webMethods Broker is the primary component in what is referred to as the “message backbone” in a webMethods integration environment. Along with other webMethods components, webMethods Broker facilitates asynchronous, message-based integration using the *publish-and-subscribe* model.

**Publish-and-Subscribe Model**

The publish-and-subscribe model is a specific type of message-based solution in which applications exchange messages (called *documents* in webMethods) through a third entity called *Broker*. In solutions based on this model, applications that produce information (publishers) send the information to the Broker entity and applications that require the information (subscribers) connect to the Broker and retrieve the information they need.

**Pub-Sub Application Model**

The Broker entity temporarily stores the documents that it receives from publishers in a message queue. Subscribers connect to the Broker entity and fetch documents from the queue. Depending on the application, a subscriber might publish a response after it retrieves and processes a message.

**Producers and Consumers Are De-coupled**

In the pub-sub model, information producers and consumers are de-coupled, meaning they do not interact with one another directly. Instead, each participant interacts only with the message and the Broker entity.

Participants in a pub-sub solution interact *asynchronously*. A program that produces information does not have to wait for the consumer to acknowledge receipt of that information. It simply publishes the information to the Broker and continues processing (also known as fire-and-forget). Consumers can connect to the message Broker and retrieve the information when they choose.

**Broker's Relationship with Other webMethods Components**

webMethods Broker functions as the message Broker for pub-sub solutions that you develop with webMethods. As shown in the following diagram, Broker provides the pub-sub infrastructure for many webMethods components.

Many webMethods components interact with the Broker.
webMethods Broker Architecture and Components

A webMethods Broker environment consists of two main components: Broker Server, the run-time component with which publishers and subscribers interact, and the Broker user interface, the administrative component that runs on My webMethods Server. One webMethods Broker environment can contain one or more Broker Servers. The Broker user interface connects to a Broker Server as an administrative client.

Any machine that hosts a Broker Server will also host a Broker Monitor. The Broker Monitor is automatically installed when you install Broker Server.

**Typical webMethods Broker Environment**

A host machine can host multiple webMethods Broker environments. You can install and run more than one instance of webMethods Broker on the same host machine.
**Broker Server**

Broker Server is the container within which one or more Brokers reside. A Broker Server performs the communication-related work of receiving client requests, dispatching requests to the requested resource (which in this case, is a Broker), and returning responses to clients. It also manages memory and disk resources for all the Brokers that reside on it.

**Elements Associated with a Broker Server**
**Brokers**
A Broker is an entity that resides on a Broker Server. When a client connects to Broker Server, the client specifies the Broker with which it wants to interact. A Broker encompasses the following types of objects:

- **Document types**, which identify the kinds of documents that the Broker's clients can exchange.
- **Client Groups**, which define specific properties and permissions that Broker applies to clients.
- **Client State Objects**, which maintain information about the individual clients that use the Broker.

A Broker Server is installed with a single Broker, called "default." When you configure multiple Brokers on a Broker Server, you can designate one of the Brokers to act as the **default Broker**. The default Broker is the one to which Broker Server will connect any client that does not explicitly specify the name of the Broker that it wants to use.

**Broker Monitor**
Broker Monitor is a program that executes alongside Broker Server on the host machine in a webMethods Broker environment. The Broker Monitor is automatically installed when you install webMethods Broker. Broker Monitor continually checks the state of the Broker Server and automatically attempts to restart it if it stops running.

Broker Monitor also is the program that starts a Broker Server. You usually do not start a Broker Server directly. Instead, you ask the Broker Monitor to start it for you. Broker Monitor monitors (and starts) all Broker Servers in the webMethods Broker environment. It is possible to have more than one Broker Monitor on a host machine if the host machine has more than one webMethods Broker environment. For each webMethods Broker environment there is only one Broker Monitor.

**Broker User Interface**
You use the Broker user interface to configure, monitor, and manage one or more Broker Servers and the Brokers that they host. The Broker user interface is a portal application that executes on My webMethods Server. It enables you to manage webMethods Broker from any browser-equipped computer in your organization's network.

**Document Types**
A document type is a schema-like definition that represents a kind of document that publishers and subscribers can exchange using the Broker. A document type has a name, a structure that consists of named fields, and a set of properties that determines how the Broker handles instances of that document type at run time. A document type must be registered on the Broker before clients can publish instances of that type. The name of the document type, which must be unique within a Broker, is carried by all documents of its type. Subscribers indicate which documents they want to receive by subscribing to specific document types. Document types are created when a developer uses Software AG Designer to define **publishable document types**. When a developer creates a publishable document type, the Integration Server to which that developer is connected automatically registers a corresponding document type on the Broker. You can also use the Broker admin APIs to create document types.

**Client Groups**
To connect to a Broker, a client program must declare the name of the **client group** to which it belongs. Conceptually, a client group represents a particular group or category of client (e.g., administrators, customers, Integration Server processes). On the Broker, a client group is represented by a client group object. This object has a name and a set of properties. The Broker applies these properties to clients that declare themselves to be members of that group.

A client states its group membership when it initially connects to a Broker. Once a client connects to a Broker and its client state object is created, the client cannot change its client group membership.
Other properties determine which document types a client is permitted to publish and to which document types it can subscribe. Client groups also play a key role in Broker security. If you enable SSL or basic authentication and require clients to authenticate themselves, the access control list in the client group determines which clients are authorized to connect as members of that particular group. As an administrator, you will create and configure client groups using the Broker user interface. You can also create client groups using the Broker admin API.

**Clients (Client State Objects)**

When a client initially establishes a connection with a Broker, the Broker generates a *client state object* for that client. The client state object maintains the following information about the client:

- **Client ID**: An ID that uniquely identifies the client.
- **Application Name**: An optional label that client programs can use to identify the system or application to which they belong.
- **Client Group**: The name of the client group to which the client belongs.
- **Subscription List**: The list of document types this client wants to receive.
- **Queue**: The list of document instances that are currently enqueued for the client.
- **Sessions**: A client can optionally enable "shared-state" when it connects to a Broker. The shared-state property enables multiple clients to connect to the Broker using the same client state object. (Typically, a client application does this to enable multiple client processes to retrieve and process documents from the client's queue, thereby improving performance.) When shared state is enabled, the client state object contains a list of sessions. Each session identifies the IP address of a client that is currently connected to the Broker using that client state object. For example a trigger on a cluster of two Integration Servers appears as one client on the Broker (that is, they share the same client state object), but the client state object has two sessions, one for each Integration Server.

**Subscriptions**

To receive documents, a client must register a subscription on the Broker. A subscription identifies the type of document the client wants to receive. If the client wants to receive only certain instances of a particular type, it can attach a filter to the subscription. When a subscription includes a filter, the subscriber receives only the documents that satisfy the filter criteria.

Subscriptions are registered by the client programs that interact with the Broker. When a developer creates a trigger on an Integration Server, for example, the Integration Server registers a subscription on the Broker for each document type associated with the trigger. When Broker receives a document for which there are no subscribers, it puts the document in the dead letter queue (if one is configured). If a dead letter queue has not been configured (which is the installed behavior), the document is discarded.

You can use the Broker user interface to view the subscriptions for a client and to delete a subscription. You can also use the Broker user interface to discover which clients subscribe to a particular document type.

**Queues**

The Broker places documents that match a client's subscription in the client's queue. Each client has one queue, which is part of its client state object. A document remains in the queue until the client retrieves it (and acknowledges that it has retrieved the document successfully) or until the document expires. To reduce memory usage, volatile documents that have expired can be proactively deleted at regular intervals, based on the size of the queue, from the client queues and forward queues before the client tries to retrieve them. When the Broker queues a document for a client, it does not actually place a copy of the document in the queue. The Broker maintains only one copy of a document instance. Queues belonging to the subscribers of that document's type contain pointers to that instance. You can use the Broker user interface to view the list of documents in a client queue and to examine the content of the documents.
themselves. You can also use the Broker user interface to delete documents from a queue, move documents from one queue to another, and change the order of the documents in a queue.

**Storage Type**

Documents published to Broker can be stored as *volatile documents* or *guaranteed documents*. Volatile documents are stored only in memory and are lost if the host machine loses power or the Broker Server is otherwise restarted. Guaranteed documents are persisted to disk so that they can be recovered in case of a power failure or a restart. Whether the Broker treats a document as volatile or guaranteed depends on the storage type associated with the document's type and the storage requirements of the client (as specified by the "Queue Type" property in the client's associated client group).

**Queue Storage (QS)**

*Queue storage* refers to the file storage that Broker Server uses to persist guaranteed documents and non-volatile Broker metadata (e.g., Broker definitions, document type definitions, client group definitions) to disk. *Queue storage consists of two files*: a *log file*, which is a fixed-sized file that Broker can write data to quickly, and one or more *storage files*, into which data is ultimately stored using a logged-commit process. Queue storage is shared by all Brokers on a Broker Server. Queue storage can be configured to operate in *combined mode* or *separate mode*. In combined mode, Broker metadata and run-time data are stored in the same log and storage files. In separate mode, the Broker uses two log files and two storage files. It stores metadata in one log/storage pair and run-time data in the other log/storage pair. When queue storage operates in separate mode, you can use the Broker Server's online backup utility to backup metadata while the Brokers on that server continue running.

As an administrator you can configure the size of the queue-storage files and monitor their usage.

**Territories**

Brokers can be linked to form units known as *territories*. Brokers that form a territory function as a single logical Broker. Clients that connect to one Broker in a territory can automatically exchange documents with clients connected to any of the Brokers in the territory. Territories support scalability by allowing you to segment traffic among multiple Brokers, while still treating the Brokers as a single operational unit. When Brokers operate in a territory, they maintain the same set of document types and client group definitions. (These objects are initially synchronized when a Broker joins a territory.) Any changes that occur to these objects on one Broker, are automatically propagated to the other Brokers in the territory. The Brokers that are members of a territory can reside on any Broker Server. The following figure shows a territory that is made up of three Brokers on two Broker Servers.
You use the Broker user interface to form territories and to monitor traffic that flows among the Brokers that make up the territory.

**Territory Gateways**

Territory gateways are links that you establish between territories. A territory gateway enables clients in one territory to receive documents that are published in another territory. Territory gateways are often used to connect territories that are geographically distant or whose clients share only a limited number of document types. Unlike territories, whose Brokers operate as a single unit and share the same document types and client groups, a gateway is simply a bridge that enables instances of specific document types to travel from one territory to another. (Message subscriptions are shared dynamically/implicitly in a territory and statically/explicitly across a gateway connection.) You use the Broker user interface to create territory gateways and to specify which document types can travel across them.

**Clusters**

Brokers can also be linked to form units known as *clusters*. Brokers that form a cluster function as a single logical Broker. All Brokers in a cluster maintain the same set of document types and client groups. However, each Broker supports its own set of clients.

In a cluster, published documents are not forwarded to other Brokers (peers). Clusters support client-side load balancing and failover by allowing you to use JMS clustering policies to distribute message traffic between multiple Brokers, while still treating the Brokers as a single operational unit. When Brokers operate in a cluster, they maintain the same set of document types and client group definitions. (These objects are initially synchronized when a Broker joins a cluster.) Any changes that occur to these objects on one Broker, are automatically propagated to the other Brokers in the cluster. The Brokers that are members of a cluster can reside on any Broker Server. The following figure shows a cluster made up of three Brokers on two Broker Servers.
**Cluster Gateways**

A *cluster gateway* enables two clusters to exchange documents. Unlike clusters, whose constituent Brokers automatically share their document types and client groups, a cluster gateway requires an administrator to explicitly specify which document types the clusters are permitted to send and receive. Moreover, the clusters never share client group information across a gateway.

**Broker Security Model**

You can secure webMethods Broker using basic authentication and SSL authentication. With basic authentication, clients must present the user name and password. With SSL authentication, clients must present digital certificates that verify their identity. For additional security, you can optionally encrypt the following connections configured with:

- Two-way SSL authentication
- One-way SSL authentication
- Basic authentication over one-way SSL

When you use SSL for encryption, Broker Server uses standard public/private key protocol to encrypt communications between it and the client, which ensures data confidentiality and integrity.

Broker also supports access control lists (ACLs) which enable you to restrict access to resources on the Broker to specified clients. To enforce ACLs, you must enable basic authentication or SSL authentication, and clients must present credentials to prove their identity.

**Managing Broker Using My webMethods**
You use the Broker user interface in My webMethods to manage and configure webMethods Broker. To access the Broker user interface, select the Administration function on the My webMethods Server home page and then click the Messaging tab.

**Selecting a Database**

If you are using My webMethods Server for the sole purpose of running the Messaging user interface in My webMethods, you can configure My webMethods Server to use the embedded database, Apache Derby, instead of an external RDBMS.

If you use My webMethods to administer Software AG webMethods products other than webMethods Broker, you must use an external RDBMS.

**To access the Broker user interface**

1. In your Web browser, enter the URL where My webMethods Server is hosted.
   
   Example: atlas:8080

2. At the log on screen, enter your My webMethods user name and password and click Login. If you just installed My webMethods Server, you can use the default administrator account as follows:
   
   For this parameter... Enter...
   
   User Name Administrator
   Password manage

3. Click Messaging to access the Broker user interface.

**Adding Broker Servers to My webMethods**

To manage a Broker Server, you must "add" the Broker Server to your Broker Server List. This list identifies the set of Broker Servers that you want to manage. When you open the Broker user interface, it automatically connects to the Broker Servers specified in your Broker Server List and displays their status.

**To add an individual Broker Server to the Broker Server List**


2. Click Add.

3. Specify the following:
   
   In this field... Specify...
   
   Host Name The name of the host machine on which the Broker Server is installed
   Port The port number on which the Broker Server is configured to listen for client requests.

4. Click Add.

**Discovering Broker Servers on a Specified Broker Monitor Host Machine**

If you know the name of the Broker Monitor host machine on which one or more Broker Servers are running.

**To discover and add Broker Servers running on a specified host machine**


2. Click Add.

3. Select the Discover Broker Server tab.

4. Specify the name of the Broker Monitor host machine on which the Broker Servers are installed.

   For example: atlas

5. Specify the port on which the Broker Monitor is runs. The Broker Server List will contain only the Broker Servers configured under the port you specify. If a port is not specified, the default port (6850) is used. If you do not know the port number, you can look it up using the server_config utility.

6. Click Discover.

7. When the list of Broker Servers appears, select the servers that you want to add to your Broker Server List and click Add to List.

**Uploading a List of Broker Servers into My webMethods**

You can also add multiple Broker Servers to your Broker Server List in a single step by creating a file that defines the locations of the Broker Servers and importing the definitions into My webMethods.
To use this approach, you must create a definitions file and upload the definitions file into My webMethods.

**Creating the Definitions File**

1. Use a text editor to create a file that identifies the Broker Servers that you want to add to your Broker Server List. For each Broker Server, identify the host computer on which the Broker Server runs and the port number to which it is assigned. Separate each entry with a carriage return (as shown below), a comma, a space, a tab, or a semicolon.

```
HostMachineName1:Broker Server port HostMachineName2
HostMachineName5:Broker Server port, Broker Monitor Port HostMachineName6, Broker Monitor Port
```

For example:
```
tokyo:9000
```
2. Save the file as an ASCII text file. When you save the file, be sure to do the following:
   - Give the file a .txt extension. The file must have this extension for My webMethods to import it.
   - Save the file to a location that you can access from your browser.

**To upload Broker Server definitions into My webMethods**

2. Click Add.
3. Select the Upload Broker Server tab.
4. Use the Browse button to select the file that contains the Broker Server definitions.
5. Click Upload.
6. When the list of Broker Servers appears, select the servers that you want to add to your Broker Server List and click Add to List.

**To remove a Broker Server from the Broker Server List**

2. In the Broker Server List, select the check box beside the Broker Server that you want to remove.
3. Click Delete.

**Viewing and Editing the Connection Parameters**

To view or edit the connection parameters
1. In My webMethods: Messaging > Settings.
2. Select the Connection tab.
3. Modify the parameters as necessary, then click Apply.

**Using My webMethods with ACL-Protected Broker Servers**

If you are using My webMethods to manage a Broker Server that is secured with an access control list (ACL), you must assign an identity to the Broker user interface. For basic authentication, an identity is composed of user name and password. For SSL authentication, an identity is composed of:

- A signed certificate, stored in a password-protected certificate file called a *keystore*.
- The trusted root of the certificate issuer (or authenticator), stored in a certificate file called a *trust store*.

**The Role of Broker Monitor**

Broker Monitor (awbrokermon) is a separate process that runs on the machine that hosts Broker Server. It has three functions:

- It starts Broker Servers.
- It monitors the state of the Broker Servers running on the host machine and automatically attempts to restart them if they fail.
- It logs status messages about the Broker Servers.
During installation, Broker Monitor binds to a port and IP address on the host machine. Broker Monitor listens for requests from administrative clients (such as the Broker user interface in My webMethods) on the port to which it is bound. The Broker Monitor default port 6850.

Typical Broker Monitor installation monitors all Broker Servers on the host machine

![Broker Monitor Diagram](image)

**How Broker Monitor Starts Broker Servers**

The Broker Monitor configuration file (awbrokermon.cfg) points to the Broker Servers that reside on the host machine. When you start Broker Monitor, it reads this configuration file and automatically starts all of the Broker Servers that are identified in the file.

The Broker Monitor's configuration file is updated automatically when you install Broker Server on the host machine and when you define additional instances of Broker Server using the server_config utility.

UNIX  
webMethods Broker_directory/bin/awbrokermon.cfg

Windows  
webMethods Broker_directory\bin\awbrokermon.cfg

**How Broker Monitor Monitors the State of Broker Servers**

When Broker Monitor starts a Broker Server, it internally captures the process ID that the operating system assigns to the server process. Using this ID, Broker Monitor continually monitors the server's run state. If the server exits unexpectedly, Broker Monitor automatically attempts to restart it. To avoid the situation where Broker Monitor keeps restarting a Broker Server that is unable to stay up and running, Broker Monitor will not restart a Broker Server that has experienced 3 unexpected exits within a 5 minute period.

**Status MessagesLogged by Broker Monitor**

Broker Monitor maintains a log file in which it records the following events for the Broker Servers that it monitors:

- Broker Server is launched (by Broker Monitor).
- Broker Server exits unexpectedly.
- Broker Server is stopped by an administrative action.

Windows Event Viewer (for example, Start > Settings > Control Panel > Administrative Tools > Event Viewer).

On UNIX, the information is written by the syslog daemon.

