective shared consumers.

If the shared consumers of the shared durable subscription myDurableSharedSub are closed, then the shared durable subscription continues to exist and accumulate messages until it is deleted, or until the application creates a new durable shared consumer named myDurableSharedSub to resume this subscription. If the shared consumers of mySharedSub are all closed, the subscription is removed from topic foo.

See [Creating a Message Consumer](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.095.htm" \l "4999850', '');" \o "Creating a Message Consumer) for details on how to create shared subscriptions.

**EMS Destination Features**

TIBCO Enterprise Message Service allows you to configure destinations to enhance the functionality of each messaging model.

The EMS destination features allow you to:

|  |  |
| --- | --- |
| • | Set a [secure](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.030.htm" \l "2844216', '');" \o "Destination Properties) mode for access control at the queue or topic level, so that some destinations may require permission and others may not. See [Destination Control](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.063.htm#2643014', '');). |
| • | Set threshold limits for the amount of memory used by the EMS server to store messages for a topic or a queue and fine-tune the server’s response to when the threshold is exceeded. See [flowControl](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.030.htm" \l "3442484', '');" \o "Destination Properties)[on page 63](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.030.htm#3442484', '');) and [overflowPolicy](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.030.htm" \l "3465530', '');" \o "Destination Properties)[on page 67](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.030.htm#3465530', '');). |

|  |  |
| --- | --- |
| • | Route messages sent to destinations to other servers. See [Working With Routes](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.146.htm" \l "2432748', '');" \o "Working With Routes). |
| • | Create bridges between destinations of the same or different types to create a hybrid messaging model for your application. This can be useful if your application requires that you send the same message to both a topic and a queue. For more information on creating bridges between destinations and situations where this may be useful, see [Destination Bridges](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.035.htm" \l "2844693', '');" \o "Destination Bridges). |

|  |  |
| --- | --- |
| • | Control the flow of messages to a destination. This is useful when message producers send messages much faster than message consumers can receive them. For more information on flow control, see [Flow Control](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.036.htm" \l "2844762', '');" \o "Flow Control). |
| • | Exchange messages with other message services, such as TIBCO FTL, TIBCO Rendezvous, and TIBCO SmartSockets. Queues can receive messages from any of these services. Topics can either receive or send messages. See [Working with TIBCO FTL](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.100.htm" \l "5296456', '');" \o "Working with TIBCO FTL), [Working With TIBCO Rendezvous](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.106.htm#2338614', '');), and [Working With TIBCO SmartSockets](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.115.htm#2187389', '');). |

|  |  |
| --- | --- |
| • | Set queues to be exclusive or non-exclusive. Only one receiver can receive messages from an exclusive queue. More than one receiver can receive messages from non-exclusive queues. See [exclusive](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.030.htm" \l "3859670', '');" \o "Destination Properties)[on page 61](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.030.htm#3859670', '');). |
| • | Specify a redelivery policy for queues. When messages must be redelivered, you can specify a property on the queue that determines the maximum number of times a message should be redelivered. See [maxRedelivery](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.030.htm" \l "3459791', '');" \o "Destination Properties)[on page 66](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.030.htm#3459791', '');). |

|  |  |
| --- | --- |
| • | Trace and log all messages passing through a destination. See [trace](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.030.htm" \l "3478922', '');" \o "Destination Properties)[on page 75](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.030.htm#3478922', '');). |
| • | Include the user name of the message producer in the message. See [sender\_name](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.030.htm" \l "2844365', '');" \o "Destination Properties)[on page 73](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.030.htm#2844365', '');) and [sender\_name\_enforced](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.030.htm" \l "2844380', '');" \o "Destination Properties)[on page 74](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.030.htm#2844380', '');). |

|  |  |
| --- | --- |
| • | Administrator operations can use wildcards in destination names. The wildcard destination name is the parent, and any names that match the wildcard destination name inherit the properties of the parent. See [Wildcards](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.033.htm" \l "2844607', '');" \o "Wildcards). |
| • | Use the store property to cause messages sent to a destination to be written to a store file. Set the destination store to store=$sys.failsafe to direct the server to write messages to the file synchronously and guarantee that messages are not lost under any circumstances. See [store](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.030.htm" \l "2844413', '');" \o "Destination Properties)[on page 74](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.030.htm#2844413', '');) for more information. |

|  |  |
| --- | --- |
| • | Specify that a consumer is to receive batches of messages in the background to improve performance. Alternatively, you can specify that queue receivers are to only receive one message at a time. See [prefetch](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.030.htm" \l "3471588', '');" \o "Destination Properties)[on page 69](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.030.htm#3471588', '');) for more information. |

**Why JNDI?**

The administration interfaces allow you to create and manage administered objects such as ConnectionFactories, Topics, and Queues. **EMS clients can retrieve references to these administered objects by using Java Naming and Directory Interface (JNDI).** Creating static administered objects allows clients to use these objects without having to implement the objects within the client.

**Modes, Roles, and States**

The *mode* of an EMS server is determined by its configuration, and dictates how it operates in its environment. If a fault tolerant mode is selected, two EMS servers are required and each operates in a defined *role*. How an EMS server is operating at any given moment can be determined by viewing its fault tolerant *state*.

For example, an EMS server operating in fault tolerant mode can play either a primary or secondary role. Once both EMS servers in the fault tolerant pair have been started, one of the two servers will be in the active state while its peer will be in the standby state. **In the event of a failover, the server that was standby becomes active.**

**Modes**

By default, the EMS server operates in standalone mode. However, it can also be configured to run in a fault tolerant mode:

|  |  |
| --- | --- |
| • | Standalone — the default EMS server mode. |
| • | Classic Fault Tolerant — configured through the [ft\_active](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.058.htm" \l "4313444', '');" \o "tibemsd.conf) parameter. |

**Roles**

Each server operating in a fault tolerant mode has a distinct role: primary or secondary.

These roles are implicit for EMS servers started using tibemsd.conf files. They are explicit for EMS servers started using a JSON configuration file. For JSON-configured servers, the primary server is the EMS server started without the -secondary command line parameter, while the secondary server is started with it. In the .conf files, each server in the fault tolerant pair has a distinct tibemsd.conf file.

**States**

The state of the EMS server will tell you about its current operations. Use the [info](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.054.htm" \l "4309810', '');" \o "Command Listing) or [show state](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.054.htm#4310933', '');) command in the administration tool to determine the state of the EMS server.

|  |  |
| --- | --- |
| *Table 4**tibemsd States* | |
| **State** | **Description** |
| active | The server is fully operational and ready to service clients. |
| standby | The server is in classic fault tolerant mode and is ready to take over should its peer fail. |

**Security**

For communication security between servers and clients, and between servers and other servers, you must explicitly configure SSL within EMS.

Secure Sockets Layer (SSL) is a protocol for transmitting encrypted data over the Internet or an internal network. SSL works by using public and private keys to encrypt data that is transferred over the SSL connection. Most web browsers support SSL, and many Web sites and Java applications use the protocol to obtain confidential user information, such as credit card numbers.

EMS supports SSL between the following components:

|  |  |
| --- | --- |
| • | between an EMS client and the EMS server |
| • | between the administration tool and the EMS server |

|  |  |
| --- | --- |
| • | between the administration APIs and the EMS server |
| • | between routed servers |

|  |  |
| --- | --- |
| • | between fault-tolerant servers |

**Fault Tolerance**

You can configure EMS servers as primary and secondary servers to provide fault tolerance for your environment. The primary and secondary servers act as a pair, one of them starting out in the active state and the other in the standby state. The active server accepts client connections and performs the work of handling messages, while the standby server acts as a backup in case of failure. If the active server fails, the standby server assumes operation and becomes the active server.

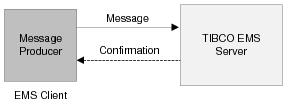
**Message Delivery Modes**

The [JMSDeliveryMode](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.015.htm" \l "2792609', '');" \o "JMS Message Structure) message header field defines the delivery mode for the message. JMS supports PERSISTENT and NON\_PERSISTENT delivery modes for both topic and queue. EMS extends these delivery modes to include a [RELIABLE\_DELIVERY](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.017.htm#2792946', '');) mode.

You can set the default delivery mode for the Message Producer, as described in [Configuring a Message Producer](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.094.htm" \l "2787774', '');" \o "Creating a Message Producer). This default delivery mode can be overridden by the client when sending a message, as described in [Sending Messages](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.096.htm#3414522', '');).

**PERSISTENT**

As shown in [Figure 4](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.017.htm" \l "2792934', '');" \o "Message Delivery Modes), when a producer sends a PERSISTENT message, the producer must wait for the server to reply with a confirmation. The message is persisted on disk by the server. This delivery mode ensures delivery of messages to the destination on the server in almost all circumstances. However, the cost is that this delivery mode incurs two-way network traffic for each message or committed transaction of a group of messages.

*Figure 4**Persistent Message Delivery*

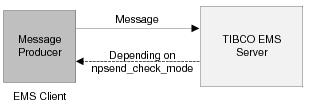
**NON\_PERSISTENT**

Sending a NON\_PERSISTENT message omits the overhead of persisting the message on disk to improve performance.

If [authorization](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.058.htm" \l "4567179', '');" \o "tibemsd.conf) is disabled on the server, the server does not send a confirmation to the message producer.

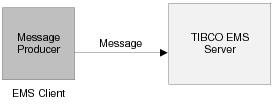
If authorization is enabled on the server, the default condition is for the producer to wait for the server to reply with a confirmation in the same manner as when using PERSISTENTmode.

Regardless of whether authorization is enabled or disabled, you can use the npsend\_check\_mode parameter in the [tibemsd.conf](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.058.htm" \l "4311405', '');" \o "tibemsd.conf) file to specify the conditions under which the server is to send confirmation of NON\_PERSISTENT messages to the producer. See the description for [npsend\_check\_mode](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.058.htm" \l "5212560', '');" \o "tibemsd.conf)[on page 206](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.058.htm#5212560', '');) for details.

*Figure 5**Non-Persistent Message Delivery*

**RELIABLE\_DELIVERY**

EMS extends the JMS delivery modes to include reliable delivery. Sending a RELIABLE\_DELIVERY message omits the server confirmation to improve performance regardless of the [authorization](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.058.htm#4567179', '');) setting.

*Figure 6**Reliable Message Delivery*

When using RELIABLE\_DELIVERY mode, the server never sends the producer a receipt confirmation or access denial and the producer does not wait for it. Reliable mode decreases the volume of message traffic, allowing higher message rates, which is useful for messages containing time-dependent data, such as stock price quotations.

When you use the reliable delivery mode, the client application does not receive any response from the server. Therefore, all publish calls will always succeed (not throw an exception) unless the connection to the server has been terminated.

In some cases a message published in reliable mode may be disqualified and not handled by the server because the destination is not valid or access has been denied. In this case, the message is not sent to any message consumer**. However, unless the connection to the server has been terminated, the publishing application will not receive any exceptions, despite the fact that no consumer received the message.**

**npsend\_check\_mode**

npsend\_check\_mode = [always | never | temp\_dest | auth | temp\_auth]

Specifies when the server is to provide confirmation upon receiving a [NON\_PERSISTENT](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.017.htm" \l "2792936', '');" \o "Message Delivery Modes) message from a producer.

The npsend\_check\_mode parameter applies only to producers sending messages using NON\_PERSISTENT delivery mode and non-transactional sessions.

Message confirmation has a great deal of impact on performance and should only be enabled when necessary. The circumstances in which a producer might want the server to send confirmation a NON\_PERSISTENT message are:

|  |  |
| --- | --- |
| • | When [authorization](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.058.htm" \l "4567179', '');" \o "tibemsd.conf) is enabled, so the producer can take action if permission to send the message is denied by the server. |
| • | When sending to a temporary destination, so the producer can take action if the message is sent to a temporary destination that has been destroyed. |

|  |  |
| --- | --- |
| • | The message exceeded queue/topic limit (requires [rejectIncoming](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.030.htm" \l "4880288', '');" \o "Destination Properties) policy for topics). |
| • | Bridging of the message has failed. |

|  |  |
| --- | --- |
| • | The server is out of memory or has encountered some other severe error. |

The possible npsend\_check\_mode parameter modes are:

|  |  |
| --- | --- |
| • | default (no mode specified) - this means the server only provides confirmation of a [NON\_PERSISTENT](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.017.htm#2792936', '');) message if [authorization](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.058.htm#4567179', '');) is enabled. |
| • | always - the server always provides confirmation of a NON\_PERSISTENT message. |

|  |  |
| --- | --- |
| • | never - the server never provides confirmation of NON\_PERSISTENT messages. |
| • | temp\_dest - the server provides confirmation of a NON\_PERSISTENT message only when sending to a temporary destination. |

|  |  |
| --- | --- |
| • | auth - the server provides confirmation of a NON\_PERSISTENT message only if authorization was enabled when the connection was created. |
| • | temp\_auth - the server provides confirmation of a NON\_PERSISTENT message if sending to a temporary destination or if authorization was enabled when the connection was created. |

**authorization**

authorization = enabled | disabled

Enable or disable server authorization.

Authorization is disabled by default. If you require that the server verify user credentials and permissions on secure destinations, you must enable this parameter.

See [Enabling Access Control](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.063.htm" \l "2642992', '');" \o "Enabling Access Control) for more information.

For example:

authorization = enabled

See [Chapter 8, Authentication and Permissions](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.060.htm" \l "2642876', '');" \o "Authentication and Permissions) for more information about these parameters.

