

Chapter 3

Viewing Administrative Commands

This chapter describes the administrative commands used to control the Calligo E-Delivery components.

What is in this Chapter?

“What are Administrative Commands?” on page 32

“Types of Administrative Commands” on page 33

“JMS Message Structure” on page 35

“Calligo E-Delivery Component States” on page 36

“Transitioning between States” on page 37

“Calligo E-Delivery Level Commands” on page 40

“Queue Agent Commands” on page 44

“Worker Agent Commands” on page 55

“Service Manager Commands” on page 60

“Request Log Commands” on page 66

What are Administrative Commands?

Each of Calligo E-Delivery's components — Queue Agent, Service Manager, Worker Agent, Request Log Agent — can be issued administrative commands that control how they function. Components can accept commands that allow them to

- initialize and start.
- stop and shut down.
- be queried for parameters and operations specific to their functionality.
- issue notifications about changes in their operational state.

Before an administrative command can be issued, however, the components must first be launched and initialized.

All components with the exception of Service Manager must be launched manually. Alternatively, a Unix shell script or Windows NT batch file could be written that would automatically run the applicable components.

Calligo E-Delivery Components

The following components must be launched before any administration commands can be processed:

- Admin Agent
- Queue Agents
- Worker Agent
- Request Log Agent

See “Chapter 1 - E-Delivery Architecture” for a detailed description of the Calligo E-Delivery Components.

Admin Commands and Admin Response Queue

- All of the Administrative Commands are submitted to an MQSeries “Admin Commands Queue” and retrieved from that queue by the Calligo E-Delivery components to which they were addressed.
- After an Administrative Command has been executed, a corresponding response message is passed back to the Admin Console via the “Admin Response” queue.

Types of Administrative Commands

Administrative commands can be sent to Calligo E-Delivery components by addressing the command

- to all components using Admin Agent (Calligo E-Delivery-level commands).
- to each component individually (component - level commands).

Each of the above are described in the section that follows.

Issuing Calligo E-Delivery Level Commands

These commands are used to control the Calligo E-Delivery components on a system-wide basis. The commands are accepted by the Admin Agent and consist of the following:

- **DistributorInit** used to initialize all components.
- **DistributorStart** used to start all components.
- **DistributorStop** used to stop processing service requests in all components.
- **DistributorShutDown** used to shutdown all components.

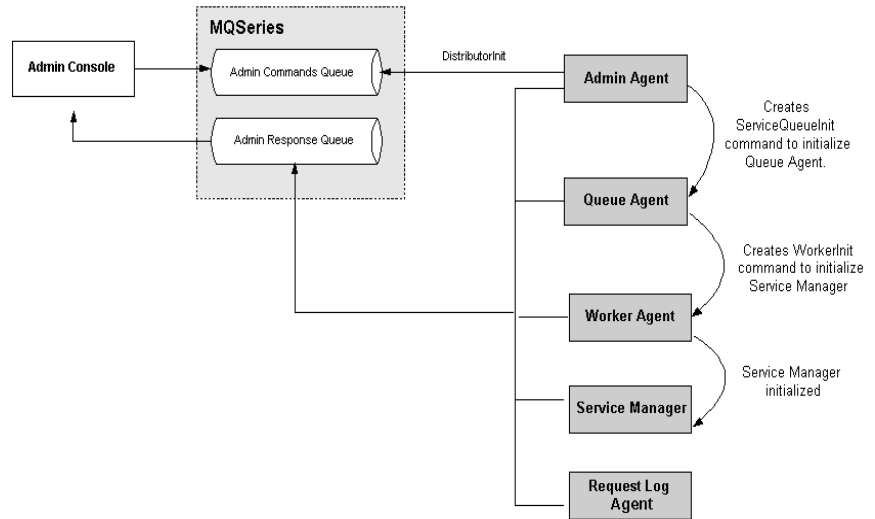
See “*Calligo E-Delivery Level Commands*” on page 40 for a more detailed description.

The Calligo E-Delivery Level commands are issued in a hierarchical manner, whereby a command sent to an upper level component will invoke a command to the next lower component, and so on, until all components have invoked the command.

For example, if a **DistributorInit** command is issued at the Calligo E-Delivery level then the following will occur:

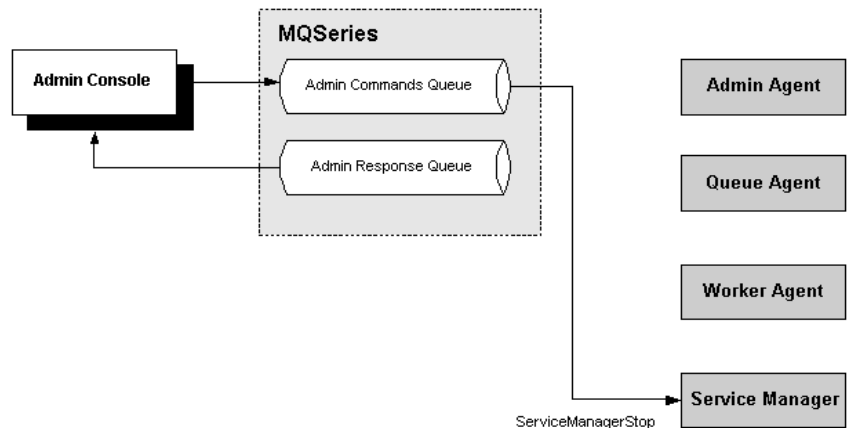
1. The **DistributorInit** command is issued to the Admin Agent.
2. The Admin Agent will create a **ServiceQueueInit** command to initialize the Queue Agent.
3. The Admin Agent will also create a **RequestLogInit** command to initialize Request Log.
4. The Queue Agent will create a **WorkerInit** command which will initialize the Service Managers.

The following diagram illustrates how a Calligo E-Delivery Level command invokes a hierarchy of commands to lower-level components.