**Starting Broker Monitor Automatically on Windows Systems**

On Windows platforms, you generally configure Broker Monitor to run automatically when the host machine starts. The Software AG Installer registers Broker Monitor as a service with the following nomenclature:  
*Software AG Broker Monitor (port_number) (instance_number)*
The Broker Monitor service starts automatically when the host machine starts. The port number in the service name is the port number specified in the awbrokermon.cfg file.

**To start Broker Monitor manually using the startup utility**
1. Navigate to the `webMethods Broker_directory\bin` directory.
2. Run the following command: `.\startup.bat`

Broker Monitor starts, and then Broker Monitor automatically starts all the Broker Servers to be monitored by it. If Broker Monitor is already running, then the `startup` script does nothing.

**To start Broker Monitor service manually on Windows**
1. From the Start menu, open Services as follows:
   Settings > Control Panel > Administrative Tools > Services
2. Right-click the Software AG Broker Monitor (`port_number`) (`instance_number`) service and click Start.

The status of the Broker Monitor switches to "Started" and Broker Monitor starts the Broker Servers that reside on the machine.

**To configure Broker Monitor to start at system boot time on UNIX**
To automatically start and stop Broker Monitor with the UNIX system, you must install the Broker Monitor start/kill script `webMethods Broker_directory/Broker/aw_broker82` to the boot system. You must have root permissions to run the following commands.

On Solaris or Linux. On Solaris and Linux, you install the Broker Monitor start/kill script into the runlevel directories manually. The following example configures Broker Monitor to start/stop with HP-UX or Linux runlevel 3:

```bash
cp webMethods Broker_directory/Broker/aw_broker82 /etc/init.d
ln -s /etc/init.d/aw_broker82 /etc/rc3.d/S45broker82 # start
ln -s /etc/init.d/aw_broker82 /etc/rc3.d/K45broker82 # kill
```

**To start Broker Monitor manually on UNIX using the startup utility**
1. Navigate to the `webMethods Broker_directory/bin` directory.
2. Run the following command: `./startup.sh`

Broker Monitor starts, and then Broker Monitor automatically starts all the Broker Servers to be monitored by it. If Broker Monitor is already running, then the `startup` script does nothing.

**To start Broker Monitor manually on UNIX using the start script**
1. Navigate to the directory that contains the Broker Monitor script.
2. Run the following command:
   `./S45broker82 start`

Where `S45broker82` specifies the file name of the start up script.
3. To confirm that Broker Monitor is running, enter the following command: `ps -ef | grep awbromon` if Broker Monitor is running, "awbrokermon" will appear in the process list.

**Stopping Broker Monitor on Windows**

**To stop Broker Monitor manually on Windows using the shutdown utility**
1. Navigate to the `webMethods Broker_directory\bin` directory.
2. Run the following command: `\shutdown.bat`

Stops all the Broker Servers being monitored by the Broker Monitor, and then stops the Broker Monitor.

**To stop Broker Monitor service manually on Windows**
1. From the Start menu, open Services as follows:
   Settings > Control Panel > Administrative Tools > Services
2. Right-click the Software AG Broker Monitor (`port_number`) (`instance_number`) service and click Stop.
3. Click Yes to confirm that you want to stop the Broker Servers that Broker Monitor is monitoring.
To stop Broker Monitor manually on UNIX using the `shutdown` utility
1 Navigate to the `webMethods Broker_directory/bin` directory.
2 Run the following command:
   ```
   ./shutdown.sh
   ```
   Stops all the Broker Servers being monitored by the Broker Monitor, and then stops the Broker Monitor.

To stop Broker Monitor on UNIX
1 Navigate to the directory where the Broker Monitor script file resides.
2 Run the following command:
   ```
   ./S45broker82 stop
   ```
   Where `S45broker82` specifies the file name of the stop script.

How to Determine Whether Broker Monitor Is Running

To check Broker Monitor on Windows
1 From the Start menu, open Services as follows:
   Settings > Control Panel > Administrative Tools > Services
2 Verify that the status of the Software AG Broker Monitor (`port_number`) (`instance_number`) service is "Started."

Checking on UNIX
To check Broker Monitor on UNIX
Enter the following command:
``` ps -ef | grep aw
If Broker Monitor is running, "awbrokermon" will appear in the process list.

Ports and Running Multiple Instances of Broker Monitor

Two Broker environments on the same host machine with the same IP address

Two Broker environments on the same host machine with different IP addresses on the same port
Viewing the Port Setting for Broker Monitor

On a Windows machine, you can also view the Broker Monitor port setting in the service name. Windows lists the Broker Monitor services in the following format:

Software AG Broker Monitor (port_number) (instance_number)  Where:

- port_number is the Broker Monitor port specified in the awbrokermon.cfg file.

Changing the Port Setting for Broker Monitor

To change the port setting for Broker Monitor and update the service names:

1. Run the following command to stop and remove the Broker Server and Broker Monitor Windows services:
   `awbrokermon -unconfig`

2. Edit the value of `monitor-port` parameter in the awbrokermon.cfg file to change the port setting for Broker Monitor.

3. Run the following command to start and add the Broker Server and Broker Monitor Windows services:
   `awbrokermon -config`

Managing Broker Servers

Broker Server is a container-like process (awbroker) that hosts one or more Brokers. Broker Server manages the communication, memory management, and queue storage functions for all Brokers that it hosts.

Broker Data Directory

Every Broker Server has its own data directory, which holds Broker Server's configuration file and log files. Frequently, the data directory also holds Broker Server's queue storage files, but often these files are placed on a separate (usually faster) storage device. When you install Broker Server using the Software AG Installer, a single Broker Server is installed.

Windows: `webMethods Broker_directory\data\awbrokers82\default\`

Broker Server Configuration File (awbroker.cfg)

The Broker Server configuration file (awbroker.cfg) contains parameters that define a single Broker Server instance. The configuration file resides in Broker Server's data directory and supplies information such as Broker Server's license key location (in the license file), base port, and the location of its queue storage files.
Starting Broker Server from My webMethods

If your Broker Server appears in the Broker Server List, you can use the following procedure to restart it.

To manually start a Broker Server using My webMethods
1 In My webMethods: Messaging > Broker Servers > Servers.
2 In the Broker Servers List, select the check box beside the Broker Server that you want to start.
3 Click Restart.

To manually start a Broker Server from the command line (using broker_start)
1 On the machine where Broker Server is installed, navigate to the following directory:
   `webMethods Broker_directory/bin`
2 Run the following command:
   `broker_start hostName:portNum -monitor_port monitorPort`
   `broker_start localhost:6845 -monitor_port 7850`

To manually start a Broker Server from the Windows control panel
1 From the Start menu in Windows, open Services as follows:
   Settings > Control Panel > Administrative Tools > Services
2 Right-click the webMethods Broker Server service that you want to start and click Start.

Stopping Broker Server

If you want to shut down all Broker Servers that are running in a webMethods Broker environment, shut down Broker Monitor. When you stop a Broker Server, it immediately disconnects all clients and shuts down. Broker Server does not wait for in-process transmissions to complete before shutting down. Volatile documents that Broker Server has in memory are lost. Destroy-on disconnect clients are also discarded and are not restored when the Broker Server is restarted.
Explicit-destroy clients and their state (e.g., queues, subscriptions) are preserved and will be restored when you restart the Broker Server. Guaranteed documents that have been successfully persisted to the Queue Storage log file are preserved and will be placed in the appropriate client queues when the Broker Server restarts.

**To stop a Broker Server using My webMethods**
2. In the Broker Servers List, select the check box beside the Broker Server that you want to stop.
3. Click Stop.

**Stopping Broker Server Using the broker_stop Command**
If you want to stop a Broker Server from the command line, use the `broker_stop` command-line utility. The `broker_stop` utility functions as an admin client and issues the stop request to the Broker Server through the network.

**To stop a Broker Server from the command line (using broker_stop)**
1. On the machine where Broker Server is installed, navigate to the following directory:
   `webMethods Broker_directory/bin`
2. Run the following command:
   `broker_stop hostName:portNum -monitor_port monitorPort idParams`

**To stop a Broker Server from the Windows control panel**
1. From the Start menu, open Services as follows:
   Settings > Control Panel > Administrative Tools > Services
2. Right-click the webMethods Broker Server service that you want to stop and click Stop.

**Broker Server Communication Ports**
When you install Broker Server or create a new Broker Server with the `server_config` utility, you specify the server's *base port*. Broker Server uses the base port for non-SSL communications. It uses the two ports immediately below the base port for SSL-based communications. By default, port 6849 is the Broker Server base port. If you do not explicitly assign a base port when you install or create a Broker Server, it uses the default port 6849 for non-SSL requests and ports 6848 and 6847 for SSL requests.

**Ports Used by Broker Server**

<table>
<thead>
<tr>
<th>Base Port</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Non-SSL communications. The default is 6849.</td>
<td></td>
</tr>
</tbody>
</table>

**Base Port -1**
One-way (server-only) SSL authentication. Clients can optionally request encrypted transmissions on this port. When the default base port (6849) is used, the one-way SSL port runs at 6848.

**Base Port -2**
Two-way (client-and-server) SSL authentication. Clients can optionally request encrypted transmissions on this port. When the default base port (6849) is used, the two-way SSL port runs at 6847.

**To view the base port for a Broker Server**
2. Click the name of the server in the Broker Server List.
The base port appears on the Broker Server Information tab.
To change the base port assignment of a Broker Server

1 Stop the Broker Server.
2 On the machine where Broker Server is installed, navigate to the following directory:
   `webMethods Broker_directory/bin`
3 Run the following command:
   `server_config update dataDir -p portNum`

For example, on Windows you could use the following:
   `server_config update "c:\webMethods8.2\Broker\data\awbrokers82\My Broker" -p 6800`
4 Restart the Broker Server in port 6800 using the following command:
   `broker_start localhost:6800`

Broker Server Memory

Broker Server requires a substantial amount of memory to hold the documents that clients publish. Most of the memory is used to hold volatile documents, which exist only in memory. However, a certain portion of memory is used to cache guaranteed documents and other objects that are kept in queue storage.

Controlling Queue Size for Volatile Documents

You can proactively delete volatile documents from a queue by setting the "queuecleanup-enable" and the "queue-cleanup-threshold" parameters in the awbroker.cfg file of each Broker Server instance.

Limiting Memory Usage by Broker Server

You can use the "max-memory-size" parameter (in awbroker.cfg) to prevent Broker Server from receiving more documents than it has the physical memory to store. This parameter specifies an upper memory limit. When this memory limit is reached, the Broker Server stops accepting documents. When the max-memory-size parameter is set, Broker Server monitors the amount of memory that it is using for document storage. If the size of the incoming document plus the total amount of memory that is
already "in use" exceeds the max-memory-size limit, Broker Server rejects the document and returns an "out of memory" error to the client.

**Configuring the max-memory-size Parameter**

To configure the memory parameters:
1. Stop the Broker Server.
2. On the machine where Broker Server is installed, locate the Broker Server's configuration file (awbroker.cfg) and make a backup copy.
3. Open the configuration file with a text editor and locate the max-memory-size parameter. If your configuration file does not include this parameter, add it to the file.

```

snmp=1
eventlog=1
internal=1
max-memory-size=500

```

4. Modify the value of the max-memory-size parameter to specify the maximum amount of memory, in megabytes, that Broker Server can allocate for document storage before it begins blocking incoming documents. The following example sets the max-memory-size parameter to 1.2 gigabytes.

```

snmp=1
eventlog=1
internal=1
max-memory-size=1200

```

5. If you want Broker Server to allocate this amount of memory when it starts, include the preallocate-memory parameter and set its value to 1. If you do not want to preallocate this amount of memory, set the preallocate-memory parameter to 0 or omit it from the configuration file.

```

snmp=1
eventlog=1
internal=1
max-memory-size=1200
preallocate-memory=1

```

Important! If you enable the preallocate-memory parameter and Broker Server cannot obtain the amount of memory specified in max-memory-size, it will not start.

6. Save the configuration file.
7. Restart the Broker Server.

**Configuring Queue Storage**

Queue storage is the storage system that Broker Server uses to persist guaranteed documents and non-volatile metadata such as Broker definitions, document types, client groups, and explicit-destroy clients. All Brokers on a Broker Server share the same queue storage space.

Queue storage is composed of two types of files: a log file and one or more storage files.

**The Log File**

The log file is where Broker Server initially stores the data that it needs to persist to disk. When a client publishes a guaranteed document, Broker Server commits the document to the log file before it returns control to the client. If the Broker Server cannot successfully write the document to the log, it returns an error so the publisher knows that the document was not received successfully.
Broker Server is able to write to the log file very quickly, in part, because the file is a fixed size and is preallocated. When the log file becomes full, Broker Server transfers data from the log file to a storage file to make room for new, incoming data.

The maximum size for a log file is 2 GB on 32-bit platforms and 8 GB (as a practical limit) on 64-bit platforms.

**The Storage Files**

Storage files are used for long-term storage of configuration data and for guaranteed documents that have not yet been retrieved by their subscribers. The Broker Server automatically moves data (in 8 MB chunks) from the log file to a storage file when the log becomes full.

Log data is also moved to long-term storage each time you start the Broker Server. This process, which is often referred to as "playing the log," effectively "empties" the entire log file by moving pending items to a storage file and purging items that are expired or have been acknowledged by all of their subscribers.

When you configure a storage file you specify its *reserve size* and its *maximum size*. The *reserve size* specifies the file's starting size (that is, the amount of space that is immediately allocated to the file). The maximum size specifies the size to which the storage file can grow.

A Broker Server can have multiple storage files. When you install Broker Server using the SAG Installer, the Installer creates one storage file. You can create additional storage files using the `server_config` utility after you install Broker Server, however you must stop and restart the server to do this.

You cannot decrease the *reserve* or *maximum size* of a storage file, delete the storage file, or change the *file's location* after it is created.

When you configure the size and location of a storage file, keep these points in mind:

- The maximum size for a storage file is 32 gigabytes.
- The minimum reserve size is 16 megabytes.
- You can create up to 62 storage files for each session.

**Combined and Separate Storage Sessions**

Broker Server can be configured to use *combined* or *separate* storage sessions. When you use a combined session, Broker Server persists metadata and run-time data to the same set of queue storage files. When you use separate storage sessions, Broker Server persists metadata to one set of files (called the *configuration session*) and run-time data to another set of files (called the *data session*).

Metadata includes:

- Brokers
- Client groups
- Territories
- Clusters
- Gateways
- Explicit-destroy clients
- Access control lists
- Document types

Run-time data includes:

- Guaranteed documents
- Client queues
- Client queue statistics

By default, **Broker Server is installed with separate storage sessions**. If you use the `server_config` utility to create a new Broker Server, you can configure the new server to use separate or combined sessions. The use of separate storage sessions is strongly recommended for production environments. This session type enables you to use the `server_conf_backup` utility to back up the Broker Server metadata without taking the server offline.

To modify the size of a log file:

1. Stop the Broker Server.
2 On the machine where Broker Server is installed, navigate to the following directory:
webMethods Broker_directory/bin
3 Run the following command. Type the entire command on one line.
server_config storage dataDir -session_type qs
   -qs_log_file logFile logSize
The following example illustrates the configuration for a combined session:
server_config storage "C:\webMethods\Broker\data\awbrokers82\my Broker" -
   session_config qs
   -qs_log_file "C:\webMethods\Broker\data\awbrokers82\my Broker\BrokerConfig01.qs.log" 20M
The following example illustrates the configuration for a run-time data session:
server_config storage "C:\webMethods\Broker\data\awbrokers82\my Broker" -
   session_data qs
   -qs_log_file "C:\webMethods\Broker\data\awbrokers82\my Broker\BrokerData01.qs.log" 1G
To modify the size of a storage file
1 Stop the Broker Server.
2 On the machine where Broker Server is installed, navigate to the following directory:
webMethods Broker_directory/bin
3 Run the following command. Type the entire command on one line.
server_config storage dataDir -session_type qs
   -qs_storage_file storeFile storeSize reserveSize
[[-qs_storage_file storeFile storeSize reserveSize ...]]
The following example illustrates a configuration session or combined session:
server_config storage "C:\webMethods8.2\Broker\data\awbrokers82\my Broker" -
   session_config qs
   -qs_storage_file "C:\webMethods8.2\Broker\data\awbrokers82\my Broker\BrokerConfig01.qs.stor" 512M 100M
The following example illustrates a run-time data session:
server_config storage "C:\webMethods8.2\Broker\data\awbrokers82\my Broker" -
   session_data qs -qs_storage_file "C:\webMethods8.2\Broker\data\awbrokers82\my Broker\BrokerData01.qs.stor" 1G 100M
To add one or more storage files to queue storage
A storage session can have a total of 62 storage files.
1 Stop the Broker Server.
2 On the machine where Broker Server is installed, navigate to the following directory:
webMethods Broker_directory/bin
3 Run the following command. Type the entire command on one line.
server_config storage dataDir -session_type qs
   -qs_storage_file storeFile storeSize reserveSize
[[-qs_storage_file storeFile storeSize reserveSize ...]]
The following example illustrates a configuration session or combined session:
server_config storage "C:\webMethods8.2\Broker\data\awbrokers82\my Broker" -
   session_config qs
   -qs_storage_file "C:\webMethods8.2\Broker\data\awbrokers82\my Broker\BrokerConfig01.qs.stor" 512M 100M
The following example illustrates a run-time data session:
server_config storage "C:\webMethods8.2\Broker\data\awbrokers82\my Broker" -
   session_data qs -qs_storage_file "C:\webMethods8.2\Broker\data\awbrokers82\my Broker\BrokerData01.qs.stor" 1G 100M
The following example illustrates a run-time data session with multiple storage files:
```
server_config storage C:\webMethods\Broker\data\awbrokers\myBroker -
session_data qs -qs_storage_file
```

To view the queue storage utilization and settings for a server
2. Click the server in the Broker Server List.
3. Click the Utilization tab to display the storage sessions associated with the Broker Server.
4. Click Details to display the Broker ServerUtilization page.

To view the cache setting
2. In the Broker Server List, click the server whose cache setting you want to view.
3. Click the Utilization tab.
4. Click Details for the storage session (that is, Configuration Data, Run-Time Data, or Configuration and Run-time Data) whose cache setting you want to view. (Be sure to click the icon for the top-level storage session, not an individual storage file.)
5. Examine the Session URL value in the Storage Session section of the Server Utilization tab.

Configuring Broker Server to Use Asynchronous Write Mode
Broker Server can be configured to write data to disk asynchronously. In this mode, Broker Server permits the operating system to cache its disk write operations in memory.

To configure asynchronous-write mode
1. Stop the Broker Server.
2. On the machine where Broker Server is installed, locate the Broker Server's configuration file (awbroker.cfg) and make a backup copy of it.
3. Open the configuration file with a text editor and locate the session-config and session-data parameters, which are underscored in the example below. (If you are running a combined storage session, your configuration file will contain only the session-config parameter.)