**Enabling Access Control**

Administrators can enable or disable access control for the server. Administrators can also enable and disable permission checking for specific destinations.

**Server Control**

The property in the main configuration file enables or disables the checking of permissions for all destinations managed by the server. The [authorization](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.058.htm" \l "4567179', '');" \o "tibemsd.conf) property also enables or disables verification of user names and passwords.

|  |  |
| --- | --- |
| * | The default setting is disabled. For secure deployments, the administrator must explicitly set authorization to enabled. |

When authorization is disabled, the server grants any connection request, and does not check permissions when a client accesses a destination (for example, publishing a message to a topic).

When authorization is enabled, the server grants connections only from valid authenticated users. The server checks permissions for client operations involving secure destinations.

To enable authorization, either edit [tibemsd.conf](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.058.htm" \l "4311405', '');" \o "tibemsd.conf) (set the authorization property to enabled, and restart the server). Or you can use the tibemsadmin tool to dynamically enable authorization with the following [set server](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.054.htm#4309956', '');) command:

   set server authorization=enabled

Authorization does affect connections between fault-tolerant server pairs; see [Authorization and Fault-Tolerant Servers](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.142.htm" \l "2488529', '');" \o "Configuring Fault-Tolerant Servers).

Administrators must always log in with the correct administration username and password to perform any administrative function—even when authorization is disabled.

**Destination Control**

When server [authorization](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.058.htm" \l "4567179', '');" \o "tibemsd.conf) is enabled, the server checks user names and password of all connections without exceptions. However, operations on destinations, such as sending a message or receiving a message, are not verified unless the destination has enabled the [secure](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.030.htm#2844216', '');) property on the destination. All operations by applications on the destination with secure enabled are verified by the server according to the permissions listed in acl.conf. Destinations with secure disabled continue to operate without any restrictions.

|  |  |
| --- | --- |
| * | The secure property is independent of SSL-level security. The secure property controls only basic authentication and permission verification. It does not affect the security of communication between clients and server. |

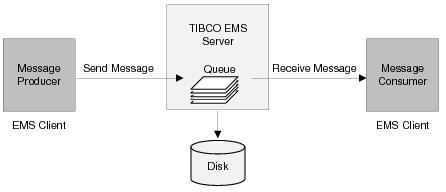
When a destination does not have the secure property set, any authenticated user can perform any actions on that topic or queue.

See [Destination Properties](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.030.htm" \l "3430490', '');" \o "Destination Properties) for more information about destination properties.

**When a destination does not have the secure property set, any authenticated user can perform any actions on that topic or queue.**

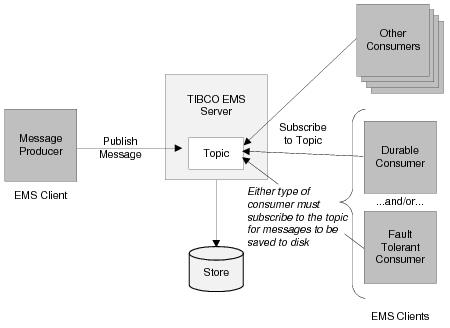
**Persistent Messages Sent to Queues**

Persistent messages sent to a queue are always written to disk. Should the server fail before sending persistent messages to subscribers, the server can be restarted and the persistent messages will be sent to the subscribers when they reconnect to the server.

*Figure 7**Persistent Messages Sent to a Queue*

**Persistent Messages Published to Topics**

Persistent messages published to a topic are written to disk *only* if that topic has at least one durable subscriber or one subscriber with a fault-tolerant connection to the EMS server. In the absence of a durable subscriber or subscriber with a fault-tolerant connection, there are no subscribers that need messages resent in the event of a server failure **(message will not written to a disk if there is no durable subscribers or subscriber with a fault-tolerant connection)** In this case, the server does not needlessly save persistent messages. This improves performance by eliminating the unnecessary disk I/O to persist the messages.

*Figure 8**Persistent Messages Published to a Topic*

This behavior is consistent with the JMS specification because durable subscribers to a topic cause published messages to be saved. Additionally, subscribers to a topic that have a fault-tolerant connection need to receive messages from the new active server after a failover. However, non-durable subscribers without a fault-tolerant connection that re-connect after a server failure are considered newly created subscribers and are not entitled to receive any messages created prior to the time they are created (that is, messages published before the subscriber re-connects are not resent).

**Persistent Messages and Synchronous File Storage**

When using file storage, persistent messages received by the EMS server are by default written asynchronously to disk. This means that, when a producer sends a persistent message, the server does not wait for the write-to-disk operation to complete before returning control to the producer. Should the server fail before completing the write-to-disk operation, the producer has no way of detecting the failure to persist the message and taking corrective action.

You can set the mode parameter to sync for a given file storage in the [stores.conf](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.059.htm" \l "4315119', '');" \o "Using Other Configuration Files) file to specify that persistent messages for the topic or queue be synchronously written to disk. When mode = sync, the persistent producer remains blocked until the server has completed the write-to-disk operation.

Each EMS server writes persistent messages to a store file. To prevent two servers from using the same store file, each server restricts access to its store file for the duration of the server process. For details on how EMS manages shared store files, see [How EMS Manages Access to Shared Store Files](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.049.htm" \l "3297784', '');" \o "How EMS Manages Access to Shared Store Files).

**JMS Message Structure**

JMS messages have a standard structure. This structure includes the following sections:

|  |  |
| --- | --- |
| • | Header (required) |
| • | Properties (optional) |

|  |  |
| --- | --- |
| • | Body (optional) |

**JMS Message Header Fields**

The header contains 11 predefined fields that contain values used to route and deliver messages. [Table 5](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.015.htm" \l "2792589', '');" \o "JMS Message Structure) describes the message header fields.

|  |  |  |
| --- | --- | --- |
| *Table 5**JMS Message Headers* | | |
| **Header Field** | **Set by** | **Comments** |
| JMSDestination | send or publishmethod | Destination to which message is sent |
| JMSDeliveryMode | send or publishmethod | Persistent or non-persistent message. The default is persistent.  EMS extends the delivery mode to include a RELIABLE\_DELIVERY mode, as described in [RELIABLE\_DELIVERY on page 28](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.017.htm#2792946', '');). |
| JMSExpiration | send or publishmethod | Length of time that message will live before expiration. If set to 0, message does not expire. The time-to-live is specified in milliseconds.  If the server [expiration](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.030.htm" \l "3442137', '');" \o "Destination Properties) property is set for a destination, it will override the JMSExpiration value set by the message producer. |
| JMSDeliveryTime | send or publishmethod | Read-only field. If the message producer has a delivery delay set, then the time returned here after calling the send method represents the earliest time when the EMS server will deliver the message to consumers. Once the message has been received, it carries that same value. This value is calculated by adding the delivery delay value held by the message producer to the time the message was sent. For transactions, the delivery time is calculated using the time the client sends the message, not the time the transaction is committed.  For more information, see [Delivery Delay](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.037.htm" \l "5022423', '');" \o "Delivery Delay). |
| JMSPriority | send or publishmethod | Uses a numerical ranking, between 0 and 9, to define message priority as normal or expedited. Larger numbers represent higher priority.  See [Message Priority](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.016.htm" \l "2820297', '');" \o "Message Priority) for more information. |
| JMSMessageID | send or publishmethod | Value uniquely identifies each message sent by a provider. |
| JMSTimestamp | send or publishmethod | Timestamp of time when message was handed off to a provider to be sent. Message may actually be sent later than this timestamp. |
| JMSCorrelationID | message client | This ID can be used to link messages, such as linking a response message to a request message. Entering a value in this field is optional. The JMS Correlation ID has a recommended maximum of 4 KB. Higher values may result in the message being rejected. |
| JMSReplyTo | message client | A destination to which a message reply should be sent. Entering a value for this field is optional. |
| JMSType | message client | Message type identifier. |
| JMSRedelivered | JMS provider | If this field is set, it is possible that the message was delivered to the client earlier, but not acknowledged at that time. |

**EMS Message Properties**

In the properties area, applications, vendors, and administrators on JMS systems can add optional properties. The properties area is optional, and can be left empty. The JMS specification describes the JMS message properties. This section describes the message properties that are specific to EMS.

TIBCO-specific property names begin with JMS\_TIBCO. Client programs may use the TIBCO-specific properties to access EMS features, but not for communicating application-specific information among client programs.

The EMS properties are summarized in [Table 6](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.015.htm" \l "2792686', '');" \o "JMS Message Structure) and described in more detail in subsequent sections in this chapter.

|  |  |  |
| --- | --- | --- |
| *Table 6**Summary of message properties* | | |
| **Property** | **Description** | **More Info** |
| JMS\_TIBCO\_CM\_PUBLISHER | Correspondent name of an RVCM sender for messages imported from TIBCO Rendezvous. | [440](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.113.htm" \l "2719396', '');" \o "Message Translation) |
| JMS\_TIBCO\_CM\_SEQUENCE | Sequence number of an RVCM message imported from TIBCO Rendezvous. | [440](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.113.htm" \l "2719396', '');" \o "Message Translation) |
| JMS\_TIBCO\_COMPRESS | Allows messages to be compressed for more efficient storage. | [41](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.021.htm" \l "3942826', '');" \o "Message Compression) |
| JMS\_TIBCO\_DISABLE\_SENDER | Specifies that the user name of the message sender should not be included in the message, if possible. | [23](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.015.htm" \l "2815845', '');" \o "JMS Message Structure) |
| JMS\_TIBCO\_IMPORTED | Set by the server when the message has been imported from TIBCO FTL, Rendezvous, or SmartSockets. | [440](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.113.htm" \l "2339345', '');" \o "Message Translation)  [460](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.122.htm" \l "2307910', '');" \o "Message Translation) |
| JMS\_TIBCO\_MSG\_EXT | Extends the functionality of the [MapMessage](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.015.htm" \l "2792872', '');" \o "JMS Message Structure) and [StreamMessage](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.015.htm" \l "2792889', '');" \o "JMS Message Structure) body types to include submessages or arrays. | [18](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.014.htm" \l "2799085', '');" \o "EMS Extensions to JMS Messages)  [440](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.113.htm" \l "2339345', '');" \o "Message Translation)  [460](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.122.htm" \l "2307910', '');" \o "Message Translation) |
| JMS\_TIBCO\_MSG\_TRACE | Specifies the message should be traced from producer to consumer. | [474](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.125.htm" \l "2041888', '');" \o "Message Tracing) |
| JMS\_TIBCO\_PRESERVE\_UNDELIVERED | Specifies the message is to be placed on the undelivered message queue if the message must be removed. | [22](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.015.htm" \l "2815822', '');" \o "JMS Message Structure) |
| JMS\_TIBCO\_SENDER | Contains the user name of the message sender. | [23](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.015.htm" \l "2815845', '');" \o "JMS Message Structure) |
| JMS\_TIBCO\_SS\_SENDER | When the EMS server imports a message from TIBCO SmartSockets, it sets this property to the SmartSockets senderheader field (in SmartSockets syntax). | [460](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.122.htm" \l "2187900', '');" \o "Message Translation) |

**Undelivered Message Queue**

If a message could not be delivered for one of the reasons below, the server checks the message’s [JMS\_TIBCO\_PRESERVE\_UNDELIVERED](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.015.htm" \l "2792799', '');" \o "JMS Message Structure) property. If that property is set to true, the server moves the message to the undelivered message queue, $sys.undelivered. Otherwise, the message is deleted by the server.

The server will examine a message’s JMS\_TIBCO\_PRESERVE\_UNDELIVERED property if any of the following conditions are met:

|  |  |
| --- | --- |
| • | the message has expired |
| • | the message has exceeded the value specified by the [maxRedelivery](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.030.htm" \l "3459791', '');" \o "Destination Properties) property on a queue |

|  |  |
| --- | --- |
| • | the message had a delivery delay that has expired and was sent to a destination that has reached its [maxmsgs](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.030.htm" \l "3454054', '');" \o "Destination Properties) limit and also has [overflowPolicy](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.030.htm" \l "3465530', '');" \o "Destination Properties)=rejectIncoming |

$sys.undelivered is a system queue that is always present and cannot be deleted. To make use of it, the application that sends or publishes the message must set the boolean JMS\_TIBCO\_PRESERVE\_UNDELIVERED property to true before sending or publishing the message.

You can only set the undelivered property on individual messages, there is no way to set the undelivered message queue as an option at the per-topic or per-queue level.

You should create a queue receiver to receive and handle messages as they arrive on the undelivered message queue. If you wish to remove messages from the undelivered message queue without receiving them, you can purge the $sys.undelivered queue with the administration tool, using thepurge queue command described under [Command Listing](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.054.htm#4309370', '');). You can also remove messages using the administrative API included with TIBCO Enterprise Message Service.

Note that $sys.undelivered ignores the [global](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.030.htm#3444052', '');) destination property setting. Messages in the undelivered message queue are not routed to other servers.

**Filtering Messages in the Undelivered Message Queue**

You can filter messages in the undelivered message queue by destination using a selector. Note that this is an exception to the JMS Specification that is made only for messages in the undelivered message queue. In the undelivered message queue, the JMSDestination header field can be used in a selector the same way that a supported header field or any other message property with a string value is used.