Issuing Component Level Commands

These commands can be sent directly through the Admin Commands Queue to a Calligo E-Delivery component. For example, you could request a specific Service Manager to stop fetching and processing service requests by issuing it a **ServiceManagerStop** command.



JMS Message Structure

Administrative commands are delivered to the Calligo E-Delivery components as JMS text messages. The actual command is contained in the body of the JMS message in XML format. See “Chapter 1- E-Delivery Architecture” for a detailed description of JMS and XML messaging.

The table below shows the attributes for the JMSType field used by Calligo E-Delivery.

Attributes	Description
JMSMessageID	A string ID that uniquely identifies the message in the systems (set automatically by the Queue Provider).
JMSCorrelationID	Used only for response messages it contains the JMSMessageID of the request to which the response message is linked.
JMSReplyTo	A destination-object (JMS) indicating the queue where the response to the command should be submitted. If null, no response is required. The queue specified in the JMSReplyTo field must reside on the same MQSeries Queue Agent as the queue used to submit the request.
Component Name	The name of the component to which the command is directly addressed. It is specified as per the JMX specification (see JMX Object Name) and it takes the following form: <code>domain_name:type=component_type[,attr1=value1[,]]</code> This field is null for broadcast commands.
ServiceQueue	This field is used for broadcast messages sent to those Service Managers that listen to a particular Queue Agent. Service Managers will filter out messages if the header field specifies a queue other than the ones to which they are listening. Service Managers will still pick up messages when this field is null.

Calligo E-Delivery Component States

Each of the Calligo E-Delivery components will go through the states outlined below.

Launched

The hardware and software infrastructure required by the component is active but off-line. The component initializes to a default configuration. The only administrative commands the component can accept in this state is Initialize.

Initialized

The component is on-line and has been configured to a known initial state. It will not become operational until it is commanded to start. In this state the component can

- accept other Initialize commands
- be started or shutdown

Started

The component is on-line, configured and operational. The only administrative command the component can accept in this state is Stop. The component cannot be further configured (Initialized) or Shutdown without an explicit Stop.

Stopped

The component is on-line and configured but not operational. In this state, the component can be initialized, Started or Shutdown. This state is equivalent to the Initialized state.

Shutdown

The component is off-line. For the Queue Agent and Worker components which have to be online all the time this command will require an explicit Initialize command for the component to go back online.

For components that can effectively shut down such as Service Managers, this command causes the process in which the component is run to shut down.

Transitioning between States

Calligo E-Delivery

1. Launched manually in off-line mode.
2. Initialized in on-line mode by **DistributorInit**.
3. Started in on-line mode by **DistributorStart**.
4. Stopped in off-line mode by **DistributorStop**. The status of the system will be the same as after the DistributorInit command.
5. Shut down in off-line mode by **Distributor ShutDown**.

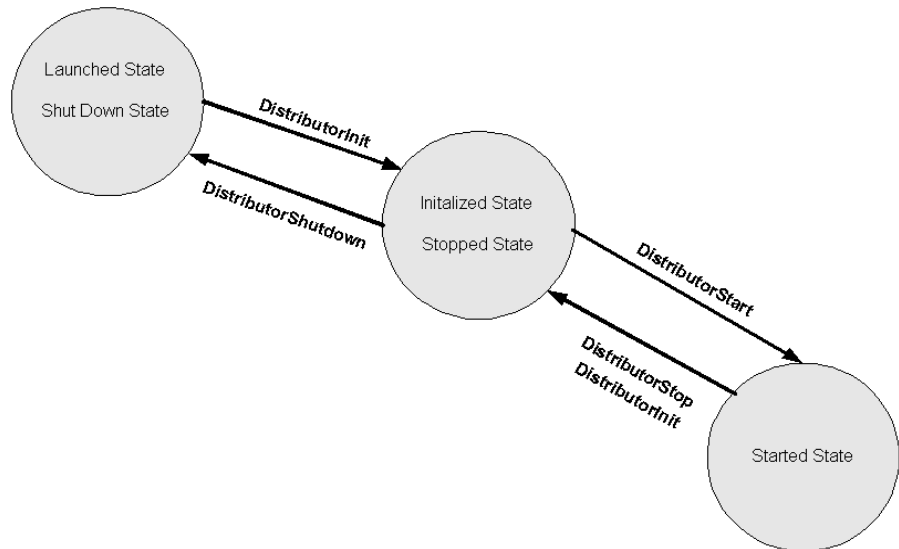


Figure 1.0: Calligo E-Delivery Transition States

Queue Agent

The Queue Agent component should always be on-line. The state of Queue Agent cannot be changed by an Administrative Command.

Worker Agent

The Worker Agent component should always be on-line. The state of the Worker cannot be changed by an Administrative command.

Request Log Agent

1. Launched in off-line mode by **RequestLogInit**.
2. Initialized in off-line mode by **RequestLogInit**.
3. Started in on-line mode by **RequestLogStart**.
4. Stopped in off-line mode by **RequestLogStop**.
5. Shut down in off-line mode by **RequestLogShutDown**.