```
  snmp=1
  eventlog=1
  internal=1
  session-config=qs:///var/opt/BrokerStorage/BrokerConfig.qs
  session-data=qs:///var/opt/BrokerStorage/BrokerData.qs?max_cache_size=512
```
4. To specify the cache size for a storage session, append the “async” argument to the session's URL. Both forms are shown in the example below.
To view the write mode setting
1 In My webMethods: Messaging > Broker Servers > Servers.
2 In the Broker Server List, click the name of the server whose write-mode setting you want to view.
3 Click the Utilization tab.
4 In the Details column, click the icon for the storage session (Configuration Data or Run-Time Data) whose write-mode setting you want to view.
5 Examine the Session URL value in the Storage Session section of the Server Utilization tab. If the URL includes the async argument, asynchronous write mode is enabled for the storage session.

To create a Broker Server with the default log and storage files
1 On the machine where Broker Server is installed, navigate to the following directory:
   webMethods Broker_directory/bin
2 Run the following command.
   server_config create dataDir -p port -k license-file-path [-d description]
   [-nostart] [-S]
   The following is an example:
   server_config create "c:\webMethods8.2\Broker\data\awbrokers82\My Broker" -p 6800
   -k WLRS8X-E073106-S000-KH-C10100-2SBR-N4NMPX -d "East Region Broker" -nostart

To create a Broker Server with specified log and storage files
1 On the machine where Broker Server is installed, navigate to the following directory:
   webMethods Broker_directory/bin
2 Run the following command.
   server_config create dataDir -p port -k license-file-path [-d description]
The following example illustrates a single run-time storage file on Windows:

```plaintext
server_config create "c:\webMethods8.2\Broker\data\awbrokers82\My Broker" -p 6800
-k WLRS8X-E073106-S000-KH-C10100-2SBR-N4NMPX -d "East Region Broker"

session_config qs
-qs_log_file F:\BrokerStore\myBroker\configData.qs.log 20M
-qs_storage_file F:\BrokerStore\myBroker\configData01.qs.stor 10G 100M

session_data qs
-qs_log_file F:\BrokerStore\myBroker\runtimeData.qs.log 256M
-qs_storage_file F:\BrokerStore\myBroker\runtimeData01.qs.stor 10G 100M

nostart
```

To delete a Broker Server
1 Stop the Broker Server.
2 On the machine where Broker Server is installed, navigate to the following directory:
`webMethods Broker_directory\bin`
3 Run the following command.
`server_config delete dataDir [-f] [-S]`
4 Example:
`server_config delete "c:\webMethods8.2\Broker\data\awbrokers82\My Broker"

**Viewing Status Information for a Broker Server**

To view status information
1 In My webMethods: Messaging > Broker Servers > Servers.
The Broker Servers List displays the following information:

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>-The Broker Server is online and operating normally.</td>
</tr>
<tr>
<td></td>
<td>-The Broker Server cannot be reached. This might occur because:</td>
</tr>
<tr>
<td></td>
<td>▪ The Broker Server is not currently running.</td>
</tr>
<tr>
<td></td>
<td>▪ The URL specified for this Broker Server is incorrect.</td>
</tr>
<tr>
<td></td>
<td>▪ My webMethods was not able to complete the connection within the timeout period. To retry the connection, click Go on the Search tab.</td>
</tr>
<tr>
<td></td>
<td>-The Broker Server is online, but one of the following abnormal conditions exist:</td>
</tr>
<tr>
<td></td>
<td>▪ The license is expired or about to expire.</td>
</tr>
<tr>
<td></td>
<td>▪ The Broker Server is running low on disk space.</td>
</tr>
<tr>
<td></td>
<td>▪ The Broker Server currently has the maximum permitted number of connections.</td>
</tr>
</tbody>
</table>
2 Click the name of the Broker Server or click Details to view the following details for a Broker Server.

**Broker Server License**

You require a license file to install Broker Server. Broker Server 8.0 version installations check for license information in a license file, while installations of earlier Broker Server versions use a license key. You use My webMethods to update the license file (for Broker Server 8.0 version) as well as the license key (for earlier versions of Broker Server).
The license key is a parameter in the awbroker.cfg file that enables Broker Server to operate in full capacity until a specified date and time. You receive the server's license key in the form of an XML file (for example, licensePIF80.xml) from Software AG, when you obtain your license. The license key is one of the parameters that you use to configure a Broker Server at the time of installation.

**Viewing and Changing the License Using My webMethods**
If Broker Server is running, you can use My webMethods to view and change the license.

**To view or change the license using My webMethods**
2. In the Broker Server List, click the server whose license information you want to view.
3. Click the License tab.
4. If you want to update the license, click the Change button.
5. In the Upload a New Server License File dialog box, type the full path of the new license file, or click Browse to navigate to the file.
6. Click Apply.
7. If you changed the license file, you must restart Broker Server for the new license information to take effect.

**Viewing and Changing the License Using server_config**
If Broker Server cannot be started due to a missing or invalid license, you must use the server_config utility to change the license.

**To change the license from the command line (using server_config)**
1. Stop Broker Server.
2. On the machine where Broker Server is installed, navigate to the following directory:
   
   `webMethods Broker_directory/bin`
3. Run the following command:
   
   `server_config update dataDir-k license-file-path`

**Monitoring Resource Utilization**
Use the following procedure to examine statistics relating to CPU, memory, and queue storage utilization.

**To display Broker Server resource utilization**
2. In the Broker Servers List, click the server whose resource utilization you want to view.
3. Click the Utilization tab to display the following statistics.
4. Click Detail to display detailed information for any of the resources.

**Logging**
There are four logs associated with webMethods Broker:

- **The Broker Server log** records operational and error messages issued by Broker Monitor and Broker Server. You can refer to this log to find out when a Broker Server started running or stopped running.
- **The Messaging log** records errors that My webMethods encounters when you use it to manage Broker Servers.
- **The configuration audit log** records changes to configuration and security settings made to a Broker Server, Broker, client group, client, document type, territory and gateway.
- **The document audit log** enables you to log published documents into the central audit database.
To configure Broker Server logging behavior
2. In the Broker Server List, click the server whose log you want to configure.
3. Click the Server Log tab and then click Broker Server Log Setting.
4. Under What to Log, select the types of messages that you want Broker Server to include in the log.
5. Under How to Log, specify whether you want Broker Server to log entries to the host system’s native logging system and/or an SNMP management system.
6. Click OK.

Viewing the Broker Server Log
To view the Broker Server log
2. In the Broker Server List, click the server whose log you want to view.
3. Click the Server Log tab to display the log entries.

Purging the Broker Server Log
To purge the Broker Server log
2. In the Broker Server List, click the server whose log you want to purge.
3. Click the Server Log tab then click Purge Log to display the Purge Broker Server Log page.
4. From the Purge Log Entries list, select the range of messages that you want to remove from the log.
5. Click Purge.

Viewing the Messaging Log
To view the Messaging log
1. In My webMethods: Messaging > Log.
The Messaging Log tab displays entries that were generated in your Messaging log during the last day.
To configure the Messaging log
1 In My webMethods: Messaging > Settings.
2 Select the Tool Log tab.
3 In Log Level, set the level at which you want messages logged. The Broker user interface logs all messages that are of the selected level or higher.
4 Enable the Log to Single File option if you want to maintain continuous (that is, not daily) log files.
5 Click Apply to save your changes.

Purging the Messaging Log that Uses Daily Files
To purge the Messaging log, you must delete the associated log files from the file system. If you keep daily log files, you can simply delete the log files from previous days.

To purge a Message log that uses daily log files
1 On the machine where My webMethods Server is installed, navigate to the directory:
   My webMethods Server_directory/Broker/logs
2 Locate and delete the log files for the user and time period that you want to purge. As shown below, the name of the log file will indicate the user to which the log belongs and the date on which it was created.
   userID.activityYYYYMMDD.xml
   userID.errorYYYYMMDD.xml
   For example:
   yourUserID.activity20060610.xml
   yourUserID.error20060610.xml

Purging the Messaging Log that Uses Continuous Files
If you use continuous files (that is, a single pair of activity and error files), you must stop My webMethods Server to purge the log.

To purge a Message Management log that uses continuous log files
1 Stop the My webMethods Server.
2 On the machine where My webMethods Server is installed, navigate to the directory:
   My webMethods Server_directory/Broker/logs
3 Locate and delete the activity and error log files for the user whose log you want to purge. Be aware that this step deletes the entire log, including the most recent entries. With continuous log files, there is no way to purge messages for a specified time period. The following shows what the names of the log files look like.
   userID.activity.xml
   userID.error.xml
   A new, empty pair of log files will be created the next time the user accesses the Broker user interface.

Backing Up and Restoring a Broker Server

Metadata that Is Backed Up
The procedures described in this section back up the following metadata:
- Brokers
- Document types
- Client groups
- Territory Membership
- Cluster Membership
- Gateways
- Explicit-destroy client state (except queue contents)
- ACLs
- Broker Server SSL configuration
Run-time information (document instances and client queues) is not backed up.
Other Files That You Should Backup

When you back up the Broker Server metadata, also back up the following files:
- Broker Server's configuration file (awbroker.cfg)
- Broker Monitor's configuration file (awbrokermon.cfg)
- Broker Server's keystore file (if the Broker Server is SSL-enabled)

Backing Up a Separate Storage Configuration

If your Broker Server is configured to use separate queue storage sessions, you use the _conf_backup utility to back up the queue storage files that contain the Broker Server's configuration data. You do not need to stop the Broker Server to use this utility. It is designed to run while the Broker Server is online.

Files that Are Backed Up

The backup utility copies the following files into a single compressed file.
- BrokerConfig.qs
- *.qs.stor
- BrokerConfig.qs.log

To back up the Broker Server using the server_conf_backup utility

1 On the machine where Broker Server is installed, navigate to the following directory:
   Software AG_directory/Broker/bin
2 Run the following command:

To restore the Broker Server using the server_conf_restore utility

1 Stop Broker Server.
2 Perform the following steps to obtain information about this Broker Server's queue storage system. You will use this information in later steps:
   a Navigate to the following directory: webMethods Broker_directory/bin
   b Run the following command to display details about the existing queue storage system:
      server_config storage dataDir -list
   where dataDir is the path to Broker Server's data directory. Enclose the entire directory name in quotes if any portion of the name contains a space.
   c Print or otherwise record the information that the command displays. You will use this information later when you create new instances of these files.
3 Delete all of the existing queue storage files for this Broker Server (that is, the files ending "qs," ".qs.stor," and ".qs.log"). Use the output from step 2c to locate these files.
4 Perform the following steps to prepare the Broker Server to accept the restored files.
   a Move the Broker Server's awbroker.cfg file out of Broker Server's data directory and into a different directory. (You cannot complete the next step if the awbroker.cfg file exists in the Broker Server's data directory.)
   b Use the "server_config create" command (shown below) to initialize Broker Server's queue storage system. The following points apply when you run this command:
      i You must create queue storage files using the exact names, locations, and sizes as the files described in the report generated in step 2c.
      ii Be sure to include the -nostart option to prevent Broker Server from starting after you run this command.
      iii If you do not know the Broker Server's port number or do not find the license file, get that information from the original awbroker.cfg file that you moved in step 4a.
   server_config create dataDir -p port -k license-file-path -nostart
   -session_config qs configSessionParameters
   -session_data qs dataSizeSessionParameters
5 Run the following command to restore the configuration queue storage files to the Broker Server:
```
server_conf_restore dataDir backupFile -o
```
Restart the Broker Server

To back up Broker Server metadata using the file system copy command
1 Disconnect clients and drain the queues of guaranteed documents.
2 Stop the Broker Server.
3 Perform the following steps to locate all of the queue storage files associated with the Broker Server's queue storage system.
   a Navigate to the following directory: 
      webMethods Broker_directory/bin
   b Run the following command to display details about the existing queue storage system:
      `server_config storage dataDir -list`
      where `dataDir` is the path to the Broker Server's data directory. Enclose the entire directory name in quotes if any portion of the name contains a space.
4 Locate and copy all of the queue storage files identified in the output from step 3b. If your Broker Server uses the default set of storage files, you would locate and copy the following files:
   - BrokerConfig.qs
   - BrokerConfig.qs.log
   - BrokerConfig.qs.stor
   Important! If the Broker Server has multiple storage files, be certain that you copy them all.
5 Restart the Broker Server.

To restore Broker Server metadata using the file system copy command
1 Stop the Broker Server.
2 Perform the following steps to locate all of the queue storage files associated with the Broker Server's queue storage system.
   a Navigate to the following directory: 
      Software AG_directory/Broker/bin
   b Run the following command to display details about the existing queue storage system:
      `server_config storage dataDir -list`
      where `dataDir` is the path to the Broker Server's data directory. Enclose the entire directory name in quotes if any portion of the name contains a space.
3 Locate and delete all of the queue storage files for this Broker Server as identified in the output from step 2b. If your Broker Server uses the default names for queue storage files, you would delete the following files:
   - Broker.qs
   - Broker.qs.log
   - Broker.qs.stor
4 Copy the queue storage files from the backup files to their original locations.
5 Restart the Broker Server.

Pausing Document Publishing on the Broker Server
When document publishing is paused on a Broker Server, the Broker Server refuses all incoming documents. The documents that exist in the publishing client queues continue to be consumed by the subscribing clients until the queues are emptied.

To clear all client queues on the Broker Server
1 In My webMethods: Messaging > Broker Servers > Servers.
2 Click the name of the server in the Broker Server List.
3 Click the Pause Publishing button.
The publishing clients refuse incoming documents until publishing is explicitly resumed.
4 To restart publishing to the client queues, click Resume Publishing.
Configuring Streaming in Broker Servers

To enable streaming in a Broker Server
1 Stop the Broker Server.
2 On the machine where Broker Server is installed, locate the Broker Server's configuration file (awbroker.cfg).
3 Open the configuration file with a text editor, and set the rbroker-streaming-enabled configuration parameter:
   rbroker-streaming-enabled=1
4 To specify the batch size of acknowledgements, set the value of the rbrokerstreaming-window-size configuration parameter.
5 Set the connection-sndbuf-size and the connection-rcvbuf-size configuration parameters for optimizing the TCP/IP connection buffering.
6 Save the configuration file.
7 Restart the Broker Server.

Enabling and Disabling Circular Gateways

To configure circular gateway connections
1 Stop the Broker Server.
2 On the machine where Broker Server is installed, locate the Broker Server's configuration file (awbroker.cfg), and open it with a text editor.
3 Configure circular gateways, as follows:
   - To enable circular gateways, set the enable-cyclic-gateway configuration parameter as follows:
     enable-cyclic-gateway=1
   - To disable circular gateways, either delete the enable-cyclic-gateway=1 line from the awbroker.cfg file or set the enable-cyclic-gateway configuration parameter as follows:
     enable-cyclic-gateway=0
4 Save and close the configuration file.
5 Restart Broker Server.

Role of Broker

Broker is an entity that resides on a Broker Server. When a client connects to Broker Server, it specifies the Broker with which it wants to interact.

Broker contains three key types of objects:
- Document types, which define the kinds of documents that clients of the Broker can exchange.
- Client groups, which are objects that represent a set of client properties and permissions. Any client that connects to a Broker must declare the client group to which it belongs.
- Clients, which are state objects that represent the client programs that connect to the Broker. The state object maintains a client's list of subscriptions and its queue. If a client program connects to the Broker as an "explicit destroy" client, the Broker maintains the state object even if the client program is physically disconnected from the Broker.
A Broker contains document types, client groups, and client state objects
Creating a Broker

When you install Broker Server using the Software AG Installer, it creates one Broker, called "default," on the Broker Server.

To create a Broker:
1. In My webMethods: Messaging > Broker Servers.
2. In the Broker Servers List, click the Broker Server on which you want to create the Broker.
3. Select the Brokers tab and click Add Broker.
4. If you want this Broker to act as the default Broker, enable the Default Broker option.
5. Click Add.
To delete a Broker
1 In My webMethods: Messaging > Broker Servers.
2 In the Broker Servers List, click the Broker Server on which the Broker resides.
3 Click the Brokers tab.
4 Select the check box beside the name of the Broker that you want to delete and click Delete.

Determining Which Broker Is the Default
To determine which Broker is the default
1 In My webMethods: Messaging > Broker Servers.
2 In the Broker Servers List, click the Broker Server on which the Broker resides.
3 Click the Brokers tab.
4 Scan the Default column. The default Broker (if one has been designated for this Broker Server) will have Yes displayed in this column.

To designate a default Broker
1 In My webMethods: Messaging > Broker Servers.
2 In the Broker Servers List, click the Broker Server on which the Broker resides.
3 Select the Brokers tab and click the Broker that you want to designate as the default.
4 Enable the Default Broker option and click Apply.
5 Click Close to return to the Brokers tab.

To determine whether Document logging is enabled on a Broker
1 In My webMethods: Messaging > Broker Servers > Brokers.
2 In the Brokers List, click the Broker whose logging option you want to check. If the Broker does not appear in the list, use the Search tab to locate it.
3 In the Configuration tab on the Broker Details page, check the state of the Document logging option.

Checking the Length of the Document Logging Queue
To check the length of the Document logging queue
1 In My webMethods: Messaging > Broker Servers > Brokers.
2 In the Brokers List, click the Broker whose logging queue you want to examine. If the Broker does not appear in the list, use the Search tab to locate it.
3 On the Broker Details page, click the Statistics tab and check the Document Log Length value. This value represents the number of documents in the log queue awaiting retrieval by the Logging Utility.

Routing Documents to the Dead Letter Queue
The dead letter queue is a queue in which Broker places dead letters. A dead letter is a document for which the Broker has no subscribers.

The Dead Letter Client
To activate the dead letter queue, you must configure the Broker's dead letter client. The dead-letter client is a system client that subscribes to dead letters. It is a member of the eventLog client group. This client group generates an explicit-destroy client with a guaranteed queue.
By default, the client ID for the Broker's dead-letter queue is "DefaultDLQ_{"], however, you can optionally append a suffix to this client ID when you activate the queue.

To activate the dead letter queue
1 In My webMethods: Messaging > Broker Servers > Brokers.
2 In the Brokers List, click the Broker whose dead letter queue you want to activate. If the Broker does not appear in the list, use the Search tab to locate it.
3 On the Broker Details page, click the Dead Letter Queue tab and (optionally) complete the following fields. If these fields are non-editable, it indicates that the dead letter client has already been configured.
4 Click Apply to generate the dead letter client.

To view the dead letter queue
1 In My webMethods: Messaging > Broker Servers > Clients.
2 Use the following steps to locate the dead letter client:
   a Type the following search string in the Keyword field on the Search tab: defaultDLQ
   b In Server Name and Broker Name fields, specify the Broker Server and Broker on which the dead letter queue resides.
   c Click Go.
3 Click the client ID associated with the dead letter queue for your Broker.
   • To view the contents of the queue, click the Browse Queue tab
   • To view statistics for the queue, click the Statistics tab.
   • To purge the queue, click the Statistics tab and then click the Clear Queue button.

Disabling the Dead Letter Queue
To disable the dead letter queue, you must delete the dead letter client. Deleting the dead letter client immediately discards any documents that are contained in the dead letter queue and de-activates the dead letter feature.

To disable the dead letter queue
1 In My webMethods: Messaging > Broker Servers > Clients.
2 Use the following steps to locate the dead letter client:
   a Type the following search string in the Keyword field on the Search tab: defaultDLQ
   b In Server Name and Broker Name fields, specify the Broker Server and Broker on which the dead letter queue resides.
   c Click Go.
3 In the Client List, select the check box beside the dead letter client ID.
4 Click Delete.

Viewing Status Information for a Broker
To view status information
1 In My webMethods: Messaging > Broker Servers > Brokers.
2 The Brokers List displays the status information about a Broker. If the Broker you want to examine does not appear in the list, use the Search tab to locate it.
3 To display additional details about the Broker, click the name of the Broker.

Viewing Basic Operating Statistics for the Broker
Use the following procedure to examine statistics relating to the number of documents that the Broker has received from clients.
To view basic Broker statistics
1 In My webMethods: Messaging > Broker Servers > Brokers.
2 In the Brokers List, click the Broker whose statistics you want to view. If the Broker does not appear in the list, use the Search tab to locate it.
3 Select the Statistics tab to display the following information
   Recent Broker Deliveries/Total Broker Deliveries /Document Log Length /Broker Retry Queue /Broker Trace Queue.
Managing Transactions
Transaction processing allows a Broker client to group the events it publishes as a single unit of work called a transaction. A transaction either completes successfully, is rolled back to some known earlier state, or it fails. Once all of the documents that make up a transaction have been published, delivered, or received, the Broker client ends the transaction. A transaction can be ended by committing the transaction or aborting the transaction.

Viewing Running Transactions
If a client has started a transaction on the Broker, that transaction appears on the Transactions tab. The transaction disappears from this tab when the client explicitly ends the transaction.

To view transactions running on Broker
1. In My webMethods: Messaging > Broker Servers > Brokers.
2. In the Brokers List, click the Broker whose transactions you want to view. If the Broker does not appear in the list, use the Search tab to locate it.
3. On the Broker Details page, click the Transactions tab. The transaction list at the bottom of this tab displays transactions that are currently running on the Broker.

Transaction ID/External ID/State

Configuring the Transaction Timeout Options
To configure the timeout settings
1. In My webMethods: Messaging > Broker Servers > Brokers.
2. In the Brokers List, click the Broker that you want to configure. If the Broker does not appear in the list, use the Search tab to locate it.
3. On the Broker Details page, click the Transactions tab and configure fields.

Pre-Prepare Timeout: The length of the pre-prepare timeout interval, in seconds, or -1 to set an infinite timeout interval.
Post-Prepare Timeout: The length of the post-prepare timeout interval, in seconds, or -1 to set an infinite timeout interval.
Post-Prepare Timeout Action: The action you want Broker to take if a transactional client exceeds the post-prepare timeout interval.
4. Click OK.

To specify the recovery mode for XA transactions
1. In My webMethods: Messaging > Broker Servers > Brokers.
2. In the Brokers List, click the Broker that you want to configure. If the Broker does not appear in the list, use the Search tab to locate it.
3. On the Broker Details page, click the Transactions tab.
4. In the Recover Mode field, select one of the following:
   Restricted: Clients can recover only transactions in which they participated.
   Global: Clients can recover any transaction.
5. Click OK.

Manually Performing a Commit or Roll Back
You can force a transaction to terminate using the Commit Transactions and Roll Back buttons on the Transactions tab.

To manually commit or roll back a transaction
1. In My webMethods, go to Messaging > Broker Servers > Brokers.
2 In the Brokers List, click the Broker on which the transaction is running. If the Broker does not appear in the list, use the Search tab to locate it.
3 On the Broker Details page, click the Transactions tab.
4 Select the check box beside the transaction that you want to rollback or commit.
5 Click Roll Back or Commit.

Viewing and Purging Lost Transactions
When a transaction ends heuristically, that is, when either the Broker or an administrator makes the decision to perform a commit or rollback for a transaction, that transaction is written to the Broker's lost transaction log. On the Broker, a heuristically completed transaction is referred to as a lost transaction.

To view and purge lost transactions
1 In My webMethods, go to Messaging > Broker Servers > Brokers.
2 In the Brokers List, click the Broker on which the transactions you want to view or purge are running. If the Broker does not appear in the list, use the Search tab to locate it.
3 On the Broker Details page, click the Transactions tab.
4 Click the Lost Transaction link to display the lost transaction log.
5 Select the check box beside the transactions that you want to purge and click Delete.

Managing Document Types
Documents are messages that travel over a network from a publisher to a subscriber, through the Broker. The role of the Broker is to route documents between information producers (publishers) and information consumers (subscribers). The Broker receives, queues, and delivers documents.
Each document is an instance of a document type. A document type is a schema-like definition that describes the structure of a document that publishers and subscribers can exchange using the Broker. A document type has a unique name, a structure that consists of named fields, and a set of properties that determines how the Broker handles instances of that document type at runtime. Clients indicate which documents they want to receive by subscribing to specific document types.

Creating Document Types
Document types can be created on Broker in one of the following ways.
- A developer uses Software AG Designer to define a document type and mark it as publishable. For each publishable document type created using Designer, Integration Server automatically registers a corresponding document type on the Broker.
- A developer can create a document type using the Broker client APIs.
- A developer can add a JMS topic as a Broker document type through My webMethods.
- Other webMethods components add document types to support their underlying system processes.

A document type must exist on the Broker before clients can publish instances of that type. The name of the document type, which must be unique within a Broker (and among all Brokers with which that Broker interacts), is carried by all documents of its type.

About Document Type Names
A document type name consists of two components: a folder path and a base document type name. 