The expected value of the JMSDestination field depends on the original message destination type and name:

JMSDestination *operator* ’Topic|Queue[*destination\_name*]’

For example:

JMSDestination='Queue[A]'

JMSDestination='Topic[B7]'

JMSDestination NOT LIKE 'Queue[A]'

JMSDestination LIKE 'Queue[A]'

JMSDestination LIKE 'Q%'

JMSDestination IS NOT NULL

JMSDestination IN ('Queue[H]','Queue[J]')

JMSDestination NOT IN ('Topic[H]','Topic[J]')

JMSDestination='Queue[A]' OR JMSDestination='Queue[B]'

**Including the Message Sender**

Within a message, EMS can supply the user name given by the message producer when a connection is created. The sender\_name and sender\_name\_enforced server properties on the destination determine whether the message producer’s user name is included in the sent message.

When a user name is included in a message, a message consumer can retrieve that user name by getting the string message property named [JMS\_TIBCO\_SENDER](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.015.htm" \l "2792809', '');" \o "JMS Message Structure).

When the sender\_name property is enabled and the sender\_name\_enforced property is not enabled on a destination, message producers can specify that the user name is to be left out of the message. Message producers can specify the [JMS\_TIBCO\_DISABLE\_SENDER](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.015.htm#2792735', '');) boolean property for a particular message, and the message producer’s user name will not be included in the message. However, if the sender\_name\_enforced property is enabled, the JMS\_TIBCO\_DISABLE\_SENDER property is ignored and the user name is always included in the message.

**JMS Message Bodies**

A JMS message has one of several types of message bodies, or no message body at all.

The types of messages are described in [Table 7](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.015.htm" \l "2792852', '');" \o "JMS Message Structure).

|  |  |
| --- | --- |
| *Table 7**JMS Message Types* | |
| **Message Type** | **Contents of Message Body** |
| Message | This message type has no body. This is useful for simple event notification. |
| TextMessage | A java.lang.String. |
| MapMessage | A set of name/value pairs. The names are java.lang.String objects, and the values are Java primitive value types or their wrappers. The entries can be accessed sequentially by enumeration or directly by name. The order of entries is undefined.  When EMS is exchanging messages with Rendezvous, you can generate a series of nested MapMessages, as described in [EMS Extensions to JMS Messages](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.014.htm" \l "2799085', '');" \o "EMS Extensions to JMS Messages). |
| BytesMessage | A stream of uninterrupted bytes. The bytes are not typed; that is, they are not assigned to a primitive data type. |
| StreamMessage | A stream of primitive values in the Java programming language. Each set of values belongs to a primitive data type, and must be read sequentially.  When EMS is exchanging messages with Rendezvous, you can generate a series of nested StreamMessages, as described in [EMS Extensions to JMS Messages](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.014.htm" \l "2799085', '');" \o "EMS Extensions to JMS Messages). |
| ObjectMessage | A serializable object constructed in the Java programming language. |

**Maximum Message Size**

EMS supports messages up to a maximum size of 512MB. However, we recommend that application programs use smaller messages, since messages approaching this maximum size will strain the performance limits of most current hardware and operating system platforms.

The JMS specification defines three levels of acknowledgement for non-transacted sessions:

|  |  |
| --- | --- |
| • | CLIENT\_ACKNOWLEDGE specifies that the consumer is to acknowledge all messages that have been delivered so far by the session. When using this mode, it is possible for a consumer to fall behind in its message processing and build up a large number of unacknowledged messages. |
| • | AUTO\_ACKNOWLEDGE specifies that the session is to automatically acknowledge consumer receipt of messages when message processing has finished.  **[Messages are automatically acknowledged when it is received]** |

|  |  |
| --- | --- |
| • | DUPS\_OK\_ACKNOWLEDGE specifies that the session is to "lazily" acknowledge the delivery of messages to the consumer. "Lazy" means that the consumer can delay acknowledgement of messages to the server until a convenient time; meanwhile the server might redeliver messages. This mode reduces session overhead. Should JMS fail, the consumer may receive duplicate messages. |

EMS extends the JMS session modes to include:

|  |  |
| --- | --- |
| • | [NO\_ACKNOWLEDGE](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.022.htm" \l "2793189', '');" \o "Message Acknowledgement) |
| • | [EXPLICIT\_CLIENT\_ACKNOWLEDGE](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.022.htm" \l "2793203', '');" \o "Message Acknowledgement) |

|  |  |
| --- | --- |
| • | [EXPLICIT\_CLIENT\_DUPS\_OK\_ACKNOWLEDGE](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.022.htm" \l "2793212', '');" \o "Message Acknowledgement) |
| * | The Simplified JMS API introduced in JMS 2.0 supports the session modes defined in the JMS specification: CLIENT\_ACKNOWLEDGE, AUTO\_ACKNOWLEDGE, DUPS\_OK\_ACKNOWLEDGE and SESSION\_TRANSACTED. However, it does not support the EMS extended session modes. |

The session mode is set when creating a Session, as described in [Creating a Session](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.091.htm" \l "2782650', '');" \o "Creating a Session).

**NO\_ACKNOWLEDGE**

NO\_ACKNOWLEDGE mode suppresses the acknowledgement of received messages. After the server sends a message to the client, all information regarding that message for that consumer is eliminated from the server. Therefore, there is no need for the client application to send an acknowledgement to the server about the received message. Not sending acknowledgements decreases the message traffic and saves time for the receiver, therefore allowing better utilization of system resources.

|  |  |
| --- | --- |
| * | Sessions created in no-acknowledge receipt mode cannot be used to create durable subscribers.  Also, queue receivers on a queue that is routed from another server are not permitted to specify NO\_ACKNOWLEDGE mode. |

**EXPLICIT\_CLIENT\_ACKNOWLEDGE**

EXPLICIT\_CLIENT\_ACKNOWLEDGE is like CLIENT\_ACKNOWLEDGE except it acknowledges only the individual message, rather than all messages received so far on the session.

One example of when EXPLICIT\_CLIENT\_ACKNOWLEDGE would be used is when receiving messages and putting the information in a database. If the database insert operation is slow, you may want to use multiple application threads all doing simultaneous inserts. As each thread finishes its insert, it can use EXPLICIT\_CLIENT\_ACKNOWLEDGE to acknowledge only the message that it is currently working on.

**EXPLICIT\_CLIENT\_DUPS\_OK\_ACKNOWLEDGE**

EXPLICIT\_CLIENT\_DUPS\_OK\_ACKNOWLEDGE is like DUPS\_OK\_ACKNOWLEDGE except it ’lazily" acknowledges only the individual message, rather than all messages received so far on the session.

*-----------------CONFIRM PALETTE NOT REQUIRED-----------------------------------Auto- Messages are automatically acknowledged when it is received.*

*NO\_ACKNOWLEDGE- Servers defines acknowledgment is not required.*

*DUPS\_OK- Not required Confirm palette to ACK messages. We may get duplicate messages. So, that y Duplicates OK.*

***-----------------CONFIRM PALETTE REQUIRED-----------------------------------***

*Client – When this acknowledgment mode is used Max Session Input box is enabled in JMSReceiver Palette. If MaxSession values =1 only one session will be running even if we have 100 msgs in the EMS Server queue. Confirm palette is used to acknowledge the message*

**TIBCO\_EMS\_ EXPLICIT\_CLIENT\_ACKNOWLEDGE- No Session blocking, so if 100 messages in EMS server 100 instance will be created and it will try to acknowledge the same. Confirm palette is used to acknowledge the message**

**TIBCO\_EMS\_EXPLICIT\_CLIENT\_DUPS\_OK\_ACKNOWLEDGE- Its laziy acknowledgement all messages received so far on the session.**

|  |
| --- |
| Copyright © TIBCO Software Inc. All Rights Reserved |
| Copyright © TIBCO Software Inc. All Rights Reserved |

[Chapter 6 Using the EMS Administration Tool](https://docs.tibco.com/pub/ems/8.4.0/doc/html/tib_ems_usr_guide/EMS.5.051.htm#4309370) : Command Listing

**Command Listing**

The command line interface of the administration tool allows you to perform a variety of functions. Note that when a system uses shared configuration files, the actions performed using the administration tool take effect only when connected to the active server.

|  |  |
| --- | --- |
| * | Many of the commands listed below accept arguments that specify the names of users, groups, topics or queues. For information about the syntax and that apply to these names, see [Naming Conventions](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.053.htm" \l "4309334', '');" \o "Naming Conventions). |
| * | Note that SSL commands are not listed in this table. SSL commands are listed in several tables in [Chapter 18, Using the SSL Protocol](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.128.htm" \l "1674339', '');" \o "Using the SSL Protocol). |

The following is an alphabetical listing of the commands including command syntax and a description of each command.

**add member**

add member *group\_name* *user\_name* [,*user2*,*user3*,...]

Add one or more users to the group. User names that are not already defined are added to the group as external users; see [Administration Commands and External Users and Groups](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.064.htm" \l "2643104', '');" \o "Users and Groups).

**addprop factory**

addprop factory *factory-name* *properties* ...

Adds properties to the factory. Property names are separated by spaces.

See [factories.conf](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.059.htm" \l "4314769', '');" \o "Using Other Configuration Files)[on page 251](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.059.htm#4314769', '');) for the list of factory properties.

An example is:

addprop factory MyTopicFactory ssl\_trusted=cert1.pem ssl\_trusted=cert2.pem ssl\_verify\_host=disabled

**addprop queue**

addprop queue *queue-name* *properties*,...

Adds properties to the queue. Property names are separated by commas.

For information on properties that can be assigned to queues, see [Destination Properties](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.030.htm" \l "3430490', '');" \o "Destination Properties).

**addprop route**

addprop route *route-name* prop=value[ *prop-value*...]

Adds properties to the route.

Destination (topic and queue) properties must be separated by commas but properties of routes and factories are separated with spaces.

You can set the zone\_name and zone\_type parameters when creating a route, but you cannot subsequently change them.

For route properties, see [Configuring Routes and Zones](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.151.htm" \l "1675257', '');" \o "Configuring Routes and Zones).

For the configuration file routes.conf, see [routes.conf on page 259](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.059.htm#4315040', '');).

**addprop topic**

addprop topic *topic\_name* *properties*,...

Adds properties to the topic. Property names are separated by commas.

For information on properties that can be assigned to topics, see [Destination Properties](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.030.htm" \l "3430490', '');" \o "Destination Properties).

**autocommit**

autocommit [on|off]

When autocommit is set to **on**, the changes made to the configuration files are automatically saved to disk after each command. When autocommit is set to **off**, you must manually use the commit command to save configuration changes to the disk.

By default, autocommit is set to **on** when interactively issuing commands.

Entering **autocommit** without parameters displays the current setting of autocommit (on or off).

|  |  |
| --- | --- |
| * | Regardless of the autocommit setting, the EMS server acts on each admin command immediately making it part of the configuration. The autocommit feature only determines when the configuration is written to the files. |

**commit**

commit

Commits all configuration changes into files on disk.

**compact**

compact *store-name* *max-time*

Compacts the store files for the specified store. Compaction is available for stores of type file and mstore, but is not available for stores of type dbstore.

Since compaction can be a lengthy operation and it blocks other operations, *max-time* specifies a time limit (in seconds) for the operation. Note that *max-time* must be a number greater than zero. For mstore files, you can optionally specify nolimit for the *max-time*. See below for more information.

For stores of type file:

|  |  |
| --- | --- |
| • | If truncation is not enabled for the store file, the compact command does not reduce the file size. Enable truncation using the file\_truncate parameter in the stores.conf file. See [stores.conf on page 261](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.059.htm#4315119', '');) for more information. |
| • | We recommend compacting the store files only when the Used Space usage is 30% or less (see [show store on page 170](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.054.htm#5215640', '');)). |

For stores of type mstore:

|  |  |
| --- | --- |
| • | Two types of compaction are available: |
| − | **Time-bound compact**  Use *max-time* to specify a time limit (in seconds) for the compact operation. Time-bound compaction may increase the fragmentation of the store files. This feature is not available by default. Before using it, you need to run the tibemsdbconvert tool with option -version 8.3 on the required mstore files. See [Using the tibemsdbconvert Tool](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.019.htm#5464209', '');) for more information. |

|  |  |  |  |
| --- | --- | --- | --- |
| − | | **No-limit compact**  Enter nolimit for *max-time*. This triggers a full re-write of the store **with no time limit.** Once started, it is not possible to interrupt the re-write. All other operations (creating new connections, sending and receiving messages, and so forth) are suspended until the store is fully re-written. This can take a very long time for large stores. Using nolimit effectively defragments a store file. | |
| • | Compaction for mstores is not affected by the value of the mstore\_truncate parameter. | |

**connect**

connect [*server-url* {admin|*user\_name*} *password*]

Connects the administration tool to the server. Any administrator can connect. An administrator is either the admin user, any user in the $admin group, or any user that has administrator permissions enabled. See [Administrator Permissions](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.062.htm#2691787', '');) for more information about administrator permissions.

*server-url*is usually in the form:

*protocol*://*host-name*:*port-number*

for example:

tcp://myhost:7222

The protocol can be tcp or ssl.

If a user name or password are not provided, the user is prompted to enter a user name and password, or only the password, if the user name was already specified in the command.

You can enter connect with no other options and the administrative tool tries to connect to the local server on the default port, which is 7222.

**create bridge**

create bridge source=*type*:*dest\_name* target=*type*:*dest\_name*  [selector=*selector*]

Creates a bridge between destinations.

*type* is either topic or queue.

For further information, see [bridges.conf](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.059.htm" \l "4314566', '');" \o "Using Other Configuration Files)[on page 249](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.059.htm#4314566', '');).