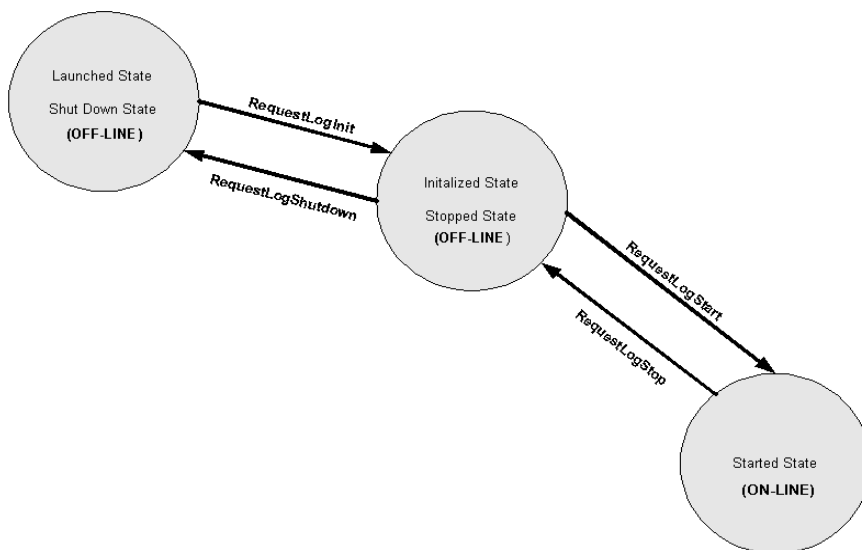


Figure 2.0: Request Log Transition States

Service Manager

1. Launched and initialized in off-line mode by **WorkerInit** or **ServiceQueueInit**.
The component can be re configured using **ServiceQueueInit**, **WorkerInit** or **ServiceManagerInit**.
2. Started in on-line mode using **ServiceQueueStart**, **WorkerStart** or **ServiceManagerStart**.
3. Stopped in off-line mode using **ServiceQueueStop**, **WorkerStop** or **ServiceManagerStop**.
4. Shut down in off-line mode using **ServiceQueueShutDown**, **WorkerShutDown** or **ServiceManagerShutDown**.

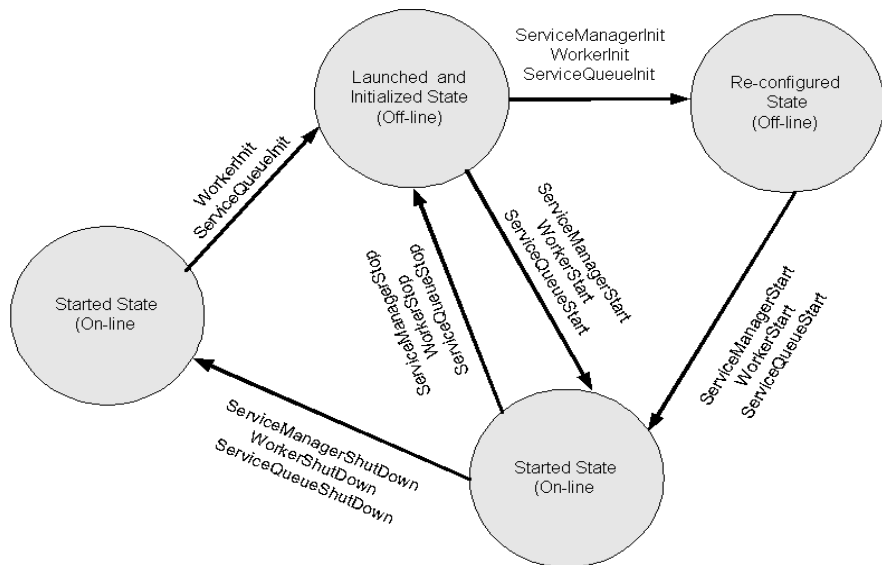


Figure 3.0: Service Manager Transition States

Calligo E-Delivery Level Commands

This section describes the commands that are used to control the running of all Calligo E-Delivery components. The commands are addressed directly to the Admin Agent which is responsible for processing system-wide administrative commands.

Initializing Calligo E-Delivery

The **DistributorInit** command will initialize all Calligo E-Delivery components without actually starting any service request processing.

The purpose of this command is to ensure that all components are initialized and operational.

DistributorInit Command Attributes

Parameters	Description
Configuration	Contains the XML configuration for the entire Calligo E-Delivery system including a description of all components.
Timeout	The period in milliseconds to wait before considering the operation a failure. If the value is zero, there is no timeout.
Return	The operation is considered successful if all the components initialized successfully. If at least one component failed to initialize, then it will <ul style="list-style-type: none">• report the component(s) that failed to initialize.• not revert the configuration for the components that initialized successfully.

Previous Administrative Commands Required:

None

Administrative Commands Accepted AFTER this Command:

DistributorStart, DistributorShutDown, DistributorInit

Starting Calligo E-Delivery

The **DistributorStart** command is used to start the following Calligo E-Delivery components:

- Queue Agent
- Worker Agent
- Service Managers
- Request Log Agent

Once started, the above components will begin processing service requests from their assigned Service Queue(s)

Note: This command will ignore any components that have been issued a direct Start command such as RequestLogStart, **ServiceQueueStart, etc.**

DistributorStart Command Attributes

Parameters	Description
Service Queues	Specifies the Service Queues to be started. If this argument is left empty all queues will be started.
Timeout	The period in milliseconds to wait before considering the operation a failure. If the value is zero, there is no timeout.
Return	The operation is considered successful if the Calligo E-Delivery components are started. If at least one component failed to start, the operation is considered to have failed.

Previous Administrative Commands Required:

DistributorInit or Distributor Stop

Administrative Commands Accepted AFTER this Command:

DistributorStop, DistributorInit

Stopping Service Request Processing in Calligo E-Delivery

The **DistributorStop** command is used to stop processing service requests in all of the Calligo E-Delivery components.

When this command is sent, no current operation is aborted, instead, the Admin Agent will wait for all services to stop.

DistributorStop Attributes

Parameters	Description
ServiceQueues	Specifies the Service Queues to be effectively stopped. If this argument is left empty all queues will be stopped.
Timeout	The period in milliseconds to wait before considering the operation a failure. If the value is zero there is no timeout.
Return	The operation is successful if all components stopped in the given time. If any component fails to stop in the set amount of time, the operation is considered to have failed. The components that have already stopped are not restarted.

Previous Administrative Commands Required:

DistributorStart

Administrative Commands Accepted AFTER this Command:

DistributorShutdown, DistributorStart, DistributorInit

Shutting Down Calligo E-Delivery

The **DistributorShutdown** command is used to shut down the following Calligo E-Delivery components.