```
folderName::subFolderName::documentTypeName
```

The folder path component can consist of one or more levels (subfolders). For example, for this fully qualified document type name: 

```
WesternRegion::Hardware::Sales::receiveCustOrder
```

The base document type name would be receiveCustOrder and the full folder path would be 

```
WesternRegion::Hardware::Sales.
```
Each document type has properties associated with it that determine how the Broker handles instances of that document type at run time. These properties include how long instances of the publishable document type should remain valid once they are published, document type validation, and the document type's storage type. You use My webMethods to modify these properties:

- **Time to live** specifies how long Broker maintains instances of this document type once they are published.
- **Validation** indicates whether Broker validates instances of this document type and if so, if the document can contain fields that are not defined in the document type.

There are three levels of document type validation:

- **Full.** The document can contain fields declared in the document type only. If the document contains fields that are not declared in the document type, Broker considers the document to be invalid.
- **Open.** The document can contain fields that are not declared in the document type. Fields that are defined in the document type must match the restrictions defined in the document type. Fields not defined in the document type can exist in the published document, but these undefined fields will not be type validated.
- **None.** Broker does not validate instances of this document type. This is the default.

**Storage type** determines whether instances of this document type are stored in memory or on disk.

### Document Type Storage Types

The document storage type setting determines how instances of that document are persisted. Documents published to Broker can be stored as **volatile documents or guaranteed documents**.

- **Volatile documents** are stored in local memory only. These documents are lost if the Broker host experiences a service interruption or the Broker Server is restarted. To reduce memory usage, volatile documents that have expired can be proactively deleted at regular intervals, based on the size of the queue, from the client queues and forward queues before the client tries to retrieve them.
  
  Volatile storage provides higher performance than guaranteed storage; however, there is a greater risk of losing documents if a hardware, software, or network failure occurs. All documents of a volatile document type and documents in a volatile client queue are lost when the Broker is shut down or when the computer restarts. This storage type is suited for documents that have a short life span or are not critical.

- **Guaranteed documents** are persisted to disk so that they can be recovered in the event of a power failure or a server restart. This is the default storage type for document types.

### Document Field Information

Document fields contain the data that your client applications use to exchange information. Document fields may contain a single value, a sequence of values with the same type, or a structure containing values of different types.

### Viewing Document Types on a Broker

To view document types on a Broker:

2. On the Document Types page, in the text box on the Search tab, type the keywords contained in the document types that you want to find.
3. Select the Broker Server and Broker whose document types you want to view.
4. Click Go to execute the search. My webMethods displays the search results in the Document Type List tab.

### Viewing Document Type Configuration Information

For each document type that exists, Broker maintains configuration information specified at the time the document type was created. Broker also maintains statistics for the document type, such as the number of subscriptions to the document type and the total number of published instances of the document type.
To view document type configuration information
1 In My webMethods: Messaging > Broker Servers > Document Types.
2 In the Document Type List, click the document type for which you want to view configuration
information. If the document type does not appear in the list, use the Search tab to locate it.
3 On the Document Type Details page, click the Configuration tab to view additional information and
statistics about a specific document type.
4 Click Close to return to the Document Types page.

Viewing Document Type Fields and Infosets
You can use My webMethods to view the fields and infosets for a specific document type. Fields contain
the data that your client applications use to exchange information. Infosets contain client application
specific information stored in document format.

To view document type fields and Infosets
1 In My webMethods: Messaging > Broker Servers > Document Types.
2 In the Document Type List, click the document type for which you want to view fields and Infosets.
If the document type does not appear in the list, use the Search tab to locate it.
3 On the Document Type Details page, click the Fields tab to view the fields defined in the document type.
4 On the Document Type Details page, click the Infosets tab to view the infoset information maintained
for the document type.
5 Click Close to return to the Document Types page.

Viewing Which Clients Subscribed to a Document Type
To view which clients have subscribed to a document type
1 In My webMethods: Messaging > Broker Servers > Document Types.
2 In the Document Type List, click the document type for which you want to view the clients that have
subscribed to it. If the document type does not appear in the list, use the Search tab to locate it.
3 On the Document Type Details page, click the Clients tab to view all the clients that have subscribed to
the document type.

Modifying Document Type Properties
To modify document type properties
1 In My webMethods: Messaging > Broker Servers > Document Types.
2 In the Document Type List, click the document type that you want to modify. If the document type does
not appear in the list, use the Search tab to locate it.
3 On the Document Type Details page, make sure the Configuration tab is selected.
4 Click Apply.

Copying Document Types Between Brokers
You can copy document types between any Brokers to which you have access in one of two ways:
- Copy and paste.
- Export and Import. When you use the export-import functionality, My webMethods places the
exported document type information in a file. The administrator for the destination Broker then imports
the file. To use the export-import functionality, you must have administrator access.

Copying and Pasting Document Types
When you copy and paste a document type, you make a duplicate copy of an existing document type on
another Broker. That is, copying and pasting documents between Brokers does not create a new document
type.
To copy and paste document types between Brokers
1 In My webMethods: Messaging > Broker Servers > Document Types.
2 In the Document Type List, select the check box next to the name of the document types that you want to copy. If the document type does not appear in the list, use the Search tab to locate it.
3 Click Copy.
   My webMethods copies the selected document types.
4 Click Paste. The Document Type List dialog box appears.
5 Under Select Target to Paste, select the Broker Server and Broker on which to paste the copied document types.
6 Click Paste. My webMethods places a copy of the document types on the selected Broker.

Deleting Document Types
Using My webMethods, you can delete document types on the Broker. You can only delete document types if:
- The document type has no subscribers.
- A client group does not have can-publish or can-subscribe permissions to the document type.
- The document type is user-defined. You cannot delete system-defined document types.
If you attempt to delete a document type for which there are subscriber, a document type to which a client group has can-publish or can-subscriber permissions, or a systemdefined document type, My webMethods displays an error and prevents the deletion of the document type.

To delete a Broker document type
1 In My webMethods: Messaging > Broker Servers > Document Types.
2 In the Document Type List, select the check box next to the name of the document types that you want to delete. If the document type does not appear in the list, use the Search tab to locate it.
3 Click Delete. My webMethods deletes the selected document type.

Managing Client Groups
Defining a Client Group
Every Broker client belongs to one client group, which you can use to define a particular class of client (such as administrators, customers, Integration Server processes). The settings for a client group belong to the following categories:
- Client characteristics, such as queue storage type and whether client information is saved after a disconnection. These are client group settings that are passed to all clients in the group.
- Security-related properties, which include ACLs, SSL encryption, and use of an access label. These settings determine which clients can join the client group, as well as other methods of protecting messages and message traffic.
- Access to document types, specified by lists of the document types available to the client group's members.

Client Queue Storage Type
Administrators need to select the storage mechanism that clients will use to queue documents.
- Volatile. Documents are stored only in local memory and are lost if the Broker Host loses power or the Broker Server is restarted.
- Guaranteed. Documents are persisted to disk and can be recovered in case of a power failure or restart.
Client Lifecycle

The *lifecycle* property determines whether the Broker maintains state information for a client after a disconnection. The lifecycle settings are:

- **Explicit destroy.** Maintains the Broker client's state object when the client disconnects. The client continues to receive documents in its queue, even though it is not actively connected. Use this setting for applications that need to maintain state information in the Broker between connections; for example, if a network or system failure occurs.
- **Destroy on disconnect.** Broker automatically destroys client state object information whenever the connection between the client and Broker is terminated, thus losing all knowledge of the client. With this setting, the state of a Broker client exists for the duration of the client's connection to the Broker.

**admin Client Group**

Members of the admin client group have full permissions and administrative privileges for a given Broker. Each Broker has one admin client group. Clients belonging to the admin client group can:

- Add any document type to any client group's can-publish and can-subscribe lists.
- Create client groups.
- Browse and modify any client's queue.
- Modify any Broker setting.
- Configure the territories or local gateways to which the Broker belongs

**accessLabelAdapter Client Group**

The accessLabelAdapter client group corresponds to an advanced security feature called the Access Label Adapter (ALA). Access labels allow you to prevent a Broker client from accessing a specific document regardless of whether its client group is authorized to access the document type to which the document belongs. The accessLabelAdapter client group has a destroy on disconnect lifecycle and a volatile storage type. You can modify the access control list (ACL) of the client group so that ALAs can authenticate using basic authentication or SSL; otherwise, you cannot modify the client group.

**eventLog Client Group**

This client group can subscribe to all document types defined within the Broker. If your installation is not using the document logging utility, you should delete this client group for security reasons. The eventLog client group is used by a Broker's dead letter queue facility, which receives messages for which there are no subscribers.

**Working with Client Groups**

The following sections describe the most commonly used procedures for working with Broker client groups:

- Searching for client groups
- Creating client groups
- Viewing information about client groups
- Editing client group settings
- Assigning publish and subscribe permissions
- Copying a client group from one Broker to another
- Deleting a client group

**To search for a client group**

1. In My webMethods: Messaging > Broker Servers > Client Groups.
2. Click the Search tab if it is not already displayed.
3. Select the Broker Server and Broker whose client groups you want to list by selecting:
To create a client group
1 In My webMethods: Messaging > Broker Servers > Client Groups.
2 On the Client Groups page, click Add Client Group.
3 On the Add Client Groups page, enter the information.
4 Click Add.

Viewing and Editing Client Groups
To view the list of client groups, view client group details, and edit client group settings
1 In My webMethods: Messaging > Broker Servers > Client Groups.
2 Select the Broker Server and Broker whose client groups you want to list and click Go.
3 For detailed information about a client group, or to edit a client group's settings, click its name. On the Client Group Details page is displayed, select the Configuration tab.
4 Click Apply to save any client group settings you modified.
5 To view a list of the document types that Broker clients in the client group are able to access, do the following:
   a Select the Can Publish tab to view the document types that clients are able to deliver.
   b Select the Can Subscribe tab to view the document types to which clients are able to receive deliveries.
   c Select the Log Publish tab to view the document types that clients are able to log when a document is published.
   d Select the Log Acknowledge tab to view the document types to which clients can make a log entry when a document is received.

Adding Document Type Permissions for Client Groups
1 In My webMethods: Messaging > Broker Servers > Client Groups
2 Select the Broker Server and Broker whose client groups you want to list and click Go.
The Broker user interface displays all client groups stored on that Broker in the Client Groups List
3 Click Add Document Type.
4 On the Document Types page, check the document types whose permissions you want to add and click Add.

Removing Document Type Permissions for Client Groups
Check the boxes of the document types to remove permissions and click Delete.

Copying Client Groups between Brokers
You can copy and paste client groups between any Brokers to which you have access. When you use the copy and paste functions, My webMethods copies and pastes the client group object to and from the clipboard. If you do not have access to the target Broker, use the export-import functionality instead of copying and pasting. In that case, you export client group definitions from the source Broker, and the administrator for the target Broker imports the client group definitions. In general, use copy and paste for simple objects, such as client groups, and importing and exporting for copying complex objects, such as a territory or a Broker.

Using Copy and Paste
To copy and paste a client group
1 In My webMethods: Messaging > Broker Servers > Client Groups.
2 Select the Broker Server and Broker of the client group you want to copy.
3 Click Go. If the client group does not appear in the list, use the Search tab to locate it
4 In the Search results, check the name of the client group you want to copy and then click Copy.
5 Click Paste. The Select Target to Paste dialog box appears.
6 Select the Broker Server and Broker (the target) on which to paste the copied client group.
7 Click Paste.

Deleting a Client Group
To delete a client group
1 In My webMethods: Messaging > Broker Servers > Client Groups.
2 Select the Broker Server and Broker of the client group you want to delete.
3 Click Go. If the client group does not appear in the list, use the Search tab to locate it
4 On the Client Groups page, click the box next to the client groups you want to delete and then click Delete.

Managing Clients
A client is an object on the Broker that represents the connection between a client program and the Broker. A client program creates a client to publish or retrieve documents.
A client program publishes documents to a Broker Server using its client. The Broker Server's publishing engine routes documents published by a client to other clients that subscribe to those documents.
A client program subscribes to documents by registering document type subscriptions with its client. Broker places the documents to which a client subscribes in the client's queue. A client program then retrieves the documents from its client queue.

Client State
Broker maintains state information for each client. State information includes the client ID, subscriptions, a queue of documents, sessions, and the client group to which the client belongs.

Shared State Clients
Multiple client programs (or multiple threads within a client program) can share the same client if the client was configured as a shared state client when it was created. Sharing a client allows multiple client programs (or multiple threads within a client program), each using its own session, to process documents from a single client queue in parallel on a first-come, first-served basis. Any changes to the state for a shared-state client affect all of the sessions using that client.

Storage Type and Lifecycle
The storage type for a client determines whether the client's state information is stored in memory or on disk. The client lifecycle determines whether the Broker maintains a client's state object after the client disconnects from the Broker. The client group to which a client belongs determines the storage type and the lifecycle for the client.

Viewing Clients on a Broker
To view clients on a Broker
1 In My webMethods: Administration > Messaging > Broker Servers > Clients.
2 To perform a keyword search, do the following:
   a In the Keywords text box on the Keyword tab, type the keywords contained in the client ID that you want to find. For example, you might specify:
      - The complete client ID.
      - A portion of the client ID to locate all the client IDs that contain the provided text.
   If you leave the text box blank, My webMethods searches for all the clients on the specified Broker.
   b Select the Broker Server and Broker whose clients you want to view
3 To perform an advanced search, click the Advanced tab.
4 Select the search condition from the Search Condition list.
5 Click Search. My webMethods displays the search results in the Client List.

To view client configuration information
1 In My webMethods: Messaging > Broker Servers > Clients.
2 In the Client List, in the Client ID column, click the name of the client for which you want to view configuration information. If the client does not appear in the list, use the search functionality to locate it. 

Note: If you selected a test client, My webMethods displays a Test Client View button in the Client Information tab. Click this button to switch to test client view.

3 On the Client Details page, click the Configuration tab to view configuration information about the client.
4 Click Close to return to the Clients page.

**Viewing Client Statistics**

My webMethods maintains statistics about a client including:
- Queue data, such as the number of documents in the queue, the number of unacknowledged documents in the queue, and the last time a document was placed in the queue
- Document information, such as the total number of documents a client has published or delivered
- Session information, such as the last time a client program established a session with the client and the time at which the client was created.

To view client statistics
1 In My webMethods: Messaging > Broker Servers > Clients.
2 On the Client List, in the Client ID column, click the name of the client for which you want to view statistics. If the client does not appear in the list, use the search functionality to locate it.
3 On the Client Details page, click the Statistics tab. The Statistics tab groups data into three different categories: Client Queue, Documents, and Sessions.
4 Click Close to return to the Clients page.

**Managing Subscriptions**

The document types to which a client subscribes determine the documents that Broker places in the client's queue. Using My webMethods, you can view and delete subscriptions for a client.

To view subscriptions for a client
1 In My webMethods: Messaging > Broker Servers > Clients.
2 On the Client List tab, in the Client ID column, click the name of the client for which you want to view document type subscriptions. If the client does not appear in the list, use the search functionality to locate it.
3 On the Client Details page, click the Subscriptions tab.
4 Click Close to return to the Clients page.

**Deleting Subscriptions**

You can delete any document type subscriptions for a client. You might want to delete a subscription to stop the unwanted or repeated delivery of documents from another client.

To delete subscriptions for a client
1 In My webMethods: Messaging > Broker Servers > Clients.
2 On the Client List tab, in the Client ID column, click the name of the client for which you want to delete document type subscriptions. If the client does not appear in the list, use the search functionality to locate it.
3 On the Client Details page, click the Subscriptions tab.
4 For each document type subscription that you want to delete, select the check box next to the name of the document type.
5 Click Delete.
6 Click Close to return to the Clients page.

**Managing Sessions**

A session represents a connection between an active client program and a client. A client program typically has one active session to a client. A client can have multiple concurrent sessions if the client is configured as a shared-state client. A shared-state client allows multiple sessions, possibly involving