**create durable**

create durable *topic-name* *durable-name* [*property*, ... ,*property*]

Creates a static durable subscriber.

For descriptions of parameters and properties, and information about conflict situations, see [durables.conf](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.059.htm" \l "4314695', '');" \o "Using Other Configuration Files)[on page 250](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.059.htm#4314695', '');).

**create factory**

create factory *factory\_name* *factory\_parameters*

Creates a new connection factory.

For descriptions of factory parameters, see [factories.conf](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.059.htm" \l "4314769', '');" \o "Using Other Configuration Files)[on page 251](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.059.htm#4314769', '');).

**create group**

create group *group\_name* "*description"*

Creates a new group of users.

Initially, the group is empty. You can use the [add member](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.054.htm" \l "4309393', '');" \o "Command Listing) command to add users to the group.

**create jndiname**

create jndiname *new\_jndiname* topic|queue|jndiname *name*

Creates a JNDI name for a topic or queue, or creates an alternate JNDI name for a topic that already has a JNDI name.

For example:

create jndiname FOO jndiname BAR

will create new JNDI name FOO referring the same object referred by JNDI name BAR

**create queue**

create queue *queue\_name* [*properties*]

Creates a queue with the specified name and properties. The possible queue properties are described in [Destination Properties](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.030.htm" \l "3430490', '');" \o "Destination Properties). Properties are listed in a comma-separated list, as described in [queues.conf on page 258](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.059.htm#4314998', '');).

**create route**

create route *name* url=*URL* [*properties* ...]

Creates a route.

The *name* must be the name of the other server to which the route connects.

The local server connects to the destination server at the specified URL. If you have configured fault-tolerant servers, you may specify the URL as a comma-separated list of URLs.

The route properties are listed in [routes.conf](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.059.htm" \l "4315040', '');" \o "Using Other Configuration Files)[on page 259](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.059.htm#4315040', '');) and are specified as a space-separated list of parameter name and value pairs.

You can set the zone\_name and zone\_type parameters when creating a route, but you cannot subsequently change them.

If a passive route with the specified *name* already exists, this command promotes it to an *active-active* route; see [Active and Passive Routes](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.150.htm#2133860', '');).

For additional information on route parameters, see [Configuring Routes and Zones](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.151.htm" \l "1675257', '');" \o "Configuring Routes and Zones).

**create rvcmlistener**

create rvcmlistener *transport\_name* *cm\_name* *subject*

Registers an RVCM listener with the server so that any messages exported to a tibrvcm transport (including the first message sent) are guaranteed for the specified listener. This causes the server to perform the TIBCO Rendezvous call tibrvcmTransport\_AddListener.

The parameters are:

|  |  |
| --- | --- |
| • | *transport\_name* — the name of the transport to which this RVCM listener applies. |
| • | *cm\_name* — the name of the RVCM listener to which topic messages are to be exported. |

|  |  |
| --- | --- |
| • | *subject* — the RVCM subject name that messages are published to. This should be the same name as the topic names that specify the export property. |

For more information, see [tibrvcm.conf](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.059.htm" \l "4315329', '');" \o "Using Other Configuration Files)[on page 265](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.059.htm#4315329', '');) and [Rendezvous Certified Messaging (RVCM) Parameters](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.108.htm#3917275', '');).

**create topic**

create topic *topic\_name* [*properties*]

Creates a topic with specified name and properties. See [Destination Properties](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.030.htm" \l "3430490', '');" \o "Destination Properties) for the list of properties. Properties are listed in a comma-separated list, as described in [topics.conf on page 267](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.059.htm#4315356', '');).

**create user**

create user *user\_name* ["*user\_description*"] [*password*=*password*]

Creates a new user. Following the user name, you can add an optional description of the user in quotes. The password is optional and can be added later using the set password command.

|  |  |
| --- | --- |
| * | User names cannot contain colon (:) characters. |

**delete all**

delete all users|groups|topics|queues|durables [*topic-name-pattern*|*queue-name-pattern*]

If used as delete all users|groups|topics|queues|durables without the optional parameters, the command deletes all users, groups, topics, or queues (as chosen).

If used with a topic or queue, and the optional parameters, such as:

delete all topics|queues *topic-name-pattern*|*queue-name-pattern*

the command deletes all topics and queues that match the topic or queue name pattern.

**delete bridge**

delete bridge source=*type*:*dest\_name* target=*type*:*dest\_name*

Delete the bridge between the specified source and target destinations.

*type* is either topic or queue.

See [Destination Bridges](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.035.htm" \l "2844693', '');" \o "Destination Bridges) for more information on bridges.

**delete connection**

delete connection *connection-id*

Delete the named connection for the client. The connection ID is shown in the first column of the connection description printed by show connection.

**delete durable**

delete durable *durable-name clientID*

Delete the named durable subscriber.

When both the durable name and the client ID are specified, the EMS Server looks for a durable named *clientID*:*durable-name* in the list of durables. If a matching durable subscriber is not found, the administration tool prints an error message including the fully qualified durable name.

See also, [Conflicting Specifications](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.059.htm" \l "4314746', '');" \o "Using Other Configuration Files).

**delete factory**

delete factory *factory-name*

Delete the named connection factory.

**delete group**

delete group *group-name*

Delete the named group.

**delete jndiname**

delete jndiname *jndiname*

Delete the named JNDI name. Notice that deleting the last JNDI name of a connection factory object will remove the connection factory object as well.

See [Chapter 13, Using the EMS Implementation of JNDI](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.097.htm" \l "2869809', '');" \o "Using the EMS Implementation of JNDI) for more information.

**delete message**

delete message *messageID*

Delete the message with the specified message ID.

**delete queue**

delete queue *queue-name*

Delete the named queue.

**delete route**

delete route *route-name*

Delete the named route.

**delete rvcmlistener**

delete rvcmlistener *transport\_name* *cm\_name* *subject*

Unregister an RVCM listener with the server so that any messages being held for the specified listener in the RVCM ledger are released. This causes the server to perform the TIBCO Rendezvous call tibrvcmTransport\_RemoveListener.

The parameters are:

|  |  |
| --- | --- |
| • | *transport\_name* — the name of the transport to which this RVCM listener applies. |
| • | *cm\_name* — the name of the RVCM listener to which topic messages are exported. |

|  |  |
| --- | --- |
| • | *subject* — the RVCM subject name that messages are published to. This should be the same name as the topic names that specify the export property. |

For more information, see [tibrvcm.conf](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.059.htm" \l "4315329', '');" \o "Using Other Configuration Files)[on page 265](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.059.htm#4315329', '');) and [Rendezvous Certified Messaging (RVCM) Parameters](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.108.htm#3917275', '');).

**delete topic**

delete topic *topic-name*

Delete the named topic.

**delete user**

delete user *user-name*

Delete the named user.

**disconnect**

disconnect

Disconnect the administrative tool from the server.

**echo**

echo [on|off]

Echo controls the reports that are printed into the standard output. When echo is off the administrative tool only prints errors and the output of queries. When echo is on, the administrative tool report also contains a record of successful command execution.

Choosing the parameter on or off in this command controls echo. If echo is entered in the command line without a parameter, it displays the current echo setting (on or off). This command is used primarily for scripts.

The default setting for echo is on.

**exit**

exit (aliases: quit, q, bye, end)

Exit the administration tool.

The administrator may choose the exit command when there are changes in the configuration have which have not been committed to disk. In this case, the system will prompt the administrator to use the commit command before exiting.

**grant queue**

grant queue *queue-name* user=*name* | group=*name* *permissions*

Grants specified permissions to specified user or group on specified queue. The name following the queue name is first checked to be a group name, then a user name.

Specified permissions are added to any existing permissions. Multiple permissions are separated by commas. Enter **all** in the *permissions* string if you choose to grant all possible user permissions.

User permissions are:

|  |  |
| --- | --- |
| • | receive |
| • | send |

|  |  |
| --- | --- |
| • | browse |

For more information on queue permissions, see [Table 47](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.065.htm" \l "2643244', '');" \o "User Permissions) in [User Permissions](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.065.htm#2643215', '');).

Destination-level administrator permissions can also be granted with this command. The following are administrator permissions for queues.

|  |  |
| --- | --- |
| • | view |
| • | create |

|  |  |
| --- | --- |
| • | delete |
| • | modify |

|  |  |
| --- | --- |
| • | purge |

For more information on destination permissions, see [Destination-Level Permissions](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.062.htm" \l "2692001', '');" \o "Administrator Permissions).

**grant topic**

grant topic *topic-name* user=*name* | group=*name* *permissions*

Grants specified permissions to specified user or group on specified topic. The name following the topic name is first checked to be a group name, then a user name.

Specified permissions are added to any existing permissions. Multiple permissions are separated by commas. Enter **all** in the *permissions* string if you choose to grant all possible permissions.

Topic permissions are:

|  |  |
| --- | --- |
| • | subscribe |
| • | publish |

|  |  |
| --- | --- |
| • | durable |
| • | use\_durable |

For more information on topic permissions, see [Table 48](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.065.htm" \l "2643269', '');" \o "User Permissions) in [User Permissions](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.065.htm#2643215', '');).

Destination-level administrator permissions can also be granted with this command. The following are administrator permissions for topics.

|  |  |
| --- | --- |
| • | view |
| • | create |

|  |  |
| --- | --- |
| • | delete |
| • | modify |

|  |  |
| --- | --- |
| • | purge |

For more information on destination permissions, see [Destination-Level Permissions](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.062.htm" \l "2692001', '');" \o "Administrator Permissions).

**grant admin**

grant admin user=*name* | group=*name* *admin\_permissions*

Grant the named global administrator permissions to the named user or group. For a complete listing of global administrator permissions, see [Global Administrator Permissions](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.062.htm" \l "2691838', '');" \o "Administrator Permissions).

**help**

help (aliases: h, ?)

Display help information.

Enter help commands for a summary of all available commands.

Enter help *command* for help on a specific command.

**info**

info (alias: i)

Shows server name and information about the connected server.

**jaci clear**

jaci clear

Empties the JACI permission cache of all entries.

**jaci resetstats**

jaci resetstats

Resets all statistics counters for the JACI cache to zero.

**jaci showstats**

jaci showstats

Prints statistics about JACI cache performance.

**purge all queues**

purge all queues [*pattern*]

Purge all or selected queues.

When used without the optional pattern parameter, this command erases all messages in all queues for all receivers.

When used with the *pattern* parameter, this command erases all messages in all queues that fit the pattern (for example: foo.\*).

**purge all topics**

purge all topics [*pattern*]

Purge all or selected topics.

When used without the optional pattern parameter, this command erases all messages in all topics for all subscribers.

When used with the *pattern* parameter, this command erases all messages in all topics that fit the pattern (for example: foo.\*).

**purge durable**

purge durable *durable-name*

Purge all messages in the topic for the named durable subscriber

**purge queue**

purge queue *queue-name*

Purge all messages in the named queue.

**purge topic**

purge topic *topic-name*

Purge all messages for all subscribers on the named topic.

**remove member**

remove member *group-name* *user-name*[,*user2*,*user3*,...]

Remove one or more named users from the named group.

**removeprop factory**

removeprop factory *factory-name* *properties*

Remove the named properties from the named factory. See [Connection Factory Parameters](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.059.htm" \l "4314785', '');" \o "Using Other Configuration Files) for a list of properties.

**removeprop queue**

removeprop queue *queue-name properties*

Remove the named properties from the named queue.

**removeprop route**

removeprop route *route-name properties*

Remove the named properties from the named route.

You cannot remove the URL.

You can set the zone\_name and zone\_type parameters when creating a route, but you cannot subsequently change them.

For route parameters, see [Configuring Routes and Zones](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.151.htm" \l "1675257', '');" \o "Configuring Routes and Zones).

For the configuration file routes.conf, see [routes.conf on page 259](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.059.htm#4315040', '');).

**removeprop topic**

removeprop topic *topic-name properties*

Remove the named properties from the named topic.

**resume route**

resume route *route-name*

Resumes sending messages to named route, if messages were previously suspended using the [suspend route](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.054.htm" \l "4579595', '');" \o "Command Listing) command.

**revoke admin**

revoke admin user=*name* | group=*name permissions*

Revoke the specified global administrator permissions from the named user or group. See [Chapter 8, Authentication and Permissions](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.060.htm" \l "2642876', '');" \o "Authentication and Permissions) for more information about administrator permissions.

**revoke queue**

revoke queue *queue-name* user=*name* | group=*name permissions*

revoke queue *queue-name* \* [user | admin | both]

Revoke the specified permissions from a user or group for the named queue.

User and group permissions for queues are receive, send, browse, and all. Administrator permissions for queues are view, create, delete, modify, and purge.

If you specify an asterisk (\*), all user-level permissions on this queue are removed. You can use the optional admin parameter to revoke all administrative permissions, or the both parameter to revoke all user-level and administrative permissions on the queue.

For more information, see [Chapter 8, Authentication and Permissions](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.060.htm" \l "2642876', '');" \o "Authentication and Permissions).

**revoke topic**

revoke topic *topic-name* user=*name* | group=*name permissions*

revoke topic *topic-name* \* [user | admin | both]

Revoke the specified permissions from a user or group for the named topic.

User and group permissions for topics are subscribe, publish, durable, use\_durable, and all. Administrator permissions for topics are view, create, delete, modify, and purge.