- Queue Agent
- Worker Agent
- Service Managers
- Request Log Agent

Note: Some components never shut down. However, the **DistributorShutdown** command will ensure that they will no longer accept any Administrative command other than a direct component initialize command, such as **ServiceManagerInit**, etc.

DistributorShutdown Attributes

Parameters	Description
Timeout	The period in milliseconds to wait before considering the operation a failure. If the value is zero, there is no timeout.
Return	The operation is successful if all required components shut down successfully

Previous Administrative Commands Required:

DistributorInit, Distributor Stop

Administrative Commands Accepted AFTER this Command:

DistributorInit, WorkerInit, ServiceManagerInit, RequestLogInit

Queue Agent Commands

You can use the Queue Agent commands to manage the service requests in the Service Queues and to administer the Service Managers that process messages from those Service Queues.

Initializing Service Managers

The **ServiceQueueInit** command will perform the following functions:

- Initialize all Service Managers attached to a Service Queue.
- Notifies Worker Agents to start the Service Manager processes.

Note: This command does not allow the Service Managers to start fetching requests from the Service Queue. See “*Fetching Request Messages from the Service Queue*” on page 45.

ServiceQueueInit Attributes

Parameters	Description
Configuration	An XML configuration that lists all of the Service Managers (and their configurations) that listen to this Service Queue. It also specifies the queue to be used for the Request Log.
Timeout	The period in milliseconds to wait before considering the operation a failure. If the value is zero there is no timeout.
Return	The operation is successful if all Service Managers initialized successfully, failure otherwise. In case of failure it will report back the names of the Service Managers that failed to initialize and the reason, if available. Service Managers that initialized successfully are not rolled back to the previous configuration.

Previous Administrative Commands Required:

None

Administrative Commands Accepted AFTER this Command:

ServiceQueueStart, ServiceQueueShutdown, ServiceQueueInfo, ServiceQueueBrowseRequests, ServiceQueueChangeRequestPriority, ServiceQueueReQueueRequest, ServiceQueuePauseRequest, ServiceQueueResumeRequest, ServiceQueueCancelRequest

Fetching Request Messages from the Service Queue

The **ServiceQueueStart** command will request all Service Managers to fetch and process requests from the Service Queues to which they are assigned.

ServiceQueueStart Attributes

Parameters	Description
Timeout	The period in milliseconds to wait before considering the operation a failure. There is no timeout if the value is zero.
Return	<p>The operation is successful if all Service Managers started, failure otherwise.</p> <p>In case of failure, it will report back the Service Managers that didn't start and the reason, if available.</p> <p>Service Managers that started successfully are not rolled back to the previous configuration.</p>

Previous Administrative Commands Required:

ServiceQueueInit or ServiceQueueStop

Administrative Commands Accepted AFTER this Command:

ServiceQueueStop, ServiceQueueShutDown, ServiceQueueInfo, ServiceQueueBrowseRequests, ServiceQueueChangeRequestPriority, ServiceQueueReQueueRequest, ServiceQueuePauseRequest, ServiceQueueResumeRequest, ServiceQueueCancelRequest

Stopping Service Managers

The **ServiceQueueStop** command will request all Service Managers to stop listening and processing requests from this queue. If a Service Manager is currently processing a request, it will first finish processing and stop gracefully.

ServiceQueueStop Attributes

Parameters	Description
Timeout	The period in milliseconds to wait before considering the operation a failure. There is no timeout, if the value is zero.
Return	The operation is successful if all the Service Managers stopped, failure otherwise. In case of failure, it will report the Service Managers that failed to stop. The reason for failure is also reported, if available. Service Managers that were already stopped will not be restarted.

Previous Administrative Commands Required:

ServiceQueueStart

Administrative Commands Accepted AFTER this Command:

ServiceQueueStart, ServiceQueueShutDown, ServiceQueueInfo
ServiceQueueBrowseRequests, ServiceQueueChangeRequestPriority
ServiceQueueReQueueRequest, ServiceQueuePauseRequest,
ServiceQueueResumeRequest, ServiceQueueCancelRequest

Shutting Down Service Managers

The **ServiceQueueShutDown** command will request all Workers to shut down all Service Managers that are attached to this queue.

The processes that the Service Managers are running are shut down and it will require another initialization command to bring them up again.

ServiceQueueStop Attributes

Parameters	Description
Timeout	The period in milliseconds to wait before considering the operation a failure. There is no timeout, if the value is zero.
Return	<p>The operation is successful if all the Service Managers stopped, failure otherwise.</p> <p>In case of failure, it will report the Service Managers that failed to stop. The reason for failure is also reported, if available.</p> <p>Service Managers that were already stopped will not be restarted.</p>

Previous Administrative Commands Required:

ServiceQueueInit, ServiceQueueStop

Administrative Commands Accepted AFTER this Command:

ServiceQueueInit

Viewing Service Queue Runtime Performance

The **ServiceQueueInfo** command will provide the following Service Queue runtime information:

- Name
- Current Depth
- Service Managers processing requests off this queue.

ServiceQueueInfo Attributes

Parameters	Description
Timeout	The period in milliseconds to wait before considering the operation a failure. There is no timeout if the value is zero.
Return	The operation is successful if information about the Service Queues is returned. If not, the reason for failure will be provided, if available.

Previous Administrative Commands Required:

ServiceQueueInit, ServiceQueueStart, ServiceQueueStop

Administrative Commands Accepted AFTER this Command:

Depends on the Service Queue status (Initialized, Started or Stopped) when the command is issued.

ServiceQueueInfo, ServiceQueueBrowseRequests, ServiceQueueChangeRequestPriority, ServiceQueueReQueueRequest, ServiceQueuePauseRequest, ServiceQueueResumeRequest, ServiceQueueCancelRequest

Obtaining a Service Request Count

The **ServiceQueueBrowseRequest** command will allow you to use a selector to enumerate the number of service requests that are currently in the Service Queue.