clients on multiple hosts, to share the same client state, including its client ID, subscriptions, and queue. By default, each client is configured to allow only one active session. Broker assigns each connection between a client program and the client object a unique session identifier.

Using My webMethods, you can monitor and manage sessions to a client by:
- Viewing session statistics, such as the time a client program last established a session with a client.
- Viewing a list of active sessions to the client.
- Viewing detailed information about a session including encryption information.
- Disconnecting a session.

**Viewing Session Information**
You can view a list of the active sessions between client programs and a client. A client can have multiple active sessions if the client is a shared-state client.

**To view session information for a client**
1. In My webMethods: Messaging > Broker Servers > Clients.
2. On the Client List tab, in the Client ID column, click the name of the client for which you want to view session information. If the client does not appear in the list, use the search functionality to locate it.
3. On the Client Details page, click the Sessions tab.

**To view detailed session information**
1. In My webMethods: Messaging > Broker Servers > Clients.
2. On the Client List tab, in the Client ID column, click the name of the client for which you want to view detailed session information. If the client does not appear in the list, use the search functionality to locate it.
3. On the Client Details page, click the Sessions tab.
4. On the Sessions tab, click the IP address for the session for which you want to view detailed information. My webMethods displays the Client Session Detail page.

The Session Details tab groups client session information into two different categories: Platform Information and Encryption and Authentication Information.

**Disconnecting a Session**

**To disconnect a session**
1. In My webMethods: Messaging > Broker Servers > Clients.
2. On the Client List tab, in the Client ID column, click the name of the client for which you want to view detailed session information. If the client does not appear in the list, use the search functionality to locate it.
3. On the Client Details page, click the Sessions tab.
4. On the Sessions tab, for each session that you want to disconnect, select the check box next to the IP address.
5. Click Disconnect.

My webMethods prompts you to confirm deleting the session.
6. Click Disconnect to disconnect the client session.

**To copy and paste clients between Brokers**
1. In My webMethods: Messaging > Broker Servers > Clients.
2. On the Client List tab, in the Client ID column, select the check box next to the name of the each client that you want to copy. If the client does not appear in the list, use the search functionality to locate it.
3. Click Copy.

My webMethods copies the selected clients.
4. Click Paste. The Client List dialog box appears.
5. Under Select Target to Paste, select the Broker Server and Broker on which to paste the copied clients.
6. Click Paste.
Deleting Clients
You can delete clients that you do not need. When you delete a client, the client program disconnects from the Broker and the Broker destroys the client, including the client queue and any documents in the queue. This occurs regardless of the lifecycle of the client (destroy on disconnect or explicit destroy). You can only delete a client with an unlocked client queue. The Queue Locked field on the Configuration tab indicates whether a client queue is locked or unlocked.

To delete a client
1 In My webMethods: Messaging > Broker Servers > Clients.
2 On the Client List tab, in the Client ID column, select the check box next to the name of each client that you want to delete. If the client does not appear in the list, use the search functionality to locate it.
3 Click Delete.

Working with Test Clients
A test client is a function built in to My webMethods that you can use to quickly create clients on the Broker Server. A test client can function as a client program running inside My webMethods and can act as a client on Broker Server. In practical terms, a test client is a diagnostic tool that you can use to test the publishing and subscribing sides of a process. Test clients can be especially useful for diagnostic purposes when used in conjunction with the trace or topology features. When used together, these features allow you to publish and then monitor a document as it is routed by Broker from the publisher and distributed to subscribing clients.

Viewing Test Clients on a Broker
To view test clients on a Broker
1 In My webMethods: Messaging > Broker Servers > Clients.
2 On the Clients page, click the Advanced tab.
3 Select the Broker Server and Broker whose clients you want to view
4 Under Filter, in the Field Name list, select Category.
5 In the Value list, select Test Client.
6 Click Go.

Creating Test Clients
When you create a test client, you must select the Broker Server and Broker on which the test client will be located. You also must determine the client group to which the test client will belong. The client group determines the following for a test client:
- The document types the test client can publish and subscribe.
- The lifecycle of the test client.
- Whether the test client needs to provide SSL authorization information used by the test client's client group ACLs.

Creating a Test Client that Uses Basic Authentication
To create a test client that uses basic authentication
1 In My webMethods: Messaging > Broker Servers > Clients.
2 In the Client List, click Add Test Client. My webMethods opens the Create Test Client page.
3 On the Test Client tab, provide the information under Client Configuration.
4 In the Test Client Authentication options, select Basic.
5 Type the basic authentication user name in the Username field and password in the Password field.
6 Click Create. Broker Server creates the test client and returns you to the Clients page.

Creating a Test Client that Uses SSL Authentication
To create a test client that uses SSL authentication
1 In My webMethods: Messaging > Broker Servers > Clients.
2 In the Client List, click Add Test Client. My webMethods opens the Create Test Client page.
3 On the Test Client tab, provide the following information under Client Configuration.
4 In the Test Client Authentication options, select SSL.
5 Perform the following:
   a In the SSL Keystore field, select the keystore file containing the SSL certificate to be used by the test client to join the client group.
   b In the Password field, type the password to the keystore file.
   c Click Get User Names to retrieve a list of distinguished names (DNs) from the keystore.
   d In the User Name list, select the user DN for the test client.
   e In the SSL Keystore Type field, select the file type of the client keystore.
   f In the SSL Truststore field, specify the absolute path and file name of the truststore file containing the trusted root for the SSL certificate stored in the specified keystore.
   g In the SSL Truststore Type field, select the file type of the truststore certificate file.
6 Click Create. Broker Server creates the test client and returns you to the Clients page.

Working in Test Client View

Test client view is a set of pages and tabs in My webMethods that you use to build and manage a test client. In test client view, you can:
- Disconnect or reconnect a test client.
- Create, view, and delete document type subscriptions for a test client.
- Publish or deliver documents with a test client.
- Retrieve documents received by the test client.
You can use the regular client view, called Broker client view, to manage and monitor the test client in the same way in which you would a regular client.

To switch between test client view and Broker client view
1 In My webMethods: Messaging > Broker Servers > Clients.
2 In the Client List, click the client ID of the test client with which you want to work. If the test client does not appear in the list, use the Search tab to locate it.
3 On the Client Details page, click Test Client View on the Client Information page.
My webMethods displays the Test Client Details page.

Creating Subscriptions for a Test Client

When you create a test client to the test the subscription side of a publish-subscribe process, you need to establish document type subscriptions. As with any client, the client group to which a test client belongs determines the document types to which a test client can subscribe to or receive. A test client can subscribe to or receive any document types for which the client group has can-subscribe permissions. When a test client subscribes to a document type, Broker places all published instances of that document type in the test client's queue.

To create subscriptions for a test client
1 In My webMethods: Messaging > Broker Servers > Clients.
2 In the Client List, click the client ID of the test client for which you want to register document type subscriptions. If the test client does not appear in the list, use the Search tab to locate it.
3 On the Client Details page, click Test Client View on the Client Information tab. My webMethods displays the Test Client Details page.
4 Click the Subscriptions tab. The Subscriptions tab lists all of the document types to which the client has can-subscribe permissions.
5 Select the check box next to each document type to which you want the test client to subscribe.
6 Click Subscribe.
On the Subscription tab, My webMethods displays a "Yes" in the Subscription column for each document type to which the client subscribes.

**Creating or Editing a Filter**

To create or edit a filter

1. In My webMethods: Messaging > Broker Servers > Clients.
2. In the Client List, in the Client ID column, click the name of the test client for which you want to create document type subscriptions. If the test client does not appear in the list, use the Search tab to locate it.
3. On the Client Details page, click Test Client View on the Client Information tab. My webMethods displays the Test Client Details page.
4. Click the Subscriptions tab.
5. Locate the document type for which you want to create or edit a filter.
7. In the Filter Expression field, type the filter that you want to save with the subscription. Broker places only documents that meet the filter criteria in the test client queue.
8. Click OK. Broker saves the filter with the document type.

**Deleting Subscriptions for a Test Client**

When you delete a document type subscription for a test client, Broker stops placing documents of that type in the test client queue. You can delete a subscription for a test client using test client view or Broker client view.

To delete subscriptions for a test client

1. In My webMethods: Messaging > Broker Servers > Clients.
2. In the Client List, click the client ID of the test client for which you want to delete document type subscriptions. If the test client does not appear in the list, use the Search tab to locate it.
3. On the Client Details page, click Test Client View on the Client Information tab. My webMethods displays the Test Client Details page.
4. Click the Subscriptions tab.
5. For each current document type subscription that you want to delete, select the check box next to the document type name.
6. Click Unsubscribe. Broker removes the selected document type subscriptions.

**Publishing Documents with a Test Client**

To publish documents with a test client, you need to do the following:
- Build a list of documents to publish.
- Publish the documents for all subscribers or deliver the documents to a specific client.

You use the Outgoing Documents tab to build and publish a set of documents for a test client.

**Building an Outgoing Documents List**

To publish documents with a test client, you first need to build a list of documents to publish. The Outgoing Documents tab provides a staging area in which you can build this set of documents. You can build an outgoing documents list by doing one or more of the following:
- Adding blank documents
- Loading documents from a file
- Pasting documents copied from other locations in My webMethods

My webMethods maintains the outgoing documents list in its memory cache for the duration of your current session only. When you log out of My webMethods, your session is disconnected, or your session expires due to inactivity, My webMethods deletes the contents of the Outgoing Documents tab. If you want to use a set of documents across multiple sessions, save the documents to a file.
**To add an empty document**

1. In My webMethods: Messaging > Broker Servers > Clients.
2. In the Client List, click the client ID of the test client for which you want to add empty documents to the outgoing documents list. If the test client does not appear in the list, use the Search tab to locate it.
3. On the Client Details page, click Test Client View on the Client Information tab. My webMethods displays the Test Client Details page.
4. Click the Outgoing Documents tab.
5. Click Add Document to display the Test Client - Add Documents page.
6. On the Add Empty Document tab, do the following:
   a. In the Can Publish Document Type list, select the document type for the empty document that you want to insert in the outgoing documents list.
   b. Next to Where to Add, select the option that indicates whether you want to insert the documents at the beginning or end of the outgoing document list.
7. Click Add.

**Loading Documents from a File**

To load documents from a file

1. In My webMethods: Messaging > Broker Servers > Clients.
2. In the Client List, click the client ID of the test client for which you want to load documents in to the outgoing documents list. If the test client does not appear in the list, use the Search tab to locate it.
3. On the Client Details page, click Test Client View on the Client Information tab. My webMethods displays the Test Client Details page.
4. Click the Outgoing Documents tab.
5. Click Add Document. My webMethods displays the Test Client - Add Documents page.
6. Click the Load Documents tab.
7. In the File Name field, enter the path to the file containing the documents you want to add to the outgoing documents list. Click the Browse button to navigate to and select the file.
8. Next to Where to Append, select an option to indicate whether you want to insert the documents at the beginning or end of the outgoing document list.
9. Click Load.

**Copying and Pasting Documents in the Outgoing Documents List**

To add documents to the outgoing documents list by copying and pasting

1. In My webMethods: Messaging > Broker Servers > Clients.
2. If you want to copy documents from the Outgoing Documents tab of a test client, do the following:
   a. In the Client List, click the client ID of the test client for which you want to copy documents in the outgoing documents list. If the test client does not appear in the list, use the Search tab to locate it.
   b. On the Client Details page, click Test Client View on the Client Information tab. My webMethods displays the Test Client Details page.
   c. Click the Outgoing Documents tab.
   d. Select the check box next to each document that you want to copy.
   e. Click Copy. Proceed to step 4.
3. If you want to copy documents from the Browse Queue tab for a client, do the following:
   a. In the Client List, click the client ID of the test client for which you want to copy documents in the Browse Queue tab. If the client does not appear in the list, use the Search tab to locate it.
   b. On the Client Details page, click the Browse Queue tab.
   c. Click Start to view the documents in the queue.
   d. Select the check box next to each document that you want to copy.
   e. Click Copy. Proceed to step 4.
4. Click Clients at the top of the current My webMethods page.
5. In the Client List, click the client ID for the test client in which you want to paste the copied documents.
6 Click Test Client View.
7 Click the Outgoing Documents tab.
8 Click Add Document.
9 Click the Paste Documents tab.
10 Next to Where to Paste, select an option to indicate whether you want to insert the documents at the beginning or end of the outgoing document list.
11 Click Paste.

**Adding Comments to Documents**
To add a comment to a document
1 In My webMethods: Messaging > Broker Servers > Clients.
2 In the Client List, click the client ID of the test client for which you want to add comments to documents in the outgoing documents list. If the test client does not appear in the list, use the Search tab to locate it.
3 On the Client Details page, click Test Client View on the Client Information tab. My webMethods displays the Test Client Details page.
4 On the Test Client Details page, click the Outgoing Documents tab.
5 In the Document Type column, click the document type for which you want to add a comment. My webMethods opens the Document Details page.
6 On the Document Information tab, in the Comment field, enter the comment you want to add to the document.
7 Click OK. My webMethods displays the comment you added in the Comments column next to the document type.

**Publishing or Delivering Documents with a Test Client**
To publish or deliver documents from a test client
1 In My webMethods: Messaging > Broker Servers > Clients.
2 In the Client List, select the test client for which you want to publish documents. If the test client does not appear in the list, use the Search tab to locate it.
3 On the Client Details page, on the Client Information tab, click Test Client View.
4 On the Test Client Details page, click the Outgoing Documents tab.
5 Select the check box next to each document that you want to publish.
6 Click Publish. My webMethods displays the Test Client Publish Documents page.
7 On the Test Client Publish Document tab, select one of the following Publish Mode options:
   - Select PUBLISH/DELIVER
8 If you selected Deliver in step 7, in the Destination Client ID field, select the client ID for the client to which you want to deliver the document.
9 In the Number of Times to Publish field, specify the number of times you want Broker to publish the set of selected documents. The default is 1.
10 In the Delay between Publishing field, specify the number of milliseconds the Broker should wait between publishing each set of documents. The default is 0 milliseconds.
11 Click Publish.

**Stopping Publishing by a Test Client**
At times, you might want to stop a test client before it completes publishing of all selected documents.

To stop publishing by a test client
1 In My webMethods: Messaging > Broker Servers > Clients.
2 In the Client List, select the test client for which you want to stop document publishing. If the test client does not appear in the list, use the Search tab to locate it.
3 On the Client Details page, on the Client Information tab, click Test Client View.
4 On the Test Client Details page, click the Outgoing Documents tab.
5 Click Stop Publish.
Retrieving Documents for a Test Client

Broker places documents that match a test client's subscriptions in the test client queue. You use the Incoming Documents tab to retrieve and view these documents. By retrieving documents, you request documents from the client queue that Broker maintains for the test client. You can limit the number of documents retrieved with each request. By default, Broker retrieves up to 10 documents with each request. Broker stores the retrieved documents in the test client's memory cache.

**To retrieve documents for a test client**
1. In My webMethods: Messaging > Broker Servers > Clients.
2. In the Client List, select the test client for which you want to retrieve documents. If the test client does not appear in the list, use the Search tab to locate it.
3. On the Client Details page, on the Client Information tab, click Test Client View.
4. On the Test Client Details page, click the Incoming Documents tab.
5. In the Number of Documents to Retrieve field, specify the maximum number of documents that you want to retrieve for this test client at one time. Broker will retrieve up to the specified number. For example, if you specify five documents, but the client queue contains only four documents, Broker retrieves five documents. The default is 10 documents.
6. Click Retrieve Documents.

**To clear the incoming documents list for a test client**
1. In My webMethods: Messaging > Broker Servers > Clients.
2. In the Client List, select the client ID for the test client for which you want to clear the incoming documents list. If the test client does not appear in the list, use the Search tab to locate it.
3. On the Client Details page, click Test Client View on the Client Information tab.
4. On the Test Client Details page, click the Incoming Documents tab.
5. Click Clear Documents. My webMethods removes all the documents from the Incoming Documents tab.

Clearing the Test Client Queue

A test client queue contains all of the published and delivered documents that match the registered subscriptions for the test client. Documents remain in the test client's queue until you retrieve them using the Incoming Documents tab. However, you might want to remove documents from the test client's queue without retrieving them. For example, other clients, including other test clients, might have published documents that you are not interested in examining or retrieving.

**To clear a test client queue**
1. In My webMethods: Messaging > Broker Servers > Clients.
2. In the Client List, select the test client for which you want to empty the queue. If the test client does not appear in the list, use the Search tab to locate it.
3. On the Client Details page, click Test Client View on the Client Information tab.
4. On the Configuration tab, click Clear Queue. My webMethods deletes all the documents in the test client queue. Notice that the Queue Length value is now 0.

Saving Outgoing or Incoming Documents

You can save any of the documents on the Outgoing Documents tab or the Incoming Documents tab to a file. By saving documents to a file, you can:
- Use the same set of documents with multiple test clients.