If you specify an asterisk (\*), all user-level permissions on this topic are removed. You can use the optional admin parameter to revoke all administrative permissions, or the both parameter to revoke all user-level and administrative permissions on the topic.

For more information, see [Chapter 8, Authentication and Permissions](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.060.htm" \l "2642876', '');" \o "Authentication and Permissions).

**rotatelog**

rotatelog

Force the current log file to be backed up and truncated. The server starts writing entries to the newly empty log file.

The backup file name is the same as the current log file name with a sequence number appended to the filename. The server queries the current log file directory and determines what the highest sequence number is, then chooses the next highest sequence number for the new backup name. For example, if the log file name is tibems.logand there is already a tibems.log.1 and tibems.log.2, the server names the next backup tibems.log.3.

**set password**

set password *user-name* [*password*]

Set the password for the named user.

If you do not supply a password in the command, the server prompts you to type one.

|  |  |
| --- | --- |
| • | To reset a password, type: |

set password *user-name*

Type a new password at the prompt.

|  |  |
| --- | --- |
| • | To remove a password, use this command without supplying a password, and press the **Enter** key at the prompt (without typing a password). |

|  |  |
| --- | --- |
| * | Passwords are a significant point of vulnerability for any enterprise. We recommend enforcing strong standards for passwords.  For security equivalent to single DES (an industry minimum), security experts recommend passwords that contain 8–14 characters, with at least one upper case character, at least one numeric character, and at least one punctuation character. |

**set server**

set server *parameter=value* [*parameter=value* ...]

The set server command can control many parameters. Multiple parameters are separated by spaces. [Table 17](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.054.htm#4309968', '');) describes the parameters you can set with this command.

|  |  |
| --- | --- |
| *Table 17**set server — parameters* | |
| **Parameter** | **Description** |
| password [= *string*] | Sets server password used by the server to connect to other routed servers. If the value is omitted it is prompted for by the administration tool. Entered value will be stored in the main server configuration file in mangled form (but not encrypted).  To reset this password, enter the empty string twice at the prompt. |
| authorization=enabled|disabled | Sets the [authorization](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.058.htm" \l "4567179', '');" \o "tibemsd.conf) mode in the [tibemsd.conf](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.058.htm" \l "4311405', '');" \o "tibemsd.conf) file.  After a transition from disabled to enabled, the server checks ACL permissions for all subsequent requests. While the server requires valid authentication for existing producers and consumers, it does not retroactively reauthenticate them; it denies access to users without valid prior authentication. |
| log\_trace=*trace-items* | Sets the trace preference on the file defined by the logfile parameter. If logfile is not set, the values are stored but have no effect.  The value of this parameter is a comma-separated list of trace options. For a list of trace options and their meanings, see [Table 81, Server Tracing Options](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.124.htm" \l "2519937', '');" \o "Log Files and Tracing).  You may specify trace options in three forms:   |  |  | | --- | --- | | • | plain  A trace option without a prefix character replaces any existing trace options. | | • | +  A trace option preceded by + adds the option to the current set of trace options. |  |  |  | | --- | --- | | • | -  A trace option preceded by -removes the option from the current set of trace options. |   **Examples**  The following example sets the trace log to only show messages about access control violations.  log\_trace=ACL  The next example sets the trace log to show all default trace messages, in addition to SSL messages, but ADMIN messages are not shown.  log\_trace=DEFAULT,-ADMIN,+SSL |
| console\_trace=*console-trace-items* | Sets trace options for output to stderr. The values are the same as for log\_trace. However, console tracing is independent of log file tracing.  If logfile is defined, you can stop console output by specifying:  console\_trace=-DEFAULT  Note that important error messages (and some other messages) are always output, overriding the trace settings.  **Examples**  This example sends a trace message to the console when a TIBCO Rendezvous advisory message arrives.  console\_trace=RVADV |
| client\_trace={enabled|disabled}  [target=*location*] [filter=*value*] | Administrators can trace a connection or group of connections. When this property is enabled, the client generates trace output for opening or closing a connection, message activity, and transaction activity. This type of tracing does not require restarting the client program.  The client sends trace output to *location*, which may be either stderr (the default) or stdout.  You can specify a filter to selectively trace specific connections. The *filter* can be user, connid or clientid. The *value* can be a user name or ID (as appropriate to the filter).  When the filter and value clause is absent, the default behavior is to trace all connections.  Setting this parameter using the administration tool does not change its value in the configuration file [tibemsd.conf](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.058.htm" \l "4311405', '');" \o "tibemsd.conf). |
| max\_msg\_memory=*value* | Maximum memory the server can use for messages.  For a complete description, see max\_msg\_memory in [tibemsd.conf on page 189](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.058.htm#4311405', '');).  Specify units as KB, MB or GB. The minimum value is 8MB. Zero is a special value, indicating no limit.  Lowering this value will not immediately free memory occupied by messages. |
| msg\_swapping=enabled|disabled | Enables or disables the ability to swap messages to disk. |
| track\_message\_ids=enabled|disabled | Enables or disables tracking messages by MessageID. |
| track\_correlation\_ids=enabled|disabled | Enables or disables tracking messages by CorrelationID. |
| ssl\_password[=*string*] | This sets a password for SSL use only.  Sets private key or PKCS#12 file password used by the server to decrypt the content of the server identity file. The password is stored in mangled form. |
| ft\_ssl\_password[=*string*] | This sets a password for SSL use with Fault Tolerance.  Sets private key or PKCS#12 file password used by the server to decrypt the content of the FT identity file. The password is stored in mangled form. |
| server\_rate\_interval=*num* | Sets the interval (in seconds) over which overall server statistics are averaged. This parameter can be set to any positive integer greater than zero.  Overall server statistics are always gathered, so this parameter cannot be set to zero. By default, this parameter is set to 1.  Setting this parameter allows you to average message rates and message size over the specified interval. |
| statistics=enabled|disabled | Enables or disables statistic gathering for producers, consumers, destinations, and routes. By default this parameter is set to disabled.  Disabling statistic gathering resets the total statistics for each object to zero. |
| rate\_interval=*num* | Sets the interval (in seconds) over which statistics for routes, destinations, producers, and consumers are averaged. By default, this parameter is set to 3 seconds. Setting this parameter to zero disables the average calculation. |
| detailed\_statistics=NONE |      PRODUCERS,CONSUMERS,ROUTES | Specifies which objects should have detailed statistic tracking. Detailed statistic tracking is only appropriate for routes, producers that specify no destination, or consumers that specify wildcard destinations. When detailed tracking is enabled, statistics for each destination are kept for the object.  Setting this parameter to NONE disables detailed statistic tracking. You can specify any combination of PRODUCERS, CONSUMERS, or ROUTES to enable tracking for each object. If you specify more than one type of detailed tracking, separate each item with a comma. |
| statistics\_cleanup\_interval=*num* | Specifies how long (in seconds) the server should keep detailed statistics if the destination has no activity. This is useful for controlling the amount of memory used by detailed statistic tracking. When the specified interval is reached, statistics for destinations with no activity are deleted. |
| max\_stat\_memory=*num* | Specifies the maximum amount of memory to use for detailed statistic gathering. If no units are specified, the amount is in bytes, otherwise you can specify the amount using KB, MB, or GB as the units.  Once the maximum memory limit is reached, the server stops collecting detailed statistics. If statistics are deleted and memory becomes available, the server resumes detailed statistic gathering. |

**setprop factory**

setprop factory *factory-name* *properties* ...

Set the properties for a connection factory, overriding any existing properties. Multiple properties are separated by spaces. See [Connection Factory Parameters](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.059.htm" \l "4314785', '');" \o "Using Other Configuration Files) for the list of the properties that can be set for a connection factory.

**setprop queue**

setprop queue *queue-name* *properties*, ...

Set the properties for a queue, overriding any existing properties. Any properties on a queue that are not explicitly specified by this command are removed.

Multiple properties are separated by commas. See [Destination Properties](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.030.htm" \l "3430490', '');" \o "Destination Properties) for the list of the properties that can be set for a queue.

**setprop route**

setprop route *route-name* *properties* ...

Set the properties for a route, overriding any existing properties. Any properties on a route that are not explicitly specified by this command are removed.

You can set the zone\_name and zone\_type parameters when creating a route, but you cannot subsequently change them.

Multiple properties are separated by spaces. For route parameters, see [routes.conf](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.059.htm" \l "4315040', '');" \o "Using Other Configuration Files)[on page 259](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.059.htm#4315040', '');) and [Configuring Routes and Zones](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.151.htm#1675257', '');).

**setprop topic**

setprop topic *topic-name* *properties*

Set topic properties, overriding any existing properties. Any properties on a topic that are not explicitly specified by this command are removed.

Multiple properties are separated by commas. See [Destination Properties](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.030.htm" \l "3430490', '');" \o "Destination Properties) for the list of the properties that can be set for a topic.

**show bridge**

show bridge topic|queue *bridge\_source*

Display information about the configured bridges for the named topic or queue. The *bridge\_source* is the name of the topic or queue established as the source of the bridge.

The following is example output for this command:

Target Name    Type Selector

queue.dest        Q

topic.dest.1      T "urgency in ('high', 'medium')"

topic.dest.2      T

The names of the destinations to which the specified destination has configured bridges are listed in the Target Name column. The type and the message selector (if one is defined) for the bridge are listed in the Type and Selector column.

**show bridges**

show bridges [type=topic|queue] [*pattern*]

Shows a summary of the destination bridges that are currently configured. The type option specifies the type of destination established as the bridge source. For example, show bridges topic shows a summary of configured bridges for all topics that are established as a bridge source. The *pattern* specifies a pattern to match for source destination names. For example show bridges foo.\* returns a summary of configured bridges for all source destinations that match the name foo.\*. The type and *pattern*are optional.

The following is example output for this command:

  Source Name        Queue Targets  Topic Targets

Q queue.source                   1              1

T topic.source                   1              2

Destinations that match the specified pattern and/or type are listed in the Source Name column. The number of bridges to queues for each destination is listed in the Queue Targets column. The number of bridges to topics for each destination is listed in the Topic Targets column.

**show config**

show config

Shows the configuration parameters for the connected server. The output includes:

|  |  |
| --- | --- |
| • | configuration files |
| • | server database |

|  |  |
| --- | --- |
| • | server JVM |
| • | server JDBC database |

|  |  |
| --- | --- |
| • | listen ports |
| • | configuration settings |

|  |  |
| --- | --- |
| • | message tracking |
| • | server tracing parameters |

|  |  |
| --- | --- |
| • | statistics settings |
| • | fault-tolerant setup |

|  |  |
| --- | --- |
| • | external transport setup |
| • | server SSL setup |

**show consumer**

show consumer *consumerID*

Shows details about a specific consumer. The *consumerID* can be obtained from the [show consumers](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.054.htm#4310251', '');) output.

**show consumers**

show consumers [topic=*name* | queue=*name*] [durable] [user=*name*] [connection=*id*] [sort=conn|user|dest|msgs] [full]

Shows information about all consumers or only consumers matching specified filters. Output of the command can be controlled by specifying the sort or full parameter. If the topic or queue parameter is specified, then only consumers on destinations matching specified queue or topic are shown. The user and/or connection parameters show consumers only for the specified user or connection. Note that while the queue browser is open, it appears as a consumer in the EMS server.

The durable parameter shows only durable topic subscribers and queue receivers, but it does not prevent queue consumers to be shown. To see only durable topic consumers, use:

show consumers topic=> durable

The sort parameter sorts the consumers by either connection ID, user name, destination name, or number of pending messages. The full parameter shows all columns listed below and can be as wide as 120-140 characters or wider. Both topic and queue consumers are shown in separate tables, first the topic consumers and then the queue consumers.

|  |  |
| --- | --- |
| * | When connected to an EMS 8.0 or higher server, this command no longer displays offline durable subscribers. In order to see offline durables, use the command [show durables](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.054.htm" \l "4310592', '');" \o "Command Listing) or [show subscriptions](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.054.htm#4771863', '');). |

|  |  |
| --- | --- |
| *Table 18**show consumers — description of output fields* | |
| **Heading** | **Description** |
| Id | Consumer ID. |
| Conn | Consumer's connection ID.  If performed on an EMS 7.x or earlier server, this field displays '-' to indicate a disconnected durable topic subscriber. |
| Sess | Consumer's session ID.  If performed on an EMS 7.x or earlier server, this field displays '-' to indicate a disconnected durable topic subscriber. |
| T | Consumer type character which can be one of:  For topic consumer:   |  |  | | --- | --- | | • | T - non-durable topic subscriber. | | • | D - durable topic subscriber. |  |  |  | | --- | --- | | • | R - system-created durable for a routed topic. | | • | P - proxy subscriber on route's temporary topic. |   For queue consumer:   |  |  | | --- | --- | | • | Q - regular queue receiver. | | • | q - inactive queue receiver. |  |  |  | | --- | --- | | • | P - system-created receiver on global queue for user receiver created in one of routes. | |
| Topic/Queue | Name of the subscription topic or queue. |
| Name | (Topics Only.) Durable or shared subscription name. This column is shown for topic consumers if at least one consumer is a durable or shared consumer. |
| SAS[NMBS] | Description of columns:   |  |  | | --- | --- | | • | S - '+' if consumer's connection started, '-' otherwise. | | • | A - mode of consumer's session, values are: |  |  |  | | --- | --- | | − | N - no acknowledge | | − | A - auto acknowledge |  |  |  | | --- | --- | | − | D - dups\_ok acknowledge | | − | C - client acknowledge |  |  |  | | --- | --- | | − | T - session is transactional | | − | X - XA or MS DTC session |  |  |  | | --- | --- | | − | Z - connection consumer | | • | S - '+' if consumer has a selector, '-' otherwise. |  |  |  | | --- | --- | | • | N - (TOPICS ONLY) '+' if subscriber is "NoLocal." | | • | B - (QUEUES ONLY) '+' if consumer is a queue browser. |  |  |  | | --- | --- | | • | S - (TOPICS ONLY) '+' if this is a shared consumer. | |
| Pre | Prefetch value of the consumer's destination. |
| Pre Dlv | Number of prefetch window messages delivered to consumer |
| Msgs Sent | Current number of messages sent to consumer which are not yet acknowledged by consumer's session. |
| Size Sent | Combined size of unacknowledged messages currently sent to consumer. Value is rounded and shown in bytes, (K)ilobytes, (M)egabytes or (G)igabytes. |
| Pend Msgs | (Topics Only.) Total number of messages pending for the topic consumer. |
| Pend Size | (Topics Only.) Combined size of messages pending for the topic consumer. Value is rounded and shown in bytes, (K)ilobytes, (M)egabytes or (G)igabytes. |
| Uptime | Uptime of the consumer. |
| Last Sent | Approximate time elapsed since last message was sent by the server to the consumer. Value is approximate with precision of 1 second. |
| Last Ackd | Approximate time elapsed since last time a message sent to the consumer was acknowledged by consumer's session. Value is approximate with precision of 1 second. |
| Total Sent | Total number of messages sent to consumer since it was created. This includes resends due to session recover or rollback. |
| Total Acked | Total number of messages sent to the consumer and acknowledged by consumer's session since consumer created. |

**show connections**

show connections [type=q|t|s] [host=*hostname*] [user=*username*] [version] [address] [counts] [full]

Show connections between clients and server. [Table 20](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.054.htm" \l "4310404', '');" \o "Command Listing) describes the output.