ServiceQueueBrowseRequest Attributes

Parameters	Description
Selector	Used to filter the enumerated requests. The selector consists of name value pairs that have to match message header properties of the request. No filtering is possible based on service request contents. See “Appendix D - Using Selectors” for a more detailed description.
Request Count Limit	Limits the number of requests returned. A system level maximum can be configured. If the number of requests that match the selector exceeds the request count limit (or the system setting, whichever is lower) a warning is returned stating that there are actually more requests than returned.
Timeout	The period in milliseconds to wait before considering the operation a failure. There is no timeout if the value is zero.
Return	The operation is successful, if it returns a list of service requests presently in the Service Queue. The list will contain blocks of metadata that identify each request, including the JMS request ID. Use the RequestLogRequestInfo command if you wish to obtain additional request information. See page 72.

Previous Administrative Commands Required:

ServiceQueueInit, ServiceQueueStart, ServiceQueueStop

Administrative Commands Accepted AFTER this Command:

Depends on the Service Queue status (Initialized, Started or Stopped) when the command is issued.

ServiceQueuePauseRequest, ServiceQueueResumeRequest, ServiceQueueCancelRequest

Changing the Priority of Service Requests in a Queue

The **ServiceQueueChangeRequestPriority** command can be used to change the priority of one or more service requests in a Service Queue.

ServiceQueueChangeRequestPriority Attributes

Parameters	Description
Selector	Used to identify the applicable service request(s). The selector consists of name value pairs that have to match message header properties of the request. No filtering is possible based on service request contents.
Timeout	The period in milliseconds to wait before considering the operation a failure. there is no timeout.
Priority	The new request priority. Priorities are expressed as numbers from 0 to 9, with 0 being the lowest priority and 9 the highest.
Return	The operation is successful if the priority has been changed for all enumerated requests, failure otherwise. <ul style="list-style-type: none">• In case of a success, it will return a list of the unique Request IDs of those requests that match the Selector for which the priority was changed.• In case of failure, it will return a list of requests for which the priority failed to change and the reason, if available. Requests for which the operation failed maintain their current priority.

Previous Administrative Commands Required:

ServiceQueueInit, ServiceQueueStart, ServiceQueueStop

Administrative Commands Accepted AFTER this Command:

Depends on the Service Queue status (Initialized, Started or Stopped) when the command is issued.

ServiceQueueInfo, ServiceQueueBrowseRequests,
ServiceQueueChangeRequestPriority, ServiceQueueReQueueRequest,
ServiceQueuePauseRequest, ServiceQueueResumeRequest,
ServiceQueueCancelRequest

Requeuing a Service Request

The **ServiceQueueReQueueRequest** command removes one or more service requests from one Service Queue and submits it to a different Service Queue.

ServiceQueueReQueueRequest Attributes

Parameters	Description
Selector	Used to identify the affected request(s). The selector consists of name value pairs that have to match message header properties of the request. No filtering is possible based on request contents.
ServiceQueue	The new queue where request(s) must be re-queued.
Timeout	The period in milliseconds to wait before considering the operation a failure. If the value is zero there is no timeout.
Return	The operation is successful if the requeuing succeeded, failure otherwise. <ul style="list-style-type: none"> • In case of success it will return a list of unique Request ID of requests that match the selector, which were re-queued. • In case of failure it will return a list of requests that couldn't be re-queued and the reason, if available. Requests for which the operation failed remain in their initial Service Queue.

Previous Administrative Commands Required:

ServiceQueueInit, ServiceQueueStart, ServiceQueueStop

Administrative Commands Accepted AFTER this Command:

ServiceQueueInfo, ServiceQueueBrowseRequests,
ServiceQueueChangeRequestPriority, ServiceQueueReQueueRequest,
ServiceQueuePauseRequest, ServiceQueueResumeRequest,
ServiceQueueCancelRequest

Pausing a Service Request

The **ServiceQueuePauseRequest** command is used to pause one or more service requests residing in a Service Queue.

If the service request being paused is an aggregate, or a sub-aggregate request, then all of the associated sub-requests are paused as well.

ServiceQueuePauseRequest Attributes

Parameters	Description
Selector	Used to filter the enumerated requests. The selector consists of name value pairs that have to match message header properties of the request. Filtering based on request contents is not available.
Timeout	The period in milliseconds to wait before considering the operation a failure. If the value is zero there is no timeout.
Return	The operation is successful if all enumerated requests were paused, failure otherwise. <ul style="list-style-type: none">• In case of success it will return a list of unique Request IDs of the paused requests that match the Selector.• In case of failure it will return a list of requests that couldn't be paused and the reason, if available. Requests for which the operation failed remain in their current state.

Previous Administrative Commands Required:

ServiceQueueInit, ServiceQueueStart, ServiceQueueStop

Administrative Commands Accepted AFTER this Command:

ServiceQueueInfo, ServiceQueueBrowseRequests,
ServiceQueueChangeRequestPriority, ServiceQueueReQueueRequest,
ServiceQueuePauseRequest, ServiceQueueResumeRequest,
ServiceQueueCancelRequest

Resuming a Paused Service Request

The **ServiceQueueResumeRequest** command will allow you to resume a service request(s) in a Service Queue that was previously paused. See “*Pausing a Service Request*” on page 52.

- If a service request was not previously paused the command will result in an error.
- If the request being resumed is an aggregate, or a sub-aggregate, request, then all associated sub-requests will be resumed as well.