- Use documents across multiple sessions. When you log out of My webMethods or your session expires, Broker deletes the contents of the Outgoing Documents tab and the Incoming Documents tab.
- Add or modify content of the documents by editing the XML file that contains the saved documents. You can then add the populated documents to the Outgoing Documents tab for a test client.
To save documents in the outgoing or incoming documents list
1. In My webMethods: Messaging > Broker Servers > Clients.
2. In the Client List, click the client ID of the test client for which you want to save the outgoing or incoming documents list. If the test client does not appear in the list, use the Search tab to locate it.
3. On the Client Details page, click Test Client View on the Client Information tab.
4. On the Test Client Details page, do one of the following:
   - To save documents on the outgoing documents list, click the Outgoing Documents tab.
   - To save documents in the incoming documents list, click the Incoming Documents tab.
5. Select the check box next to the documents that you want to save to a file.
6. Click Save to File. My webMethods displays the Export Documents page, which identifies the documents that will be saved to a file.
7. Click Save to File. The Internet browser prompts you to open the file or save the file to disk.
8. Select the option to save the file to disk. Enter a name and location for the exported documents and save the file.
9. On the Export Documents page, click Back to return to the Test Client Details page.

Disconnecting a Test Client

To disconnect a test client
1. In My webMethods: Messaging > Broker Servers > Clients.
2. In the Client List, click the client ID of the test client that you want to disconnect. If the test client does not appear in the list, use the Search tab to locate it.
3. On the Client Details page, on the Configuration tab, click Disconnect.
4. Click Close to return to the Clients page.

Reconnecting a Test Client that Uses Basic Authentication

To reconnect a test client that uses basic authentication
1. In My webMethods: Messaging > Broker Servers > Clients.
2. In the Client List, click the client ID for the test client that you want to reconnect. If the client does not appear in the list, use the Search tab to locate it.
3. On the Client Details page, on the Client Information tab, click Test Client View. My webMethods displays the Test Client Details page.
4. On the Configuration tab, click Reconnect.
5. In the Test Client Authentication options, select Basic.
6. Type the basic authentication user name in the Username field and the password in the Password field.
7. Click Reconnect.

Reconnecting a Test Client that Uses SSL Authentication

To reconnect a test client that uses SSL authentication
1. In My webMethods: Messaging > Broker Servers > Clients.
2. In the Client List, click the client ID for the test client that you want to reconnect. If the client does not appear in the list, use the Search tab to locate it.
3. On the Client Details page, on the Client Information tab, click Test Client View.
4. On the Configuration tab, click Reconnect.
5. On the Test Client Reconnect page, select SSL.
6. Perform the following:
a In the SSL Keystore field, select the keystore file containing the SSL certificate to be used by the test client to join the client group.
b In the Password field, type the password to the keystore file.
c Click Get User Names to retrieve a list of distinguished names (DNs) from the keystore.
d In the User Name list, select the user DN for the test client.
7 Click Reconnect.

Deleting a Test Client
You can delete a test client the same way in which you delete any client on the Broker. Broker continues to deliver documents to the test client as long as the test client exists. If you no longer need a test client, you should delete it.

To delete a client
1 In My webMethods: Messaging > Broker Servers > Clients.
2 On the Client List tab, in the Client ID column, select the check box next to the name of each client that you want to delete. If the client does not appear in the list, use the search functionality to locate it.
3 Click Delete.

Managing Client Queues

A client queue contains the published documents to which a client subscribes. Each client has one queue, which is part of its client-state object. A document remains in the queue until the client retrieves it (and acknowledges that it has retrieved the document successfully) or until the document expires (as determined by the Time to Live document type parameter). To reduce memory usage, volatile documents that have expired can be deleted at regular intervals, based on the size of the queue, from the client queues and forward queues before the client tries to retrieve them.

Client Queue Ownership
The client queue ownership is decided based on how the client queue is created.

Viewing Client Queue Statistics
My webMethods maintains statistics for a queue, including:
- The total number of documents in the queue, including the number of documents available for delivery and the number of unacknowledged documents in the queue
- The last time a document was placed in the queue
- The last time a client retrieved a document from the queue
- The size of the queue

Locking a Client Queue
Before you can modify the contents of a queue, you must lock the queue.
- You can have multiple queues locked at one time.
- You must have administrative privileges to lock a queue.
- Other users, including the queue owner, cannot clear the contents of a queue or delete the queue while it is locked.
- Broker continues to place documents to which the client subscribes in a locked queue.
- A client cannot retrieve documents from a locked queue.
- While the queue is locked, Broker processes document acknowledgements for documents that the client retrieved before you locked the queue.
To lock a client queue
1 In My webMethods: Messaging > Broker Servers > Clients.
2 In the Client List, click the client ID of the client whose queue you want to lock. If the client does not appear in the list, use the Search tab to locate it.
3 On the Client Details page, click the Configuration tab.
4 Click Lock Queue.

To unlock a client queue
1 In My webMethods: Messaging > Broker Servers > Clients.
2 In the Client List, click the client ID of the client whose queue you want to unlock. If the client does not appear in the list, use the Search tab to locate it.
3 On the Client Details page, click the Configuration tab.
4 Click Unlock Queue.

Viewing Lock Information for a Queue
My webMethods maintains and displays information about the lock status of a queue. Specifically, on the Configuration tab for a client, you can view the client ID for the client that locked the queue, the session during which the queue was locked, and the time at which the queue was locked.

To browse client queue contents
1 In My webMethods: Messaging > Broker Servers > Clients.
2 In the Client List, click the client ID of the client whose queue you want to browse. If the client does not appear in the list, use the Search tab to locate it.
3 On the Client Details page, click the Browse Queue tab.

To filter client queue contents
1 In My webMethods: Messaging > Broker Servers > Clients.
2 In the Client List, click the client ID of the client whose queue you want to browse. If the client does not appear in the list, use the Search tab to locate it.
3 On the Client Details page, click the Browse Queue tab.
4 Click Yes next to Document Type Filters.
5 On the Document Type Filters page, on the Filters tab, click Add Filter.
6 In the Document Type Name field, specify the fully qualified name of the document type you want to filter while browsing the queue.
7 In the Filter Expression field, type the expression that you want to use to filter the contents of the queue while browsing.
8 Click OK. My webMethods displays the Document Type Filters page.
9 Do one or more of the following:
   - Click Close to return to the Browse Queue tab on the Client Details page.
   - Click Add Filter to add a filter for a different document type.
   - Click Reset Filters to clear all filters.
   - Click the document type name to view details about the document type.

To save client queue documents to a file
1 Browse the queue to locate the documents you want to save.
2 On the Browse Queue tab, select the check box next to the documents that you want to save to a file.
3 Click Save to File. My webMethods displays the Export Documents page, which identifies the documents that will be saved to a file.
4 Click Save to File. The Internet browser prompts you to open the file or save the file to disk.
5 Select the option to save the file to disk. Enter a name and location for the exported documents and save the file.
6 On the Export Documents page, click Back to return to the Client Details page.
Inserting Documents into a Client Queue

You can add documents to a queue to make the documents available for processing by the client. For example, you might want to take a document from the dead letter queue and place it in the queue of the client that you want to process the document. Or, you might want to copy a document from a client queue experiencing a connection problem and place it in the queue of a client that you want to process the document. You might also want to reorder the documents in a queue by inserting documents at a new location and deleting documents at the old location.

To insert documents into a queue, you need to complete the following tasks:

- Build a list of documents to insert into the queue. You can either load documents from a file or copy documents from another client queue.
- Insert the documents at the beginning or end of the queue.
- You can build a list of documents for locked or unlocked queue. However, you must lock a queue before you insert documents into the queue.
- You can only insert documents that the client group to which the client belongs has can-subscribe permissions.

To load documents from a file

1. In My webMethods: Messaging > Broker Servers > Clients.
2. In the Client List, click the client ID of the client for which you want to insert documents. If the client does not appear in the list, use the Search tab to locate it.
3. On the Client Details page, click the Browse Queue tab.
4. Click Insert Documents.
5. Click Add Document. My webMethods displays the Add Documents page.
6. On the Load Documents tab, in the File Name field, enter the path to the file containing the documents you want to add to the queue. Click the Browse button to navigate to and select the file.
7. Next to Where to Append, select an option to indicate whether you want to add the documents at the beginning or end of the insert documents list.
8. Click Load.

To add documents to a queue by copying and pasting

1. In My webMethods: Messaging > Broker Servers > Clients.
2. Copy the documents you want to paste into the queue. You can copy documents from the following locations:
   - The Browse Queue tab of any client
   - The Outgoing Documents tab of a test client
   - The Incoming Documents tab of a test client
3. Click Clients at the top of the current My webMethods page.
4. In the Client List, click the client ID for the client in whose queue you want to paste the copied documents.
5. On the Client Details page, click the Browse Queue tab.
6. Click Insert Documents.
7. Click Add Document.
8. Click the Paste Documents tab.
9. Next to Where to Paste, select an option to indicate whether you want to insert the documents at the beginning or end of the insert documents list.
10. Click Paste.

To insert documents into a client queue

1. In My webMethods: Messaging > Broker Servers > Clients.
2. In the Client List, click the client ID of the client to which you want to add documents.
To the queue. If the client does not appear in the list, use the Search tab to locate it.
3 Lock the queue
4 On the Client Details page, click the Browse Queue tab.
5 Click Insert Documents.
6 Select the check box next to each document you want to insert into the queue.
7 Click Insert to Queue. My webMethods displays the Insert Documents to Queue Page.
8 On the Insert Documents tab, next to Where to Insert, select the option that indicates where in the queue you want to insert the documents.
9 Click Insert.

To delete documents from a client queue
1 In My webMethods: Messaging > Broker Servers > Clients.
2 In the Client List, click the client ID of the client for which you want to delete documents in its queue. If the client does not appear in the list, use the Search tab to locate it.
3 Lock the queue
4 On the Client Details page, click the Browse Queue tab.
5 Browse the queue to locate the documents you want to delete.
6 Select the check box next to each document that you want to delete from the queue.
7 Click Delete. My webMethods removes the documents from the client queue.

Clearing Client Queues

Before Clearing a Client Queue
Keep the following points in mind before clearing a queue:
- You can only clear an unlocked queue.
- Only the owner of a queue (that is, the client that created the queue) or a My webMethods user with administrative privileges can clear a queue.

To clear a client queue
1 In My webMethods: Messaging > Broker Servers > Clients.
2 In the Client List, click the client ID of the client for which you want to clear the queue. If the client does not appear in the list, use the Search tab to locate it.
3 On the Client Details page, click the Statistics tab.
4 Under Client Queue, next to the Length field, click Clear Queue.

Proactively Deleting Documents from a Client Queue
Expired documents in the client queues and forward queues; that is, the documents that have completed their Time to Live, are deleted only when the client tries to retrieve them. In cases where a client retrieves documents infrequently, this can lead to overloaded queues or memory overflow, which can further affect operations of other Brokers in the topology. To reduce the memory usage considerably, you can proactively delete the expired volatile documents from the client queues and forward queues even before the client tries to retrieve them. Expired documents are deleted whenever the total memory used by the volatile documents exceeds the specified threshold limit.
You can proactively delete volatile documents from a queue by setting the queuecleanup-enable and the queue-cleanup-threshold parameters in the awbroker.cfg file of each Broker Server instance.

To proactively delete volatile documents from a client queue
1 Stop Broker Server.
2 On the machine where Broker Server is installed, locate the Broker Server's configuration file (awbroker.cfg) and make a backup copy.
3 Open the configuration file with a text editor.
4 Locate the queue-cleanup-enable and the queue-cleanup-threshold parameters. If your configuration file does not include them, add them to the file `queue-cleanup-enable=1
The total size of the volatile documents in all the queues of the Broker Server instance. By default, the memory size threshold is 1024 MB, and the minimum allowed threshold is 1 MB.
For example, to delete expired volatile documents when their total size exceeds 2500 MB, specify: `queue-cleanup-threshold=2500`
5 Save the configuration file.
6 Restart Broker Server

Managing Forwarding Queues

A **forwarding queue** contains the published documents to which a remote Broker subscribes. For each remote Broker, a forwarding queue resides on the local Broker. A document remains in the queue until the queue forwards the document to the remote Broker and the remote Broker acknowledges that it has retrieved the document successfully.

Navigating to the Forwarding Queue for a Remote Broker in Your Territory

To navigate to the forwarding queue browser page for a remote Broker in your territory
1 In My webMethods: Messaging > Broker Territories > Territory Brokers.
2 In the Territory Brokers List, click on a Broker (local Broker) on which you want to browse the forwarding queue for the remote Broker. If your local Broker does not appear in the Territory Brokers list, use the Search tab to locate it.
3 On the Territory Broker Details page, click on the Forward Queue Length value corresponding to the remote Broker. This action opens the Forwarding Queue Browser Page on My webMethods.

Navigating to the Forwarding Queue for a Remote Broker in Your Cluster

To navigate to the forwarding queue browser page for a remote Broker in your cluster
1 In My webMethods: Messaging > Broker Clusters > Cluster Brokers.
2 In the Cluster Brokers List, click on a Broker (local Broker) on which you want to browse the forwarding queue for the remote Broker. If your local Broker does not appear in the Cluster Brokers list, use the Search tab to locate it.
3 On the Cluster Broker Details page, click on the Forward Queue Length value corresponding to the remote Broker in the cluster. This action opens the Forwarding Queue Browser Page on My webMethods.

Navigating to the Forwarding Queue for a Remote Broker in a Remote Territory

To navigate to the forwarding queue browser page for a remote Broker that resides in a remote territory across a gateway
1 In My webMethods: Messaging > Broker Territories > Territory Gateways.
2 In the Territory Gateway List, click the details icon corresponding to the gateway that connects your local territory (containing your local Broker on which you want to browse the remote Broker's forwarding queue) with the remote territory (containing the remote Broker whose forwarding queue you want to browse). If your gateway does not appear in the list, use the Search tab to locate it.
3 On the Territory Gateway Information page, click the Configuration tab.
4 Click on the Queue Length value. This action opens the Forwarding Queue Browser Page. This action opens the Forwarding Queue Browser Page on My webMethods.

Navigating to the Forwarding Queue for a Remote Broker in a Remote Cluster
To navigate to the forwarding queue browser page for a remote Broker that resides in a remote cluster across a gateway
1 In My webMethods: Messaging > Broker Clusters > Cluster Gateways.
2 In the Cluster Gateway List, click the details icon corresponding to the gateway that connects your local cluster (containing your local Broker on which you want to browse the remote Broker's forwarding queue) with the remote cluster (containing the remote Broker whose forwarding queue you want to browse). If your gateway does not appear in the list, use the Search tab to locate it.
3 On the Cluster Gateway Information page, click the Configuration tab.
4 In the table corresponding to Queue Length, click on the Forward Queue Length value corresponding to the cluster Broker whose forwarding queue you want to browse. This action opens the Forwarding Queue Browser Page on My webMethods.

**Locking and Unlocking the Forwarding Queue**

To lock or unlock the forwarding queue for a remote Broker
1 Navigate to the Forwarding Queue Browser page
2 To lock the queue, click the Lock button next to the lock icon. Or, click the Unlock button next to the lock icon to unlock the queue.

**Browsing a Forwarding Queue**

To view the contents of a forwarding queue for remote Broker, you browse the queue. When you browse a forwarding queue, you retrieve a list of documents from the queue. For each document, MywebMethods displays the acknowledgement status of the document, the document size, and the document sequence number. For example, you might want to browse a forwarding queue to verify whether a sales order was routed to the proper remote Broker.

By using My webMethods you can browse both unlocked, and locked forwarding queues.

To browse the forwarding queue contents for a remote Broker
1 Navigate to the Forwarding Queue Browser page.
2 On the Forwarding Queue Browser page, specify values for the queue browser settings. Start Position /Number of Documents/Timeout.
3 If you want to create (or edit) filters while viewing the forwarding queue content, click Yes next to Document Type Filters. Yes indicates that filters will be used to limit thenumber of documents for browsing.
4 Click Start to browse the first set of documents.

To filter forwarding queue contents for a remote Broker in your territory or cluster
1 Navigate to the Forwarding Queue Browser page.
2 On the Forwarding Queue Browser page, click Yes next to Document Type Filters.
3 On the Document Type Filters page, on the Filters tab, click Add Filter.
4 In the Document Type Name field, specify the fully qualified name of the document type you want to filter while browsing the queue.
5 In the Filter Expression field, type the expression that you want to use to filter the contents of the queue while browsing.
6 Click OK. My webMethods displays the Document Type Filters page.
7 Do one or more of the following:
   - Click Add Filter to add another filter for a different document type.
   - Click Reset Filters to clear all filters.
   - Click the Edit icon to edit the filter expression.
8 Click Close to return to the forwarding queue browser page.
To filter forwarding queue contents for a remote Broker that resides in a remote territory or a remote cluster across a gateway

1. Navigate to the Forwarding Queue Browser page.
2. On the Forwarding Queue Browser page, click "Yes" next to Document Type Filters.
3. On the Document Type Filters page, on the Filters tab, click Add Filter.
4. In the Document Type Name field, specify the fully qualified name of the document type you want to filter while browsing the queue.
5. In the Filter Expression field, type the expression that you want to use to filter the contents of the queue while browsing.
6. Click OK. My webMethods displays the Document Type Filters page.
7. Do one or more of the following:
   - Click Close to return to the Gateway Details page.
   - Click Add Filter to add a filter for a different document type.
   - Click Reset Filters to clear all filters.
8. Click the document type name to view details about the document type.

Saving the Forwarding Queue Documents to a File

You can save documents contained in the forwarding queue for one or more of the following reasons:
- To maintain copies of one or more documents in the forwarding queue before you delete the documents or clear the queue.
- To use the documents for testing purposes later. For example, you might want to save the documents and publish it with a test client.
- To modify the documents and then reinsert them into the forwarding queue.

To save documents from a queue to a file

1. Browse the queue to locate the documents you want to save.
2. On the forwarding queue browser page, select the check box next to the document type list that you want to save to a file.
3. Click Save to File. My webMethods displays the Export Documents page, which identifies the documents types that are being saved to the file.
4. The Internet browser prompts you to open the file or save the file to disk. Select the option to save the file to disk. Enter a name and location for the exported documents and save the file.

Inserting Documents into a Forwarding Queue

You insert documents into a forwarding queue for a remote Broker for the following reasons:
- You want to copy a document from a client queue (or forwarding queue) that is experiencing a connection problem and place it in the forwarding queue for another remote Broker.
- You want to reorder the documents in a queue by inserting documents at a new location and deleting documents at the old location.

To insert documents into a queue, you need to complete the following tasks:
- Build a list of documents to insert into the queue. You can either load documents from a file or copy documents from another client queue or forwarding queue.
- Insert the documents into the queue.

Keep the following points in mind when inserting documents to a queue:
- You can build a list of documents for locked or unlocked queue. However, you must lock a queue before you insert documents into it.
- My webMethods maintains the list of documents to insert into the queue for the duration of your current session only. When you log out of My webMethods, or your session gets disconnected, or your...
session expires due to inactivity, My webMethods loses the list of documents that you want to insert into the queue.

To load documents from a file:
1. Navigate to the Forwarding Queue Browser page.
2. Click Insert Documents.
3. Click Add Document. My webMethods displays the Add Documents page.
4. On the Load Documents tab, in the File Name field, enter the path to the file containing the documents you want to add to the queue. Click the Browse button to navigate to and select the file.
5. Next to Where to Append, select an option to indicate whether you want to add the documents at the beginning or end of the insert documents list.
6. Click Load.

Copying and Pasting Documents
To add documents to a forwarding queue by copying and pasting:
1. Navigate to the Forwarding Queue Browser page.
2. In the Forwarding Queue Browser page, click Insert Documents.
3. Click Add Document.
4. Click the Paste Documents tab.
5. Next to Where to Paste, select an option to indicate whether you want to insert the documents at the beginning or end of the insert documents list.
6. Click Paste.

Inserting Documents
After you build the list of documents to add to queue, you can insert the documents into the queue. Keep the following points in mind before inserting documents into a queue:
- You must lock the queue before inserting documents.
- You can insert any documents listed on the Insert Documents view of the Browse Queue tab.
- When you insert a document in the forwarding queue, you are adding it at the tail end of the queue.
- Broker inserts the documents into the queue in the same order in which the documents appear on the Insert Documents view of the Browse Queue tab. You use the buttons in the Move Up and Move Down columns to change the order of the documents.

To insert documents into a forwarding queue:
1. Navigate to the Forwarding Queue Browser page.
2. In the Forwarding Queue Browser page, click Insert Documents.
3. Select the check box for each document you want to insert into the queue.
4. Click Insert to Queue.

Deleting Documents from a Forwarding Queue
You can delete any document from a forwarding queue. You might want to delete documents from a queue for the following reasons:
- A document causes the application that processes the document to behave unexpectedly.
- A local Broker forwards invalid documents to the forwarding queue of the remote Broker.
- The remote Broker's subscriptions changed and you want to delete the already enqueued documents that the remote Broker no longer subscribes to.

You can delete documents from a forwarding queue regardless of the document's acknowledgement status.

To delete documents from a forwarding queue:
1. Lock the forwarding queue.
2. Browse the queue to locate the documents you want to delete.
3. Select the check box for each document you want to delete from the queue.
4. Click Delete.
**Topology View of Brokers**

The Broker topology view provides a graphical representation of a Broker. The topology view shows:

- Clients connected to the current Broker
- Remote Brokers connected to the current Broker
- Gateway Brokers in other territories where there is a gateway between the current Broker and the gateway Brokers shown in the topology view.

Additionally, you can trace documents in the topology view to show the documents that are actively flowing from producers to Broker to consumers.

Note: The Broker topology view requires that you have the Scalable Vector Graphics (SVG) viewer, which is provided by Adobe, installed on the machine where your browser is running. You can download this viewer from [http://www.adobe.com/svg/viewer/install/](http://www.adobe.com/svg/viewer/install/).

**Data Displayed in the Broker Topology View**

Icons and Lines

- The current Broker
- Remote Brokers
- Gateway Broker
- Client
Displaying the Broker Topology View

To display the topology view for a Broker
1 In My webMethods: Messaging > Broker Servers > Brokers.
2 In the Brokers List, click the Broker for which you want to display the topology view. If the Broker does not appear in the list, use the Search tab to locate it.
3 On the Broker Details page, click the Topology tab.

About Tracing Documents in Broker Topology View
While in the topology view, you can start a trace to view documents that are actively flowing across Broker connections, from producers to Broker to consumers. By default, Broker traces documents of all document types. The following shows a sample of a document being traced.

After you start a document trace, the trace continues as long as you stay on the Topology tab of the Broker Detail page until you manually stop the trace. However, if you switch to another page within My webMethods, a timeout timer starts. The timeout value is 30 seconds. To conserve system resources, after 30 seconds elapse, the system automatically stops the document trace. If you switch back to the Topology tab of the Broker Detail page before the timeout expires, the system clears the timeout timer and the trace will continue. You cannot configure the timeout value.

Tracing Documents in the Topology View
To run a document trace in the topology view
1 Display the topology view.
2 If you want to customize the display in the topology view, narrow down the clients and Brokers in the display, and/or narrow down the documents that Broker traces.
3 Click the start trace button.
While the trace is running, the start trace button is replaced with the stop trace button.
Stop the trace by clicking the stop trace button.

To define how to display data in topology view
1 Display the topology view.
2 Click the Display Settings link.
3 On the Display Options tab, set the fields:
4 Click Save to save your changes.
5 Click OK.

Defining the Brokers and Clients to Display in the Topology View
By default, the system displays the current Broker and all its connected clients in the topology view. You can select to display remote Brokers; disconnected, remote Brokers; and disconnected clients using the settings on the Display Options tab.
To define the Brokers and clients to display in the topology view
1 Display the topology view.
2 Click the Display Settings link.
3 Click the Node Grouping tab.
4 Click New Group to define a new node grouping.
5 On the Add Topology Display Group page, set the following fields.
6 Click OK to close the Add Topology Display Group page and return to the Node Grouping tab on the Broker Display Options page.
7 If you want to create another grouping, repeat steps step 4 through step 6.
8 After creating the node groupings, use the buttons in the Move Up and Move Down columns, which are on the Broker Display Options page, to put the groupings in the order you want.
9 Click OK on the Broker Display Options page.

**Editing Existing Node Groupings**

After you create a node grouping, you might want to update it. For example, you might originally have the Combined check box clear so that the members of the node group use individual nodes and want to change the node grouping to select the Combined check box so all members use a single node.

**To edit the settings for an existing node grouping**
1 Display the topology view
2 Click the Display Settings link.
3 Click the Node Grouping tab.
4 Click Edit for the node grouping you want to update.
5 On the Edit Topology Display Group page, update the Match Type, Match Name, Visible, and/or Combined fields.
6 Click OK to close the Edit Topology Display Group page and return to the Node Grouping tab on the Broker Display Options page.
7 Click OK on the Broker Display Options page.

**Deleting Node Groupings**

**To delete node groupings**
1 Display the topology view.
2 Click the Display Settings link.
3 Click the Node Grouping tab.
4 Select the check boxes in the rows of the node groupings you want to delete.
5 Click Delete.
6 Click OK on the Broker Display Options page.

**Ordering Node Groupings**

To order the node groupings
1 Display the topology view.
2 Click the Display Settings link.
3 Click the Node Grouping tab.
4 Use the buttons in the Move Up and Move Down columns to put the groupings in the order you want.
5 Click OK.

**To define the documents to include in traces**
1 Display the topology view.
2 Click the Display Settings link.
3 Click the Documents tab.
4 Select the check box for each document type that you do not want to trace.
   - To easily select all document types, click Hide All.
If you want to clear all check boxes for all document types, click Show All.
Tip! You can export the list of document types to a .csv file by clicking Export Table.

5 Click OK.

Table View of Trace Information
You can have the system collect data about documents that are actively flowing across client connections. You can then use the Trace tab to view the trace data in a table format, which is referred to as the trace view. After you start a document trace in trace view, the trace continues until you manually stop it. The trace does not timeout, even if you switch to another My webMethods page.

Running Document Traces in Trace View
You start or resume a document trace to indicate that you want the system to start storing information about documents that are actively flowing across client connections.

To start or resume a document trace in trace view
1 In My webMethods: Messaging > Broker Servers > Brokers.
2 In the Brokers List, click the Broker for which you want to run a document trace. If the Broker does not appear in the list, use the Search tab to locate it.
3 On the Broker Details page, click the Trace tab.
4 Click Resume Trace.
When you start or resume a trace, the system begins storing trace data in the memory of the My webMethods Server. You must refresh the trace view to have the trace data displayed on the Trace tab.

To stop a document trace
On the Trace tab, click Pause Trace.
If you want to start the trace again, you can resume it by clicking Resume Trace.

To export the trace data to a .csv file
1 On the Trace tab, click Export Table.
2 In the EXPORT TABLE dialog, select the character coding you want to use.
3 Click Export.

Clearing the Trace Data
If you no longer need to view the trace data, you can clear it from the page. Clearing the trace data from the display does not remove the data from memory. You must stop the trace to remove trace data from the memory.

To clear trace data from your machine's memory
On the Trace tab, click Clear.

Topology View of Territories
The territory topology view provides a graphical representation of territories. The territories in the display are those associated with the Broker Servers in your Broker Server list. The display also shows the gateways between territories, if any.

Data Displayed in the Territory Topology View
To display the territory topology
- In My webMethods: Messaging > Broker Territories > Topology.

**About Tracing a Document in the Territory Topology View**

While in the topology view, you can start a trace to view documents that are actively flowing across gateways. By default, the system traces documents of all document types. The following shows a sample of a document being traced.

Set display options to specify whether the document is to be labeled using the document type. The system includes the number of documents when multiple documents of the same type are crossing connections at the same time.

**To run a document trace in the topology view**
1. To limit the documents that the system traces, you must define the document types to trace per Broker.
2. To view the list of document types that the system will trace, you can use the Documents tab of the territory topology view display options.
3. Display the topology view
4. To customize the display of the topology view or specify the territories to include
5. Click the start trace button.
While the trace is running, the start trace button is replaced with the stop trace button. Stop the trace by clicking the stop trace button.

**To define how to display data in the territory topology view**
1. Display the topology view
2. Click the Display Settings link.
3. On the Display Options tab, set the following fields:
4. Click Save to save your changes.
5. Click OK.

**To identify the territories to display in the topology view**
1. Display the topology view
2. Click the Display Settings link.
3. Click the Node Grouping tab.
4. Click New Group to define a new node grouping
5. On the Add Topology Display Group page, set the following fields
6. Click OK to close the Add Topology Display Group page and return to the Node Grouping tab on the Territory Display Options page.
   Click OK on the Territory Display Options page.

**To edit the settings for an existing node grouping**
1. Display the topology view
2. Click the Display Settings link.
3. Click the Node Grouping tab.
4. Click Edit for the node grouping you want update.
5. On the Edit Topology Display Group page, update the Match Type, Match Name, Visible, and/or Combined fields.
6. Click OK to close the Edit Topology Display Group page and return to the Node Grouping tab on the Topology Display Options page.
7. Click OK on the Territory Display Options page.

**Deleting Node Groupings**
If you no longer need a node grouping, you can delete it.

**To delete node groupings**
1. Display the topology view
2. Click the Display Settings link.
3. Click the Node Grouping tab.
4. Select the check boxes in the rows of the node groupings you want to delete.
5. Click Delete.
6. Click OK on the Territory Display Options page.

**To order the node groupings**
1. Display the topology view
2. Click the Display Settings link.
3. Click the Node Grouping tab.
4. Use the buttons in the Move Up and Move Down columns to put the groupings in the order you want.
5. Click OK.
To view the documents that the system will trace
1 Display the topology view
2 Click the Display Settings link.
3 Click the Documents tab.
4 Click OK.

Managing Broker Security

The setup and usage of basic authentication, SSL authentication, keystore files, truststore files, and Access Control Lists (ACLs), which are key elements of the Broker security model are covered in detail. Step-by-step instructions are provided to guide you through the basic authentication, SSL, FIPS, and CRL configuration and implementation process for all your Broker system components.

Broker Security Model

The Broker security model is designed to safeguard all major components of your webMethods Broker installation as well as Broker data. The Broker security model uses:

- Basic authentication to authenticate Broker Servers and clients. Authentication is the process of validating the identity of an entity attempting to establish a connection. Basic authentication provides a light-weight mechanism to authenticate Broker Servers and clients with users existing in the operating system, Lightweight Directory Access Protocol (LDAP), or Active Directory Service Interfaces (ADSI).
- SSL to authenticate Broker Servers and clients. The reference integrity through SSL guarantees the identity of a Broker Server to a requesting client and that of a client to a Broker Server.
- FIPS to provide additional security.
- CRL to provide increased security.

A certificate revocation check against a CRL provides a mechanism for protecting against using certificates that are compromised.

- Access Control List (ACL) to do following:
  - Authorize administrative access to a Broker Server and client access to Broker objects and data.
  - Allow secure sharing of data between Brokers in territories, clusters, or gateways.

You can also set access permissions to control which Brokers can join a territory or a cluster.

For using ACL, Broker Server must be configured with one of the following:

- Basic Authentication
- Basic Authentication with SSL
- SSL (and optionally configure FIPS, or CRL, or both)
- Broker Server Port and Broker Monitor Port security.

You can specify individual IP addresses to which the Broker Server port and Broker Monitor port must bind to. You do this by specifying a value for the broker-ipaddress parameter in the awbroker.cfg file and the monitor-ipaddress parameter in the awbrokermon.cfg file.

Authentication Identities

A key concept of the Broker security model is that of an identity. An identity represents the credentials of any entity requesting authentication or authorization within a Broker installation.

- For basic authentication, an identity is composed of user name authenticated by the alias (authentication system such as LDAP and ADSI) configured in the basicauth.cfg file.
- For SSL authentication, an identity is composed of SSL user DN and the SSL issuer DN.

An entity can be:

- A Broker Server
- The Broker user interface (or Broker admin components)
- Java, JMS, and C# client programs attempting to access the Broker
Access Control Lists (ACLs)

There are two ACLs you can attach to a Broker component:

- A user ACL, which is a list of basic authentication user names and SSL user DNs. For example, “brokeruser” and “CN=Broker Server, O=My Company”.
- An authenticator ACL, which is a list of basic authentication system alias names and DNs for certification authorities or CAs (issuers of the user certificates). For example, “BrokerLDAP” and “CN=My Company SSL Issuer, O=My Company”.

You configure ACLs to achieve one or more of the following:

- To restrict a client's administrative access to the Broker Server (for example, whether a client can stop or restart a Broker Server)
- To control whether a client can add or delete Brokers
- To control whether clients have access to the system-defined admin client group
- To grant permissions for a Broker to access other Brokers in a territory.
- To grant permissions for Brokers in one Broker territory to access remote Brokers in another territory through a Broker gateway
- To grant permissions for a Broker to access other Brokers in a cluster
- To grant permissions for Brokers in one Broker cluster to access remote Brokers in another cluster through a Broker cluster gateway.

Securing Broker System Components

The Broker security model uses ACLs with basic authentication or SSL to secure the following Broker system components:

- Broker Server
- Clients, including the Broker user interface and Broker command-line utilities
- Remote Brokers in a territory
- Remote Brokers in a territory gateway
- Remote Broker in a cluster
- Remote Brokers in a cluster gateway

Securing the Broker Server

The Broker Server is the first Broker component you secure using basic authentication or SSL.

Securing Broker Command-Line Utilities

When basic authentication or SSL authorization is used, most Broker command-line utilities must present a valid basic authentication identity or SSL identity before connecting (only server_config, server_conf_restore, and broker_start do not require an identity).

To simplify administration using SSL, it is recommended that you use the same keystore, truststore, and DN that you use for the Broker user interface.

Securing Client Groups

A client application that participates in the authentication process must still be granted authorization to access specific document types. Assigning individual clients to client groups with particular sets of access permissions accomplishes that purpose. For example, you may want to retain a document type, such as "expenses," that is only accessible to clients belonging to the Broker client group "finance."

Securing Territories and Gateways

You can use a territory ACL to require permissions for a Broker to join and participate in a territory, and a gateway ACL to control whether Broker data is allowed to flow from a connected territory across a gateway.
Securing Clusters and Cluster Gateways
You can use a cluster ACL to require permissions for a Broker to join and participate in a cluster, and a cluster gateway ACL to control whether Broker data is allowed to flow from a connected cluster across a cluster gateway.

Securing Broker Server Using Basic Authentication
Broker supports basic authentication to authenticate Broker clients by using the user name and password. You can include basic authentication users in an ACL to control access to the protected Broker objects. Use basic authentication to secure Broker Servers, remote Brokers in a territory or a cluster, and remote Brokers in a territory gateway or a cluster gateway.