The type parameter selects the subset of connections to display as shown in [Table 19](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.054.htm#4310378', '');). The host and user parameters can further narrow the output to only those connections involving a specific host or user. When the version flag is present, the display includes the client’s version number.

If the address parameter is specified, then the IP address is printed in the output table. If the counts parameter is specified, then number of producers, consumers and temporary destinations are printed. Specifying the full parameter prints all of the available information.

|  |  |
| --- | --- |
| *Table 19**show connections — type parameter* | |
| **Type** | **Description** |
| type=q | Show queue connections only. |
| type=t | Show topic connections only. |
| type=s | Show system connections only. |
| absent | Show queue and topic connections, but not system connections. |

|  |  |
| --- | --- |
| *Table 20**show connections — description of output fields* | |
| **Heading** | **Description** |
| L | The type of client. Can be one of the following:   |  |  | | --- | --- | | • | J — Java client | | • | C — C client |  |  |  | | --- | --- | | • | # — C# client | | • | - — unknown system connection | |
| Version | The EMS version of the client. |
| ID | Unique connection ID. Each connection is assigned a unique, numeric ID that can be used to delete the connection. |
| FSXT | Connection type information.  The F column displays whether the connection is fault-tolerant.   |  |  | | --- | --- | | • | - — not a fault-tolerant connection, that is, this connection has no alternative URLs | | • | + — fault-tolerant connection, that is, this connection has alternative URLs |   The S column displays whether the connection uses SSL.   |  |  | | --- | --- | | • | - — connection is not SSL | | • | + — connection is SSL |   The X column displays whether the connection is an XA or MS DTC transaction.   |  |  | | --- | --- | | • | - — connection is not XA or MS DTC | | • | + — connection is either an XA or MS DTC connection |   The T column displays the connection type.   |  |  | | --- | --- | | • | C — generic user connection | | • | T — user TopicConnection |  |  |  | | --- | --- | | • | Q — user QueueConnection | | • | A — administrative connection |  |  |  | | --- | --- | | • | R — system connection to another route server | | • | F — system connection to the fault-tolerant server | |
| S | Connection started status, + if started, - if stopped. |
| IP Address | Shows client IP address.  The address or full parameter must be specified to display this field. |
| Port | The ephemeral port used by the client on the client machine.  The address or full parameter must be specified to display this field. |
| Host | Connection's host name. (If the name is not available, this column displays the host’s IP address.) |
| Address | Connection's IP address.  If you supply the keyword address, then the table includes this column. |
| User | Connection user name. If a user name was not provided when the connection was created, it is assigned the default user name anonymous. |
| ClientID | Client ID of the connection. |
| Sess | Number of sessions on this connection. |
| Prod | Number of producers on this connection.  The counts or full parameter must be specified to display this field. |
| Cons | Number of consumers on this connection.  The counts or full parameter must be specified to display this field. |
| TmpT | Number of temporary topics created by this connection.  The counts or full parameter must be specified to display this field. |
| TmpQ | Number of temporary queues created by this connection.  The counts or full parameter must be specified to display this field. |
| Uncomm | Number of messages in uncommitted transactions on the connection.  The counts or full parameter must be specified to display this field. |
| UncommSize | The combined size, in bytes, of messages in uncommitted transactions on the connection.  The counts or full parameter must be specified to display this field. |
| Uptime | Time that the connection has been in effect. |

**show db**

show db

Print a summary of the server’s databases. Databases are also printed by [show stores](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.054.htm" \l "4311050', '');" \o "Command Listing), the preferred command.

See the [show store](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.054.htm" \l "5215640', '');" \o "Command Listing)[on page 170](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.054.htm#5215640', '');) for details about a specific database.

**show durable**

show durable *durable-name*

Show information about a durable subscriber.

|  |  |
| --- | --- |
| *Table 21**show durable — table Information* | |
| **Heading** | **Description** |
| Durable Subscriber | Fully qualified name of the durable subscriber. This name concatenates the client ID (if any) and the subscription name (separated by a colon). |
| Subscription name | Full name of the durable subscriber. |
| Shared | yes if this is a shared durable subscription, no otherwise. |
| Client ID | Client ID of the subscriber’s connection. |
| Topic | The topic from which the durable subscription receives messages. |
| Type | dynamic—created by a client  static—configured by an administrator |
| Status | online  offline |
| Username | Username of the durable subscriber (that is, of the client’s connection).  If the durable subscriber is currently offline, the value in this column is offline. |
| Consumer ID | This internal ID number is not otherwise available outside the server. |
| No Local | enabled—the subscriber does not receive messages sent from its local connection (that is, the same connection as the subscriber).  disabled—the subscriber receives messages from all connections. |
| Selector | The subscriber receives only those messages that match this selector. |
| Pending Msgs | Number of all messages in the topic. (This count includes the number of delivered messages.) |
| Delivered Msgs | Number of messages in the topic that have been delivered to the durable subscriber, but not yet acknowledged. |
| Pending Msgs Size | Total size of all pending messages |

**show durables**

show durables [*pattern*]

If a pattern is not entered, this command shows a list of all durable subscribers on all topics.

If a pattern is entered (for example foo.\*) this command shows a list of durable subscribers on topics that match that pattern.

This command prints a table of information described in [Table 22](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.054.htm" \l "4310605', '');" \o "Command Listing).

|  |  |
| --- | --- |
| *Table 22**show durables — table Information* | |
| **Heading** | **Description** |
| Topic Name | Name of the topic.  An asterisk preceding this name indicates a dynamic durable subscriber. Otherwise the subscriber is static (configured by an administrator). |
| Durable | Full name of the durable subscriber. |
| Shared | Y to indicate that this is a shared durable subscription, Notherwise. |
| User | Name of the user of this durable subscriber. If the durable subscriber is currently offline, the value in this column is offline. If this is a shared durable subscription, the value of this column is shared.  For users defined externally, there is an asterisk in front of the user name. |
| Msgs | Number of pending messages |
| Size | Total size of pending messages |

For more information, see [Destination Properties](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.030.htm" \l "3430490', '');" \o "Destination Properties).

**show factory**

show factory *factory-name*

Shows properties of specified factory.

**show factories**

show factories [generic|topic|queue]

Shows all factories. You can refine the listed output by specifying only generic, topic, or queue factories be listed.

**show jndiname**

show jndiname *jndi-name*

Shows the object that the specified name is bound to by the JNDI server.

**show jndinames**

show jndinames [*type*]

The optional parameter *type* can be:

|  |  |
| --- | --- |
| • | destination |
| • | topic |

|  |  |
| --- | --- |
| • | queue |
| • | factory |

|  |  |
| --- | --- |
| • | topicConnectionFactory |
| • | queueConnectionFactory |

When type is specified only JNDI names bound to objects of the specified type are shown. When type is not specified, all JNDI names are shown.

**show group**

show group *group-name*

Shows group name, description, and number of members in the group.

For groups defined externally, there is an asterisk in front of the group name. Only external groups with at least one currently connected user are shown.

**show groups**

show groups

Shows all user groups.

For groups defined externally, there is an asterisk in front of the group name.

**show members**

show members *group-name*

Shows all user members of specified user group.

**show message**

show message *messageID*

Shows the message for the specified message id.

This command requires that tracking by message ID be turned on using the track\_message\_ids configuration parameter.

**show messages**

show messages *correlationID*

Shows the message IDs of all messages with the specified correlation ID set as JMSCorrelationID message header field. You can display the message for each ID returned by this command by using the [show message](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.054.htm#4770136', '');) *messageID* command.

This command requires that tracking by correlation ID be turned on using the track\_correlation\_ids configuration parameter.

**show parents**

show parents *user-name*

Shows the user’s parent groups. This command can help you to understand the user’s permissions.

**show queue**

show queue *queue-name*

Shows the details for the specified queue.

|  |  |
| --- | --- |
| * | If the queue is a routed queue, specify only the name of the queue (do not specify the server using the *queue-name*@*server* form). |

|  |  |
| --- | --- |
| *Table 23**show queue — table Information* | |
| **Heading** | **Description** |
| Queue | Full name of the queue. |
| Type | dynamic—created by a client  static—configured by an administrator |
| Properties | A list of property names that are set on the queue, and their values. For an index list of property names, see [Destination Properties](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.030.htm" \l "3430490', '');" \o "Destination Properties). |
| JNDI Names | A list of explicitly assigned JNDI names that refer to this queue. |
| Bridges | A list of bridges from this queue to other destinations. |
| Receivers | Number of consumers on this queue. |
| Pending Msgs | Number of all messages in the queue, followed by the number of persistent messages in parenthesis.  These counts include the number of delivered messages. |
| Delivered Msgs | Number of messages in the queue that have been delivered to a consumer, but not yet acknowledged. |
| Pending Msgs Size | Total size of all pending messages, followed by the size of all persistent messages in parenthesis. |

**show queues**

show queues [*pattern-name* [notemp|static|dynamic] [first=*n*|next=*n*|last=*n*]]

If a *pattern-name* is not entered, this command shows a list of all queues.

If a *pattern-name* is entered (for example foo.\* or foo.>) this command shows a list of queues that match that pattern. See [Wildcards \* and >](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.033.htm#2844610', '');) for more information about using wildcards.

You can further refine the list of queues that match the pattern by using one of the following parameters:

|  |  |
| --- | --- |
| • | notemp — do not show temporary queues |
| • | static — show only static queues |

|  |  |
| --- | --- |
| • | dynamic — show only dynamic queues |

When a *pattern-name* is entered, you can also cursor through the list of queues using one of the following commands, where *n* is whole number:

|  |  |
| --- | --- |
| • | first=*n* — show the first *n* queues |
| • | next=*n* — show the next *n* queues |

|  |  |
| --- | --- |
| • | last=*n* — show the next *n* queues and terminate the cursor |

The cursor examines *n* queues and displays queues that match the *pattern-name*. Because it does not traverse the full list of queues, the cursor may return zero or fewer than *n* queues. To find all matching queues, continue to use next until you receive a Cursor complete message.

The show queues command prints a table of information described in [Table 24](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.054.htm#4787361', '');). A \* appearing before the queue name indicates a dynamic queue.

|  |  |
| --- | --- |
| *Table 24**show queues — table Information* | |
| **Heading** | **Description** |
| Queue Name | Name of the queue. If the name is prefixed with an asterisk (**\***), then the queue is temporary or was created dynamically. Properties of dynamic and temporary queues cannot be changed. |
| SNFGXIBCT | Prints information on the topic properties in the order  (S)ecure (N)sender\_name or sender\_name\_enforced (F)ailsafe (G)lobal e(X)clusive (I)mport (B)ridge (C)flowControl (T)race  The characters in the value section show:  - Property not present  + Property is present, and was set on the topic itself  \* Property is present, and was inherited from another queue  Note that inherited properties cannot be removed. |
| Pre | Prefetch value. If the value is followed by an asterisk (**\***), then it is inherited from another queue or is the default value. |
| Rcvrs | Number of currently active receivers |
| All Msgs | |
| Msgs | Number of pending messages |
| Size | Total size of pending messages |
| Persistent Msgs | |
| Msgs | Number of pending persistent messages |
| Size | Total size of pending persistent messages |

For more information, see [Destination Properties](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.030.htm" \l "3430490', '');" \o "Destination Properties).

**show route**

show route *route-name*

Shows the properties (URL and SSL properties) of a route.

**show routes**

show routes

Shows the properties (URL and SSL properties) of all created routes.