ServiceQueueResumeRequest Attributes

Parameters	Description
Selector	Used to filter the enumerated requests. The selector consists of name value pairs that have to match the message header properties of the request. Filtering based on request contents is not available.
Timeout	The period in milliseconds to wait before considering the operation a failure. If the value is zero there is no timeout.
Return	<p>The operation is successful if all enumerated requests were resumed, failure otherwise.</p> <ul style="list-style-type: none"> • In case of success, it will return a list of unique Request IDs of the resumed requests that match the Selector. • In case of failure, it will return a list of requests that couldn't be resumed and the reason, if available. <p>Requests for which the operation failed remain in their current state.</p>

Previous Administrative Commands Required:

ServiceQueueInit, ServiceQueueStart, ServiceQueueStop

Administrative Commands Accepted AFTER this Command:

WorkerInit, ServiceQueueInfo, ServiceQueueBrowseRequests, ServiceQueueChangeRequestPriority, ServiceQueueReQueueRequest, ServiceQueuePauseRequest, ServiceQueueResumeRequest, ServiceQueueCancelRequest

Canceling a Request in a Service Queue

The **ServiceQueueCancelRequest** command cancels one or more service requests in a Service Queue.

- The cancellation is logged in the Request Log as an administration failed request.
- If the request being canceled is an aggregate, or a sub-aggregate, then all of the associated sub-requests are canceled as well.

ServiceQueueCancelRequest Attributes

Parameters	Description
Selector	Used to filter the enumerated requests. The selector consists of name value pairs that have to match message header properties of the request. Filtering based on request contents is not available.
Timeout	The period in milliseconds to wait before considering the operation a failure. If the value is zero there is no timeout.
Return	The operation is successful if all enumerated requests were cancelled, failure otherwise. <ul style="list-style-type: none">• In case of success it will return a list of unique Request IDs of those cancelled requests that matched the Selector.• In case of failure it will return a list of requests that couldn't be canceled and the reason, if available. Request for which the operation failed remain in their current state.

Previous Administrative Commands Required:

ServiceQueueInit, ServiceQueueStart, ServiceQueueStop

Administrative Commands Accepted AFTER this Command:

Depends on the Service Queue status (Initialized, Started or Stopped) when the command is issued.

ServiceQueueInfo, ServiceQueueBrowseRequests, ServiceQueueChangeRequestPriority, ServiceQueueReQueueRequest, ServiceQueuePauseRequest, ServiceQueueResumeRequest, ServiceQueueCancelRequest

Worker Agent Commands

You can use the Worker Agent commands to start, initialize and administer Service Managers.

Loading and Initializing Service Managers

The **WorkerInit** command loads and initializes the Service Managers on each of the Worker machines.

WorkerInit Attributes

Parameters	Description
Configuration	<p>The XML configuration for the Worker containing the Service Managers to be initialized.</p> <p>In addition:</p> <ul style="list-style-type: none"> • Multiple WorkerInit commands can be issued before a WorkerStart. This allows multiple Service Queues to add their Service Managers to the Worker. • Service Managers are identified by name, and if a Service Manager is already active it will appear as a failed Service Manager. • A Service Manager can only listen to only one Service Queue.
Timeout	The period in milliseconds to wait before considering the operation a failure. If the value is zero there is no timeout
Return	<p>The operation is a success if all of the Service Manager processes have initialized, failure otherwise. In case of failure, it will return a list of Service Managers that failed to initialize and the reason, if available.</p> <p>Service Managers that initialized successfully will not revert back to their initial configuration.</p>

Previous Administrative Commands Required: None

Administrative Commands Accepted AFTER this Command:

WorkerStart, WorkerStop, WorkerShutDown, WorkerInfo

Fetching and Processing Service Requests

The **WorkerStart** command requests all Service Managers, located on a specific Worker machine, to listen to their assigned Service Queues and fetch all available service requests for processing.

WorkerStart Attributes

Parameters	Description
Timeout	The period in milliseconds to wait before considering the operation a failure. If the value is zero there is no timeout.
Return	The operation is successful if all Service Managers started, failure otherwise. In case of failure it will return the Service Managers that failed to start, and the reason, if available. Service Managers that started successfully will not be stopped.

Previous Administrative Commands Required:

WorkerInit

Administrative Commands Accepted AFTER this Command:

WorkerInit, WorkerStop, WorkerShutDown, WorkerInfo

Stop Fetching and Processing Service Requests

The **WorkerStop** command requests all Service Managers, located on a Worker to stop fetching and processing service requests from their assigned Service Queues.

WorkerStop Attributes

Parameters	Description
Timeout	The period in milliseconds to wait before considering the operation a failure. If the value is zero there is no timeout.
Return	<p>The operation is successful if all Service Managers stopped, failure otherwise.</p> <p>In case of failure it will return the Service Managers that failed to stop and the reason, if available.</p> <p>Service Managers that stopped successfully will not be restarted.</p>

Previous Administrative Commands Required:

WorkerStart or ServiceManagerStart

Administrative Commands Accepted AFTER this Command:

WorkerInit, WorkerStart, WorkerShutDown, WorkerInfo

Shutting Down the Worker Agent

The **WorkerShutdown** command allows you to shut down the Worker Agent by instructing all Service Managers to shut down.

When you issue this command, keep in the following;

- No further administration commands (except **WorkerStatus**) can be accepted, until the Worker Agent is relaunched using the **WorkerInit** command. See “*Loading and Initializing Service Managers*” on page 55
- Service Managers that do not shut down normally in a specified amount of time are not forced to shutdown by the Worker Agent.
- The shut down command does not impact the status of the Worker machine.

WorkerShutdown Attributes

Parameters	Description
Timeout	The period in milliseconds to wait before considering the operation a failure. If the value is zero there is no timeout.
Return	The operation is successful if the Worker Agent shut down, failure otherwise. In case of failure it will return the reason, if available.

Previous Administrative Commands Required:

WorkerInit, WorkerStop or ServiceManagerStop

Administrative Commands Accepted AFTER this Command:

WorkerInit, WorkerInfo

Retrieving Worker Agent Statistical Information

The **WorkerInfo** command will return the following Worker Agent configuration information:

- Name of the Worker Agent (hostname by default)
- Service Managers present on this Worker Agent (either started or stopped).
- Types of services supported.

WorkerInfo Attributes

Parameters	Description
Timeout	The period in milliseconds to wait before considering the operation a failure. If the value is zero there is no timeout.
Return	The operation is successful if it returns information about the Worker Agent, failure otherwise. It will return the reason for the failure, if available.