These commands print the information described in [Table 25](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.054.htm" \l "4310844', '');" \o "Command Listing).

|  |  |
| --- | --- |
| *Table 25**show routes — table Information* | |
| **Heading** | **Description** |
| Route | Name of the route. |
| T | Type of route:   |  |  | | --- | --- | | • | A indicates an active route. | | • | P indicates a passive route. | |
| ConnID | Unique ID number of the connection from this server to the server at the other end of the route.  A hyphen (-) in this column indicates that the other server is not connected. |
| URL | URL of the server at the other end of the route. |
| ZoneName | Name of the zone for the route. |
| ZoneType | Type of the zone:   |  |  | | --- | --- | | • | m indicates a multi-hop zone. | | • | 1 indicates a one-hop zone. | |

**show rvcmtransportledger**

show rvcmtransportledger *transport\_name* [*subject-or-wildcard*]

Displays the TIBCO Rendezvous certified messaging (RVCM) ledger file entries for the specified transport and the specified subject. You can specify a subject name, use wildcards to retrieve all matching subjects, or omit the subject name to retrieve all ledger file entries.

For more information about ledger files and the format of ledger file entries, see TIBCO Rendezvous documentation.

**show rvcmlisteners**

show rvcmlisteners

Shows all RVCM listeners that have been created using the create rvcmlistener command or by editing the [tibrvcm.conf](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.059.htm#4315329', '');) file.

**show server**

show server (aliases: info, i)

Shows server name and information about the connected server.

**show stat**

show stat consumers [topic=*name*|queue=*name*] [user=*name*]

          [connection=*id*] [total]

show stat producers [topic=*name*|queue=*name*] [user=*name*]

          [connection=*id*] [total]

show stat route *name* [topic=*name*|queue=*name*] [total] [wide]

show stat topic *name* [total] [wide]

show stat queue *name* [total] [wide]

Displays statistics for the specified item. You can display statistics for consumers, producers, routes, or destinations. Statistic gathering must be enabled for statistics to be displayed. Also, detailed statistics for each item can be displayed if detailed statistic tracking is enabled. Averages for inbound/outbound messages and message size are available if an interval is specified in the rate\_interval configuration parameter.

The total keyword specifies that only total number of messages and total message size for the item should be displayed. The wide keyword displays inbound and outbound message statistics on the same line.

See [Working with Server Statistics](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.127.htm" \l "1862780', '');" \o "Working with Server Statistics) for a complete description of statistics and how to enable/disable statistic gathering options.

|  |  |
| --- | --- |
| * | When connected to an EMS 8.0 or higher server, this command does not return statistics for offline durable subscribers. |

**show state**

show state

Shows the state and a minimal subset of the information about the connected EMS server.

**show store**

show store *store-name*

Show the details of a specific store. This command can be used to get details about either a file-based store or a database store.

The *store-name* must be the exact name of a specific store.

This command prints a table of information described in [Table 26](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.054.htm" \l "4310943', '');" \o "Command Listing).

|  |  |
| --- | --- |
| *Table 26**show store — table Information* | |
| **Heading** | **Description** |
| Store | Name of the store. |
| Type | Type of store:   |  |  | | --- | --- | | • | file indicates a file-based store. | | • | dbstore indicates a database store. |  |  |  | | --- | --- | | • | mstore indicates an mstore. | |
| Message Count | The number of messages that are stored in the file. |
| Swapped Count | The number of messages that have been swapped from process memory to store file. |
| Average Write Time | Average time in seconds a write call takes. (Not available for asynchronous file stores.) |
| Write Usage | The ratio between time spent within write calls and the time specified by the [server\_rate\_interval](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.058.htm" \l "4313801', '');" \o "tibemsd.conf). (Not available for asynchronous file stores.) |
| **Headings specific to file-based stores** | |
| File | File name associated with this store file, as it is set by the file parameter in the [stores.conf](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.059.htm#4315119', '');) file. |
| Access Mode | asynchronous—the server stores messages in the file using asynchronous I/O calls.  synchronous—the server stores messages in the file using synchronous I/O calls. |
| Pre-allocation Minimum | The amount of disk space, if any, that is preallocated to this file. |
| CRC | enabled—the server uses CRC to validate checksum data when reading the store file.  disabled—the server does not validate checksum data when reading the store file. |
| Periodic Truncation | enabled—the EMS server occasionally truncates the store file, relinquishing unused disk space.  disabled—the EMS server does not truncate the store file to relinquish unused disk space. |
| Destination Defrag Batch Size | The size of the batch used by the destination defrag feature. |
| File Size | The size of the store file, including unused allocated file space. |
| Free Space | The amount of unused allocated file space. |
| Fragmentation | The level of fragmentation in the file. |
| Used Space | The amount of used space in the file. |
| Message Size | Total size of all messages in the file. |
| Swapped Size | The total size of swapped messages in the file. |
| Storage Write Rate | The number of bytes written per second. |
| **Headings specific to mstores**  Note that output for mstores includes many of the same fields available to file-based stores. | |
| Access Mode | asynchronous — the server writes messages in the mstore files using asynchronous I/O calls.  synchronous — the server writes messages in the mstore files using synchronous I/O calls. |
| Time-bound compact | available — this mstore can be compacted in a time-bound manner or through the mstore\_truncateproperty.  unavailable — this mstore *cannot* be compacted in a time-bound manner or through the mstore\_truncateproperty. |
| Periodic Truncation | enabled — the server occasionally truncates the mstore files, relinquishing unused disk space.  disabled — the server does not truncate the mstore files to relinquish unused disk space. |
| Discard Scan Interval | The maximum length of time that the EMS server takes to examine all messages in the mstore. This interval is controlled with the scan\_iter\_interval parameter in the [stores.conf](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.059.htm#4315119', '');) file. |
| Discard Scan Interval Bytes | The bytes read and processed every Discard Scan Interval. This number is proportional to the mstore file size, and must be kept within the limits of your storage medium. See [Understanding mstore Intervals](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.019.htm" \l "4608517', '');" \o "Store Messages in Multiple Stores) for more information. |
| First Scan Finished | true—all the data in the store has been examined at least once since the EMS server startup.  false—not all data has been examined since the EMS server last started. When false, certain server statistics (such as the Message Count field) may be underreported as a result of expired or purged messages still in the store. See [Implications for Statistics](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.019.htm#4656600', '');) for more information. |
| Storage Write Rate | The number of bytes written per second. |
| **Headings specific to database stores** | |
| JDBC Driver Name | The name of the JDBC database server. |
| JDBC URL | The location of the JDBC database server. |
| Username | The username that the EMS server uses to access the database. |
| Dialect | The SQL dialect used to construct SQL commands. |

**show stores**

show stores

Print a list of the server’s stores.

**show topic**

show topic *topic-name*

|  |  |
| --- | --- |
| *Table 27**show topic — table Information* | |
| **Heading** | **Description** |
| Topic | Full name of the topic. |
| Type | dynamic—created by a client  static—configured by an administrator |
| Properties | A list of property names that are set on the topic, and their values. For an index list of property names, see [Destination Properties](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.030.htm" \l "3430490', '');" \o "Destination Properties). |
| JNDI Names | A list of explicitly assigned JNDI names that refer to this topic. |
| Bridges | A list of bridges from this topic to other destinations. |
| Subscriptions | Number of subscriptions on this topic. (This count also includes durable subscriptions.) |
| Durable Subscriptions | The number of durable subscriptions on the topic. |
| Consumers | Number of active consumers on this topic.  Note: When a durable consumer is offline, it is not included in the count reported here.  However, if this command is performed on an EMS 7.x or earlier server, the count also includes offline durable consumers. |
| Durable Consumers | Number of active durable consumers on this topic.  Note: When a durable consumer is offline, it is not included in the count reported here.  However, if this command is performed on an EMS 7.x or earlier server, the count also includes offline durable consumers. |
| Pending Msgs | The total number of messages sent but not yet acknowledged by the consumer, followed by the number of persistent messages in parenthesis. These counts include copies sent to multiple subscribers. |
| Pending Msgs Size | Total size of all pending messages, followed by the size of all persistent messages in parenthesis. |
| The server accumulates the following statistics only when the administrator has enabled statistics. Otherwise these items are zero. | |
| Total Inbound Msgs | Cumulative count of all messages delivered to the topic. |
| Total Inbound Bytes | Cumulative total of message size over all messages delivered to the topic. |
| Total Outbound Msgs | Cumulative count of messages consumed from the topic by consumers. Each consumer of a message increments this count independently of other consumers, so one inbound message results in *n* outbound messages (one per consumer). |
| Total Outbound Bytes | Cumulative total of message size over all messages consumed from the topic by consumers. Each consumer of a message contributes this total independently of other consumers. |

**show topics**

show topics [*pattern-name* [notemp|static|dynamic] [first=*n*|next=*n*|last=*n*]]

If a *pattern-name* is not entered, this command shows a list of all topics.

If a *pattern-name* is entered (for example foo.\* or foo.>) this command shows a list of topics that match that pattern. See [Wildcards \* and >](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.033.htm#2844610', '');) for more information about using wildcards.

You can further refine the list of topics that match the pattern by using one of the following parameters:

|  |  |
| --- | --- |
| • | notemp — do not show temporary topics |
| • | static — show only static topics |

|  |  |
| --- | --- |
| • | dynamic — show only dynamic topics |

When a *pattern-name* is entered, you can also cursor through the list of topics using one of the following commands, where *n* is whole number:

|  |  |
| --- | --- |
| • | first=*n* — show the first *n* topics |
| • | next=*n* — show the next *n* topics |

|  |  |
| --- | --- |
| • | last=*n* — show the next *n* topics and terminate the cursor |

The cursor examines *n* topics and displays topics that match the *pattern-name*. Because it does not traverse the full list of topics, the cursor may return zero or fewer than *n*topics. To find all matching topics, continue to use next until you receive a Cursor complete message.

The show topics command prints a table of information described in [Table 28](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.054.htm#4607154', '');).

|  |  |
| --- | --- |
| *Table 28**show topics — table information* | |
| **Heading** | **Description** |
| Topic Name | Name of the topic. If the name is prefixed with an asterisk (**\***), then the topic is temporary or was created dynamically. Properties of dynamic and temporary topics cannot be changed. |
| SNFGEIBCTM | Prints information on the topic properties in the order:  (S)ecure (N)sender\_name or sender\_name\_enforced (F)ailsafe (G)lobal (E)xport (I)mport (B)ridge (C)flowControl (T)race (M)ulticast  The characters in the value section show:  - Property not present  + Property is present, and was set on the topic itself  \* Property is present, and was inherited from another topic  Note that inherited properties cannot be removed. |
| Subs | Number of current subscriptions on the topic, including durable subscriptions.  If this command is performed on an EMS 7.x or earlier server, the count reflects the number of *subscribers*, not the number of subscriptions. |
| Durs | Number of durable subscriptions on the topic.  If this command is performed on an EMS 7.x or earlier server, the count reflects the number of durable *subscribers*, not the number of subscriptions. |
| All Msgs | |
| Msgs | The total number of messages sent but not yet acknowledged by the consumer. This count includes copies sent to multiple subscribers.  To see the count of actual messages (not multiplied by the number of topic subscribers) sent to all destinations, use the [show server](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.054.htm" \l "4310911', '');" \o "Command Listing) command. |
| Size | Total size of pending messages |
| Persistent Msgs | |
| Msgs | The total number of persistent messages sent but not yet acknowledged by the consumer. |
| Size | Total size of pending persistent messages |

For more information, see [Destination Properties](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.030.htm" \l "3430490', '');" \o "Destination Properties).

**show subscriptions**

show subscriptions [topic=*name*] [name=*sub-name*] [shared=only|none] [durable=only|none] [sort=msgs|topic|name|cons|id]

This command prints information about all topic subscriptions, or only subscriptions matching specified filters. Command output is controlled using the sort parameter.

If topic=*name* is specified, then only subscriptions on destinations matching specified topic are shown. If name=*sub-name* is specified, then only subscriptions of that name are shown.

If durable=only is specified, then only durable subscriptions are shown.

If durable=none is specified, then only non-durable subscriptions are shown.

If shared=only is specified, then only shared subscriptions are shown.

If shared=none is specified, then only unshared subscriptions are shown.

The parameter sort allows you to specify how the command output is sorted in the output table. You can use to sort by number of pending messages, topic name, subscription name, number of consumers on that subscription, or the subscription's identifier.

The show subscriptions command prints a table of information described in [Table 29](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.054.htm#5038508', '');).

|  |  |
| --- | --- |
| *Table 29**show subscriptions — table information* | |
| **Heading** | **Description** |
| Id | The ID of the subscription. |
| T | The subscription type:   |  |  | | --- | --- | | • | T — non-durable subscription | | • | D — durable subscription | |
| Topic | Name of the topic associated with the subscription. |
| Name | Name of the subscription (durable or shared name).  If this is an unshared non-durable subscription, this value is empty. |
| SS | Description of columns:   |  |  | | --- | --- | | • | S - '+' if the subscription has a selector, '-' otherwise. | | • | S - '+' if the subscription is shared, '-' otherwise. | |
| Cons Count | The number of active consumers on this subscription.  For an unshared non-durable subscription, the value is always 1.  For a durable subscription, the value can be 0, meaning that there is no active consumer and the subscription is offline. |
| Pend Msgs | Total number of messages pending for the subscription. |
| Pend Size | Combined size of messages pending for the subscription.  Value is rounded and shown in bytes, (K)ilobytes, (M)egabytes or (G)igabytes. |
| Uptime | The length of time, in hours, minutes, and seconds, since the subscription was created. |

**show transaction**

show transaction *XID*

Shows a list of messages that were sent or received within the specified transaction. This command returns information on transactions in prepared, ended, and roll back states only. Transactions in a suspended or active state are not included.

[Table 30](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.054.htm" \l "4864963', '');" \o "Command Listing) describes the information shown in each column.

|  |  |
| --- | --- |
| *Table 30**show transactions — table information* | |
| **Heading** | **Description** |
| State | Transaction state:   |  |  | | --- | --- | | • | **A** active | | • | **E** ended |  |  |  | | --- | --- | | • | **R** rollback only | | • | **P** prepared |  |  |  | | --- | --- | | • | **S** suspended |   Suspended transactions can be rolled back, but cannot be rolled forward (committed). |
| Remaining time before timeout | The seconds remaining before the TX timeout is reached. For example, 3 sec.  This field is only applicable for transactions in State ENDSUCCESS or ROLLBACKONLY. |
| **Messages to be consumed** | |
| Message ID | The message ID of the message. null indicates the message ID could not be obtained or was disabled. If [track\_message\_ids](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.058.htm#4313567', '');) is not enabled, this field displays Disabled. |
| Type | The destination type to which the message was sent:   |  |  | | --- | --- | | • | **Q** queue | | • | **T** topic | |
| Destination | The destination name to which the message was sent. null indicates that destination could not be found. |
| Consumer ID | The consumer ID of the Consumer that is consuming the message. Zero indicates that the consumer is offline. |
| **Messages to be produced** | |
| Message ID | The message ID of the message. null indicates the message ID could not be obtained or was disabled. If [track\_message\_ids](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.058.htm#4313567', '');) is not enabled, this field displays Disabled. |
| Type | The destination type to which the message was sent:   |  |  | | --- | --- | | • | **Q** queue | | • | **T** topic | |
| Destination | The destination name to which the message was sent. null indicates that destination could not be found. |
| JMSTimestamp | The timestamp indicating the time at which the message was created. |

**show transactions**

show transactions

Shows the XID for all client transactions that were created using the XA or MS DTC interfaces. Each row presents information about one transaction. The XID is the concatenation of the Format ID, GTrid Len, Bqual Len, and Data fields for a transaction. For example, if show transactions returns the row:

State Format ID GTrid Len Bqual Len Data

  E      0          6          2        branchid

then the XID is 0 6 2 branchid. Note that the spaces are required.

[Table 31](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.054.htm" \l "4311203', '');" \o "Command Listing) describes the information shown in each column.

|  |  |
| --- | --- |
| *Table 31**show transactions — table information* | |
| **Heading** | **Description** |
| State | Transaction state:   |  |  | | --- | --- | | • | **A** active | | • | **E** ended |  |  |  | | --- | --- | | • | **R** rollback only | | • | **P** prepared |  |  |  | | --- | --- | | • | **S** suspended |   Suspended transactions can be rolled back, but cannot be rolled forward (committed). |
| Format ID | The XA transaction format identifier.  0 = OSI CCR naming is used  >0 = some other format is used  -1 = NULL |
| GTrid Len | The number of bytes that constitute the global transaction ID. |
| Bqual Len | The number of bytes that constitute the branch qualifier. |
| Data | The global transaction identifier (gtrid) and the branch qualifier (bqual). |

**show transport**

show transport *transport*

Displays the configuration for the specified transport defined in [transports.conf](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.059.htm" \l "4315393', '');" \o "Using Other Configuration Files).

See [Configuring EMS Transports for TIBCO FTL](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.102.htm" \l "5297050', '');" \o "Configuring EMS Transports for TIBCO FTL), [Configuring Transports for Rendezvous](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.108.htm#2338741', '');), and [Configuring Transports for SmartSockets](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.117.htm#2187558', '');) for details.

**show transports**

show transports

Lists all configured transport names in [transports.conf](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.059.htm" \l "4315393', '');" \o "Using Other Configuration Files).

**show user**

show user *user-name*

Shows user name and description. If no user name is specified, this command displays the currently logged in user.

For users defined externally, there is an asterisk in front of the user name.

**show users**

show users

Shows all users.

For users defined externally, there is an asterisk in front of the user name. Only currently connected external users are shown.

**showacl admin**

showacl admin

Shows all administrative permissions for all users and groups, but does not include administrative permissions on destinations.

**showacl group**

showacl group *group-name* [admin]

Shows all permissions set for a given group. Shows the group and the set of permissions. You can optionally specify admin to show only the administrative permissions for destinations or principals. Specifying showacl admin shows all administrative permissions for all users and groups (not including administrative permissions on destinations).

**showacl queue**

showacl queue *queue-name* [admin]

Shows all permissions set for a queue. Lists all entries from the acl file. Each entry shows the “grantee” (user or group) and the set of permissions. You can optionally specify admin to show only the administrative permissions for destinations or principals. Specifying showacl admin shows all administrative permissions for all users and groups (not including administrative permissions on destinations).

**showacl topic**

showacl topic *topic-name* [admin]

Shows all permissions set for a topic. Lists all entries from the acl file. Each entry shows the “grantee” (user or group) and the set of permissions. You can optionally specify admin to show only the administrative permissions for destinations or principals. Specifying showacl admin shows all administrative permissions for all users and groups (not including administrative permissions on destinations).

**showacl user**

showacl user *user-name* [admin | all | admin-all]

Shows the user and the set of permissions granted to the user for destinations and principals.

showacl user *username* — displays permissions granted directly to the user. (An administrator can use this form of the command to view own permissions, even without permissions to view any other user permissions.)

showacl user *username* admin — displays administrative permissions granted directly to the user.

showacl user *username* all — displays direct and inherited (from groups to which the user belongs) permissions.

showacl user *username* admin-all — displays all administrative permissions for a given user (direct and inherited)

|  |  |
| --- | --- |
| * | The output from this command displays inherited permissions prefixed with a '\*'. Inherited permissions cannot be changed. An attempt to revoke an inherited permission for the principal user will not change the permission. |

**shutdown**

shutdown

Shuts down currently connected server.

**suspend route**

suspend route *route-name*

Suspends outgoing messages to the named route.

Message flow can be recovered later using the command [resume route](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.054.htm" \l "4579421', '');" \o "Command Listing).

**time**

time [on | off]

Specifying on places a timestamp before each command’s output. By default, the timestamp is off.

**timeout**

timeout [*seconds*]

Show or change the current command timeout value. The timeout value is the number of seconds the Administration Tool will wait for a response from the server after sending a command.

By default, the timeout is 30 seconds. When timeout is entered with the optional *seconds* parameter, the timeout value is reset to the specified number of seconds. When entered without parameter, the current timeout value is returned.

**transaction commit**

transaction commit *XID*

Commits the transaction identified by the transaction ID. The transaction must be in the ended or prepared state. To obtain a transaction ID, issue the show transactionscommand, and cut and paste the XID into this command.

**transaction rollback**

transaction rollback *XID*

Rolls back the transaction identified by the transaction ID. The transaction must be in the ended, rollback only, or the prepared state. To obtain a transaction ID, issue the show transactions command, and cut and paste the XID into this command.

|  |  |
| --- | --- |
| * | Messages sent to a queue with [prefetch](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.030.htm" \l "3471588', '');" \o "Destination Properties)=none and [maxRedelivery](javascript:WWHClickedPopup('tib_ems_usr_guide',%20'EMS.5.030.htm" \l "3459791', '');" \o "Destination Properties)=*number*properties are not received *number* times by an EMS application that receives in a loop and does an XA rollback after the XA prepare phase. |

**updatecrl**

updatecrl

Immediately update the server’s certificate revocation list (CRL).

**whoami**

whoami

Alias for the show user command to display the currently logged in user.

|  |
| --- |
| Copyright © TIBCO Software Inc. All Rights Reserved |
| Copyright © TIBCO Software Inc. All Rights Reserved |

|  |  |
| --- | --- |
| *Table 47**Queue Permission* | |
| **Name** | **Description** |
| receive | permission to create queue receivers |
| send | permission to create queue senders |
| browse | permission to create queue browsers |

|  |  |
| --- | --- |
| *Table 48**Topic Permission* | |
| **Name** | **Description** |
| subscribe | permission to create non-durable subscribers on the topic |
| publish | permission to publish on the topic |
| durable | permission to create, delete, or modify durable subscribers on the topic |
| use\_durable | permission to use an existing durable subscriber on the topic, but *not* to create, delete, or modify the durable subscriber |

**Central Administration**

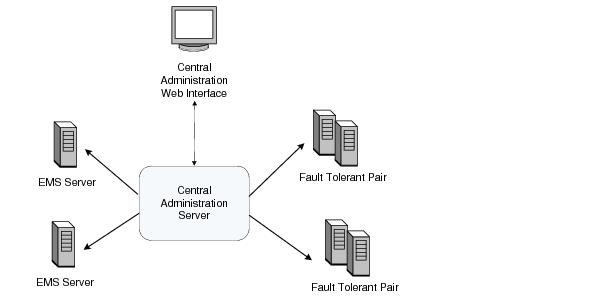
**Structure**

Central Administration offers a simple architecture. The Central Administration server connects to TIBCO Enterprise Message Service servers and stores a snapshot of the configuration from the running EMS server.

|  |  |
| --- | --- |
|  | Be aware that the snapshot of the EMS server may not reflect its current running configuration. Because it’s possible to modify the running EMS server with direct edits to the JSON configuration file, programming API calls, or commands issued through the administration tool, Central Administration may be out-of-sync with the running configuration. |

Administrators connect to the Central Administration server through the web interface, and from there can view the snapshot server configurations, make changes, and deploy the new configurations.

*Figure 1**Central Administration Structure*



**How Configuration Changes are Saved and Deployed**

The Central Administration server stores configuration files for each EMS server it manages in two directories:

|  |  |
| --- | --- |
| • | The **working directory**stores the last read configuration from the EMS server. It also stores the **lock file**, which contains the edits being made by a user to a server configuration. |
| • | The **deployment directory** contains details on each deployment of the EMS server. This directory contains only records of configuration files that have already been deployed using Central Administration. |

All commands issued through the Central Administration web interface modify the server configuration files in these two directories. [Figure 2](javascript:WWHClickedPopup('tib_ems_ca',%20'config.5.03.htm" \l "1723215', '');" \o "Structure) shows which files are modified for each action performed through the Central Administration web interface:

|  |  |
| --- | --- |
| • | **Lock & Edit**  enables the user to make changes to the configuration. As the user edits the configuration, the edits are saved in the lock file. |
| • | **Save**  saves configuration changes made to the EMS server through the web interface to the lock file. Note that changes are also automatically saved at regular intervals, even if the user does not click Save. |

|  |  |
| --- | --- |
| • | **Deploy**  takes the configuration lock file from the working directory, and deploys it to the EMS server. If the deployment succeeds, the lock file is copied to the deployment directory for the current deployment and to the working file for the EMS server. |

If deployment fails, the lock file is not copied to the deployment directory. It remains in the working directory and can be further edited and redeployed, or the lock can be reverted and changes discarded.

Note that the EMS server keeps a copy of the previous JSON configuration file upon receiving a deployment from the Central Administration server. When the deployment is successful, the previous configuration is kept in a file of the same name as the current configuration with an additional .bak suffix. For example, tibemsd.json.bak. During subsequent deployments, the backup file is overwritten.

|  |  |
| --- | --- |
| • | **Refresh**  causes the Central Administration server to retrieve the currently deployed configuration file from the running EMS server, and save that file in the working directory. This is the only way to obtain configuration changes made directly to the EMS server. (That is, changes that were not made through Central Administration.) |

*Figure 2**Central Administration — Flow of Information*

**Start the Central Administration Server**

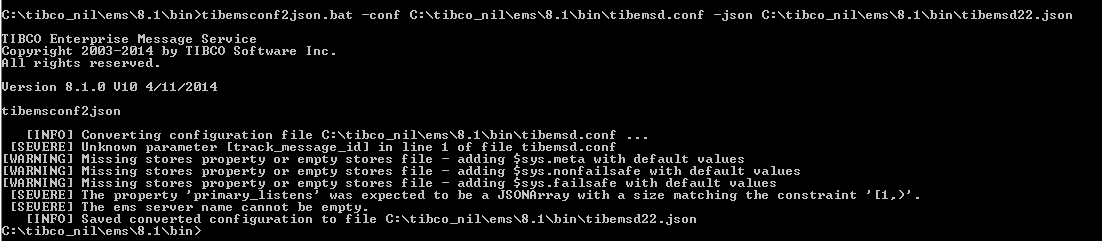
To start the Central Administration server from the command line, navigate to *EMS\_HOME*/bin and run the script:

**On UNIX**

tibemsca [*options*]

**On Windows**

tibemsca.bat [*options*]



**Routes Concepts: Chapter 20**

create route *name* url=*URL* zone\_name=*zone\_name* zone\_type=1hop|mhop *properties*

create route B url=tcp://B:7454 zone\_name=Z1 zone\_type=1hop

 create route C url=tcp://C:7455 zone\_name=ZM zone\_type=mhop

setprop route USA    outgoing\_topic="orders" selector="country=’USA’"

**Create file - create directory or file**

**Read file - read file data**

**copy file - copy files and directories**

**Rename file - rename empty directory or file. then movie files not directory**

**File Pollar -awaiting for file change(moldification,deletion**