Previous Administrative Commands Required:

None

Administrative Commands Accepted AFTER this Command:

WorkerInit, WorkerStart, WorkerStop, WorkerShutDown

Service Manager Commands

The Service Manager is responsible for retrieving messages from the Service Queue that it services, and then delivering them to the appropriate Calligo E-Delivery service. The Service Manager is started by the Worker Agent using the **WorkerInit** command. See “*Loading and Initializing Service Managers*” on page 55.

Initializing Service Managers

The **ServiceManagerInit** command is used to configure the Service Manager. The Worker Agent will issue this command after it starts the process in which the Service Manager runs.

Note: The Service Manager must be stopped to accept this command.

ServiceManagerInit Attributes

Parameters	Description
Configuration	<p>The XML configuration for the Service Manager. See “Appendix B Configuration DTD”. It contains the following:</p> <ul style="list-style-type: none">• Name of the Service Manager and the number of concurrently running queue sessions.• Types of services to support and the Queue Agent to listen to• Request Log Queue to send request status updates to. <p>The Service Manager does not start processing requests from the queue.</p>
Timeout	<p>The period in milliseconds to wait before considering the operation a failure. If the value is zero there is no timeout.</p>
Return	<p>The operation is successful if the Service Manager initialized successfully, failure otherwise.</p>

Previous Administrative Commands Required:

WorkerInit, WorkerStop, ServiceManagerInit, ServiceManagerStop

Administrative Commands Accepted AFTER this Command:

WorkerStart, WorkerShutDown, ServiceManagerStart, ServiceManagerShutDown, ServiceManagerStatus, ServiceManagerInfo

Fetching and Processing Requests

The **ServiceManagerStart** command requests the Service Manager to fetch and process requests from the Service Queue to which it has been assigned.

Typically, the Service Queue will use this command as a broadcast to start the Service Managers that it has been assigned.

ServiceManagerStart Attributes

Parameters	Description
Timeout	The period in milliseconds to wait before considering the operation a failure. If the value is zero there is no timeout.
Return	The operation is successful if the Service Manager started successfully, failure otherwise.

Previous Administrative Commands Required:

WorkerInit, ServiceManagerInit, ServiceManagerStop

Administrative Commands Accepted AFTER this Command:

WorkerStop, WorkerShutDown, ServiceManagerStop, ServiceManagerStatus, ServiceManagerInfo, ServiceManagerShutDown.

Stop Fetching Request Messages

The **ServiceManagerStop** command is used to stop the Service Manager from fetching messages from the Service Queue.

Typically the Service Queue uses this command as a broadcast to stop the Service Managers that it has been assigned. The Service Manager will finish processing the currently fetched request before it will stop.

ServiceManagerStop Attributes

Parameters	Description
Timeout	The period in milliseconds to wait before considering the operation a failure. If the value is zero there is no timeout.
Return	The operation is successful if the Service Manager stopped successfully, failure otherwise.

Previous Administrative Commands Required:

WorkerStart, ServiceManagerStart

Administrative Commands Accepted AFTER this Command:

WorkerInit, WorkerStart, WorkerShutDown, ServiceManagerInit, ServiceManagerStart, ServiceManagerShutDown, ServiceManagerStatus, ServiceManagerInfo

Shutting down Processing on the Service Manager

The **ServiceManagerShutdown** command requests the Service Manager to shut down its processing completely.

ServiceManagerShutdown Attributes

Parameters	Description
------------	-------------

Timeout	The period in milliseconds to wait before considering the operation a failure. If the value is zero there is no timeout.
---------	--

Return	The operation is successful if the Service Manager stopped successfully, failure otherwise.
--------	---

Previous Administrative Commands Required:

WorkerInit, WorkerStart, WorkerStop, ServiceManagerInit, ServiceManagerStop

Administrative Commands Accepted AFTER this Command:

WorkerInit

Reviewing Service Manager Runtime Performance

The **ServiceManagerStatus** command allows you to obtain a runtime status report for each Service Manager in the Calligo E-Delivery system.

ServiceManagerStatus Attributes

Parameters	Description
------------	-------------

Timeout	The period in milliseconds to wait before considering the operation a failure. If the value is zero there is no timeout.
---------	--

Return	Returns Online if the Service Manager is up and running, Offline otherwise.
--------	---

Previous Administrative Commands Required:

WorkerInit, WorkerStart, WorkerStop, ServiceManagerInit, ServiceManagerStart, ServiceManagerStop

Administrative Commands Accepted AFTER this Command:

Depends on the Service Queue status (Initialized, Started or Stopped) when the command is issued.

ServiceManagerStatus, ServiceManagerInfo

Retrieving Service Manager Statistical Information

The **ServiceManagerInfo** command will returns configuration information about the Service Manager. This includes information that is passed to ServiceManagerInit.

ServiceManagerInfo Attributes

Parameters	Description
Timeout	The period in milliseconds to wait before considering the operation a failure. If the value is zero there is no timeout.
Return	The operation is successful if it returns the configuration information for the Service Manager, failure otherwise.

Previous Administrative Commands Required:

WorkerInit, WorkerStart, WorkerStop, ServiceManagerInit, ServiceManagerStart, ServiceManagerStop

Administrative Commands Accepted AFTER this Command:

Depends on the Service Queue status (Initialized, Started or Stopped) when the command is issued.

ServiceManagerStatus, ServiceManagerInfo

Request Log Commands

The Request Log is a service that assists in tracing the location and status of all requests in the Calligo E-Delivery system.

The following is a list of the possible status values that can be logged for a request:

- Pending
- Paused
- Processing
- Completed-Success
- Completed-Failure
- Completed-Canceled.

Keep in mind the following when sending administration commands:

- Before you send an administration command to a Request, you should use the **RequestLogFindRequest** command to identify a request using metadata. For example, a submitter name.

This allows the **RequestLogRequestInfo** to return detailed information about the request.

- Aggregated requests have both a parent request ID and an aggregated request ID, in addition to the request ID.
 - When tracing an aggregated request there are multiple responses, one for each simple request that makes up the aggregate.
 - Each simple request also contains the request ID of its parent in the aggregate dependency tree. This allows the tree structure to be recreated from individual requests.

Initializing the Request Log Service

The **RequestLogInit** command is used to initialize the Request Log service.

RequestLogInit Attributes

Parameters	Description
Configuration	<p>The XML configuration for the Request Log service. It contains the following:</p> <ul style="list-style-type: none"> • Name of the queue where the request status messages are sent. • Number of concurrent queue listeners on the Request Log Queue • Note that other parameters that impact the performance of the Request Log Agent can be added in the future. • Parameters that define the static configuration are retrieved by the Request Log component from a configuration whose location can be configured. For example, the database name, database credentials, etc.
Timeout	The period in milliseconds to wait before considering the operation a failure. If the value is zero there is no timeout.
Return	Success if the Request Log service initialized, failure otherwise.

Previous Administrative Commands Required:

None

Administrative Commands Accepted AFTER this Command:

RequestLogStart, RequestLogShutDown RequestLogFindRequest, RequestLogRequestInfo, RequestLogStatus

Starting the Request Log Service

The **RequestLogStart** command is used to request the Request Log service to start accepting log entries.

RequestLogStart Attributes

Parameters	Description
------------	-------------

Timeout	The period in milliseconds to wait before considering the operation a failure.
---------	--

If the value is zero there is no timeout.

Return	Success if the Request Log service started, failure otherwise.
--------	--

Previous Administrative Commands Required:

RequestLogInit, RequestLogStop

Administrative Commands Accepted AFTER this Command:

RequestLogStop, RequestLogShutDown RequestLogFindRequest, RequestLogRequestInfo, RequestLogStatus

Stopping the Request Log Service

The **RequestLogStop** command is used to inform the Request Log to stop accepting log entries.

RequestLogStop Attributes

Parameters	Description
------------	-------------

Timeout	The period in milliseconds to wait before considering the operation a failure. If the value is zero there is no timeout.
---------	--

Return	Success if the Request Log service stopped, failure otherwise.
--------	--

Previous Administrative Commands Required:

RequestLogStart

Administrative Commands Accepted AFTER this Command:

RequestLogStart, RequestLogShutDown RequestLogFindRequest, RequestLogRequestInfo, RequestLogStatus

Shutting down the Request Log Service

The **RequestLogShutdown** command will shut down the Request Log service. The only accepted command after a shut down is RequestLogInit. See “*Transitioning between States*” on page 37.

RequestLogShutdown Attributes

Parameters	Description
------------	-------------

Timeout	The period in milliseconds to wait before considering the operation a failure. If the value is zero there is no timeout.
Return	Success if the Request Log service shut down, failure otherwise.

Previous Administrative Commands Required:

RequestLogInit, RequestLogStop

Administrative Commands Accepted AFTER this Command:

RequestLogInit

Retrieving a list of Request IDs

The **RequestLogFindRequest** command is used to return identifying information about one or more requests that match the search criteria specified in a Selector. See “*Appendix D - Using Selectors*”.

The information returned contains a list of Request IDs that can then be used to retrieve detailed information about each particular request. See “*Viewing Information about a Specific Request*” on page 72.

RequestLogFindRequest Attributes

Parameters	Description
Selector	Used to identify the searched request(s). The selector consists of name value pairs that have to match message header properties of the request. Filtering based on request contents is not available.
RequestCountLimit	Limits the number of requests returned. A system level maximum can be configured. If the number of requests that match the selector exceeds the request count limit, or the system setting whichever is lower, a warning is returned to inform you that there are actually more requests than were returned.
Timeout	The period in milliseconds to wait before considering the operation a failure. If the value is zero there is no timeout.
Return	<ul style="list-style-type: none"> • If successful it will return information about the request(s). If there are no requests matching the selector, the command it is still considered successful, but the list of requests will be empty. • If failed it will return the reason, if available.

Previous Administrative Commands Required:

RequestLogInit, RequestLogStart, RequestLogStop

Administrative Commands Accepted AFTER this Command:

Depends on Request Log status (Initialized, started or stopped),
RequestLogFindRequest RequestLogRequestInfo, RequestLogStatus

Viewing Information about a Specific Request

The **RequestLogRequestInfo** command will provide you with detailed information about a specific request. The request is identified based on its unique Request ID.

The information returned contains the request metadata and the status change information for the latest status change. For additional information see “*Chapter 5 - Tracking and Monitoring Requests.*”

RequestLogRequestInfo Attributes

Parameters	Description
RequestIdentification	This is the Request ID that uniquely identifies the request in the system.
Timeout	The period in milliseconds to wait before considering the operation a failure. If the value is zero there is no timeout.
Return	If successful it will return the detailed request information.

Previous Administrative Commands Required:

RequestLogInit, RequestLogStart, RequestLogStop

Administrative Commands Accepted AFTER this Command:

Depends on Request Log status (Initialized, started or stopped), RequestLogFindRequest RequestLogRequestInfo, RequestLogStatus

Viewing the Current Status of the Request Log

The **RequestLogStatus** command displays the current status of the Request Log service.

RequestLogStatus Attributes

Parameters	Description
------------	-------------

Timeout	The period in milliseconds to wait before considering the operation a failure. If the value is zero there is no timeout.
---------	--

Return	Returns Online if the Request Log service is up and running, Offline otherwise
--------	--

Previous Administrative Commands Required:

RequestLogInit, RequestLogStart, RequestLogStop

Administrative Commands Accepted AFTER this Command:

Depends on Request Log status (Initialized, started or stopped), RequestLogFindRequest RequestLogRequestInfo, RequestLogStatus