Enabling Basic Authentication
You must enable basic authentication on Broker, if you want Broker to authenticate Broker clients using the basic authentication credentials (user name and password).

To enable basic authentication on Broker
1. Set up a user authentication system. Broker authenticates the users against this user directory. Users can be defined in one or more of the following authentication system:
   - Operating system
   - LDAP
   - ADSI
Note: Broker does not handle administrative tasks such as changing passwords, creating, deleting, or modifying users.
2. Configure the authentication mechanism by specifying the configuration parameters for the authentication systems (for example, LDAP and ADSI) in the basic authentication configuration file (basicauth.cfg).
3. Configure the basic authentication property in the awbroker.cfg file to point to the basicauth.cfg file, if it is not already set. For example, set basic-auth-cfgfile=

Basic Authentication Configuration File
You will find a sample basic authentication configuration file, in the following location:
%BROKER_HOME%\config folder
logfile=basicauth.log
loglevel=0

# Sample OS Authentication entry
#alias=LocalOS
#authtype=OS
#os-win-auth-user-exist=true
#defaultdomain=mydomain

# Sample LDAP Authentication entry
#alias=LDAP1
#authtype=LDAP
#servername=ldap://myserver.domain:389
#ldap-person-base-binddn=ou=users,o=myorganization
#ldap-group-base-binddn=ou=groups,o=myorganization
#ldap-person-objectclass=organizationalPerson
#ldap-group-objectclass=groupOfUniqueNames
#ldap-group-prs-attr=uniqueMember
#ldap - server type=OpenLDAP
#ldap - userid field=uid
#ldap - groupid field=cn
#ldap - allow domain-as-base bind dn=true
#ldap - person property attr=cn, sn
#ldap - group property attr=cn
#resolve groups=ru

# Sample LDAP Authentication entry
#alias=LDAP2
#authtype=LDAP
#serverhost=myserver.domain
#serverport=389
#defaultdomain=mydomain
#ldap - person base binddn=ou=mycountry, dc=ad, dc=mycompany
#ldap - group base binddn=ou=mycountry, dc=ad, dc=mycompany
#ldap - person objectclass=user, OrganizationalPerson, person, top
#ldap - group objectclass=group, top
#ldap - group prs attr=memberOf
#ldap - server type=ActiveDirectory
#ldap - userid field=samAccountName
#ldap - groupid field=cn
#ldap - allow domain-as-base bind dn=true
#ldap - person property attr=cn, sn, uid, description
#ldap - group property attr=cn, description
#resolve groups=ru

# Sample ADSI Authentication entry
#alias=ADSI1
#authtype=ADSI
#serverhost=myserver.domain
#defaultdomain=mydomain
#adsi - forest dn=dc=ad, dc=mycountry

You need to manually maintain the basic authentication configuration file on the file system. After making changes to the file, you must restart Broker Server for the changes to take effect. When a client connects to Broker Server and passes the credentials (user name, and password), Broker Server authenticates them against the configured aliases in sequence. If an alias is not accessible, or authentication fails against that alias, Broker Server tries to authenticate the user against the next alias in the list. If authentication fails for all aliases, the connection request is rejected.

### Configuring Basic Authentication Identity for a Broker Server

Use the following procedure to configure the basic authentication identity for a Broker Server. This is the outbound identity that the Broker Server uses to identify itself to other Brokers in a territory or a cluster, or to a Broker at the other end of the gateway. Configure the basic authentication identity for a Broker Server with ACL secured territory or cluster, and/or gateway.

**To assign a basic authentication identity to a Broker Server**

2. In the Broker Servers list, click the server on which to assign the basic authentication identity. If the server does not appear in the list, use the Search tab to locate it.
3 In the Broker Server Details page, click the Server Identity tab.
4 Click the Basic Identity tab.
5 Click Change Configuration.
6 Type the user name and password in the designated fields.
7 Click Apply.
8 Check that the Status field reads: Enabled

To disable the basic authentication identity for a Broker Server
1 In My webMethods: Messaging > Broker Servers > Servers.
2 In the Broker Servers List, click the server with the basic authentication identity you want to disable. If the
server does not appear in the list, use the Search tab to locate it.
3 Click the Server Identity tab.
4 Click the Basic Identity tab.
5 Click Disable Basic Identity.

Securing Broker Server Using SSL
Broker supports SSL to safeguard your Broker Servers and clients. SSL allows secure sharing of data
between clients and Broker Servers, and between Brokers in territories, clusters, or gateways.

Broker Security Model and OpenSSL
SSL on the Broker is built using the OpenSSL library, an open-source implementation of the SSL/TLS
protocol and general source cryptographic library. OpenSSL includes a command-line tool for SSL
operations such as managing certificate files. You can use the OpenSSL command-line tool along with
My webMethods when configuring and managing SSL for Broker.

Assigning SSL Identities for a Broker Server
An SSL identity is composed of a user's SSL signed certificate and the trusted root of the certificate issuer
(or authenticator). Information about an identity is stored in two certificate files: a keystore containing a
private key/signed certificate pair and the truststore containing the trusted root.
The SSL identity must be presented whenever:
- An attempt is made to connect to the Broker Server SSL port
- A request is made to access Broker Server administrative functions or a Broker component where an
ACL has been configured.

One-Way and Two-Way SSL Authentication
The Broker security model allows you implement one-way or two-way SSL authentication between the
Broker Server and a client.
- **One-way SSL authentication.** A client with truststore but without keystore. If you configure the client
for one-way SSL authentication, you establish a much higher level of security than with a non-SSL
connection. In one-way authentication, the identity of the Broker Server is authenticated by the client, and
must be guaranteed through the Broker Server’s SSL certificate before a connection is made. You can
optionally encrypt this connection through settings on the client for an even higher degree of security.
You can also configure basic authentication with one-way SSL authentication.
- **Two-way SSL authentication.** A client with both keystore and truststore. If you configure
the client for two-way SSL authentication so that both the Broker Server and the client connecting to the
Broker Server must be SSL authenticated, you can fully implement the Broker security model. You will
be able to configure ACLs to protect data and access to Broker administrative functions. With two-way
authentication, each client must be associated with a signed digital certificate in order to establish an SSL
connection. You can optionally encrypt this connection through settings on the client for an even higher
degree of security.
SSL Authentication and Broker Port Usage

Depending on the client's SSL configuration, clients connect to Broker Server through one of the three Broker Server ports available for client connections. The base port of the Broker Server is used for non-SSL connections. The port that is numbered one less than the base port is used when the client is configured for one-way SSL authentication. The port numbered two less than the base port is used when the client is configured for two-way SSL authentication.

For example, if the default port 6849 is the base port of the Broker Server, port 6849 is used for non-SSL connections, port 6848 is used for one-way SSL connections, and the port 6847 is used for two-way SSL connections.

Broker Server Identity and SSL

The Broker will start enforcing ACLs on protected Broker objects when the Broker Server has established an identity for itself. At this point, anonymous clients are no longer able to access ACL-protected Broker objects.

The following figure illustrates how client-to-Broker Server connections work when the Broker Server has an identity.

Client-to-Broker connections: Broker Server SSL identity configured and enabled

When a Broker Server and client authenticate each other's SSL identities, a connection is made through the (two-way) SSL port. That port's number is two below that of the Broker Server's assigned port. Under these circumstances, a client can access any Broker object protected by an ACL, so long as that client's identity is listed in the object's ACL.

Clients that do not present an SSL identity to the Broker Server for authentication (through the non-SSL port) cannot access Broker objects protected by an ACL (illustrated by the crosses). The following figure illustrates how client-to-Broker Server connections work when the Broker Server does not have an SSL identity (or its SSL identity is disabled).

Client-to-Broker connections: no Broker Server SSL identity

not enforced. Any client that can connect to the Broker Server can access any of its ACL-protected Broker objects. For the SSL port to be used, the Broker Server must have an identity. Thus, any attempt to connect to the Broker Server by first authenticating its identity will fail (illustrated by the crosses). Only non-SSL connections will be successful.
Certificate Files
webMethods Broker stores its SSL certificates in keystore files and the trusted roots for the certificates in truststore files.

Keystore File
webMethods Broker uses a special file called a keystore to store each Broker component's SSL certificate. A keystore file contains one key pair, composed of a client's private key and signed certificate for its public key. The certificate contains the client's distinguished name (DN). This DN is usually associated with an alias in the Broker user interface and referred to there as the user name. The keystore should be strongly protected with a password, and stored (either on the file system or elsewhere) so that it is accessible only to administrators.

Keystore File Formats
Before creating a keystore file to store a certificate, you need to know which file format to use.
- You can use the PKCS#12 certificate format for the keystore for any Broker component: the Broker Server, the Broker admin component, or a client. PKCS#12 is a commonly used, standardized, certificate file format that provides a high degree of portability.
- You can use PEM (Privacy Enhanced Mail) format for the keystore for the Broker Server. You cannot use this format for the keystore for the Broker admin component or a client. PEM is a base-64 encoded data format used for text-based communications; it provides the ability to encrypt the data before encoding it.

Certificate Chains
It is possible for a single-entity certificate to have a list of signing certificates leading up to the original, self-signed root certificate. Such a certificate list, with each certificate signed by the next, is termed a certificate chain.

With a certificate chain, it is necessary to validate each subsequent signature in the list until a trusted CA certificate is reached. For Java and JMS clients that use SSL with Broker (Entrust certificate format), you must include the entire chain of certificates in a keystore and truststore. (For Broker clients using OpenSSL, that restriction is not necessary.) Also, any root CA certificates in use by clients must be in the Broker Server's truststore.
Installing a Certificate
To install a signed certificate, you generate a certificate request using a certificate editing tool and send the request to the CA (the same issuing entity as specified on the trusted root). After receiving the X.509-compliant signed certificate back from the CA, you again use the certificate editor and install the certificate into the keystore. Setting up a keystore file is one of the first steps needed for configuring a Broker component for SSL.

Truststore File
webMethods Broker uses a file called a truststore to store trusted roots. A trusted root is a certificate belonging to a Certification Authority (CA). The trusted root contains the CA's public key and the distinguished name of the issuing entity (referred to as the authenticator name in the Broker user interface). A truststore may contain the trusted roots for an entire certificate chain. Truststores should be stored in a safe place on the file system to prevent unauthorized access and tampering.

Truststore File Formats
The allowable formats for truststores are different than those for keystores; PKCS#12 cannot be used as the format for truststores because it cannot hold multiple certificates.
- The format for the Broker Server is PEM.
- The format of truststores for Java client programs, JMS client programs, and the Broker admin component is Java keystore (JKS). You create and manage JKS truststores at the command line using keytool, Sun Microsystems' Java certificate editor.
- You manage the truststore for C# clients through the Microsoft Management Console (MMC). Setting up a truststore file is one of the first steps in configuring SSL for a Broker component.

How Broker Uses a Keystore and Truststore
For a Broker component to be SSL authenticated, it must have a valid, authorized X.509 certificate installed in a keystore file and the trusted root for the CA that issued the certificate installed in a truststore file. The following figure illustrates these requirements and the relationship between the two files.

Example Truststore File and Keystore File Showing Relationship
As shown in the above figure, the same truststore file can contain multiple trusted roots. These trusted roots may be associated with numerous keystore files. A keystore file contains the key pair for a single Broker component, and can contain the entire certificate chain required for a Broker component’s authentication.

Protecting Keystore and Truststore Files
Keystore and truststore files exist within the file system of your operating system, and since these are critically important files, you want to maintain them in a secure directory path. If either of the certificate files for a Broker component cannot be located, SSL authentication will be disabled and the connection may be switched to an open port. It is recommended that only users serving as My webMethods or Broker administrators have access to these certificate files.

Using SSL Encryption
You can encrypt SSL authenticated connections to the Broker Server as well as SSL authorized communications at the client group ACL level. Broker uses a standard SSL 128-bit encryption key. The SSL support in webMethods Broker uses RSA encryption algorithms.

Configuring SSL for Broker Server
1. Create the keystores.
2. Create the truststores.
3. Configure the Broker Server for SSL.
4. Configure the Broker user interface component for SSL.
5. Configure each client to enable SSL.
You use certificate editing tools to create and manage the keystores and truststores for clients, and use the client applications to assign the SSL identities and perform any additional SSL configuration.

Creating Keystores and Truststores
Broker does not supply its own keystore or truststore files. You must create your own, and configure and manage them.

Loading Keystores and Truststores into My webMethods
After you create the keystores and truststore(s) and move them to a secure location in your file system, you load the keystores and truststores to My webMethods. You can then manage SSL identities through the same user interface as your other My webMethods applications.

Load a keystore (from the file system of the same machine used to administer Broker)
1. In My webMethods: Messaging > Settings > SSL Keystore.
2. Click the Upload Keystore tab.
3. Type an alias for the keystore in the Keystore Name box.
4. Click Browse and locate the keystore file.
5. Click Upload.

To load a keystore (from the file system of the machine hosting My webMethods Server)
1. In My webMethods: Messaging > Settings > SSL Keystore.
2. Click the Add Keystore tab.
3. Type an alias for the keystore in the Keystore Name box.
4. Type the full path name of the keystore in the Keystore File box.
5. Click Add

To load a truststore (from the file system of the same machine used to administer Broker)
1. In My webMethods: Messaging > Settings > SSL Truststore.
2. Click the Upload Truststore tab.
3. Type an alias for the truststore in the Truststore Name box.
4. Click Browse and locate the truststore file.
5. Click Upload.
To load a truststore (from the file system of the machine hosting My webMethods Server)
1 In My webMethods: Messaging > Settings > SSL Truststore.
2 Click the Add Truststore tab.
3 Type an alias for the truststore in the Truststore Name box.
4 Type the full path name of the truststore in the Truststore File box.
5 Click Add.

Modifying Keystores and Truststores
To download, modify, and reload a keystore (truststore)
1 In My webMethods: Messaging > Settings > SSL Keystore (SSL Truststore).
2 Click Download Keystore (Download Truststore).
3 On the Download Keystore (Download Truststore) panel, click Keystore Name (Truststore Name) and select the keystore (truststore) from the list.
4 Click Download.
5 Save the keystore (truststore) file.
6 Edit the keystore (truststore) using a certificate editor.
7 Reload the keystore (truststore) into the Broker user interface.

Configuring an SSL Identity for a Broker Server
You set either an SSL Identity for the Broker Server in My webMethods. To configure an SSL identity for a Broker Server, you open its keystore, select the DN to use as the Broker Server's identity, and assign that DN to the Broker Server.

To assign an SSL identity to the Broker Server
1 In My webMethods: Messaging > Broker Servers > Servers.
2 In the Broker Servers List, click the server on which to assign an identity. If the server does not appear in the list, use the Search tab to locate it.
3 In the Broker Server Details page, click the Server Identity tab.
4 Click the SSL tab. The Status should read: Secure Sockets Layer needs to be configured
5 Click Change Configuration.
6 On the Change Broker SSL Settings page, do one of the following:
   - If the keystore containing the identity is on a different machine than the one hosting Broker Server:
     1 Click the Remote Keystore tab.
     2 Type the full path name of the Remote SSL Keystore in the box.
     3 Type the full path name of the Remote SSL Truststore in the box.
   - If the keystore containing the identity is on the same machine as the one hosting Broker Server:
     1 Click the Local Keystore tab.
     2 Click Local SSL Keystore and select a keystore name from the list.
     3 Click Local SSL Truststore and select a truststore name from the list.
     7 Click Keystore Type and select the file format of the keystore.
     8 Click Truststore Type and select the file format of the truststore.
     9 The following options are available for advanced users:
        a Select an SSL Protocol to use for authentication (the default is All).
        b Enter the cryptographic Cipher Suite to use for authentication (the default is ALL:eNULL).
        c Reconfigure the maximum Verification Depth allowable for a certificate chain (the default is 9).
     10 Type the Password to the keystore file.
     11 Click Get User Name to retrieve the user name (user DN) from the keystore.
     12 Click Save.
     13 Click the Identity tab.
Click the SSL tab on the Broker Server Details page for the Broker Server that you just configured. Information about the SSL configuration is displayed, and the Status should read: Secure Sockets Layer is configured and working.

**To disable the SSL identity for a Broker Server**
2. In the Broker Servers List, click the server with the SSL connection you want to disable. If the server does not appear in the list, use the Search tab to locate it.
3. Click the Identity tab.
4. Click the SSL tab.
5. Click Disable SSL.

**Securing Broker Server Using CRL**
A certificate revocation list (CRL) is a list of certificates (or more specifically, a list of serial numbers for certificates) that are revoked by a certification authority (CA) and are no longer valid. A CRL provides increased security because it provides a mechanism for protecting against certificates that are compromised. The implementation of SSL with Broker Server supports the checking of certificates against the CRL.

**To configure certificate revocation checking by using CRLs**
2. Configure SSL.
3. In the Broker Servers List, click the Broker Server on which to configure revocation checking. If the Broker Server does not appear in the list, use the Search tab to locate it.
4. Click the SSL tab and then click the Change Configuration button.
5. Click Local Keystore tab.
6. In the Certificate Revocation List File field, type the full path name of the CRL file.
   - Note: You must place the CRL file in a location accessible to the Broker Server.
7. In the Certificate Revocation List File Type field, select either PEM or DER file format depending on the CRL type.
8. Click Apply.
9. Repeat steps 5 through 8 to configure certificate checking for remote keystore by clicking the Remote Keystore tab in step 5.

**Configuring CRL Checking For the Entire Certificate Chain**
To enable CRL checking for the entire certificate chain, set the check-crl-all configuration parameter in the Software AG_directory\Broker\data\awbrokers82\default\awbroker.cfg file as shown below:
```plaintext
check-crl-all=1
```
By default, the CRL checking is not done for the entire certificate chain.

**To start Broker Server in FIPS mode**
1. Stop Broker Server.
2. In Broker data directory > awbroker.cfg file.
3. If the value of enable-fips-mode is set to 0 (default), change it to 1.

**Client Group ACLs**
You use client group ACLs to identify the clients that have permission to join a particular client group. Once a client becomes a member of the group, it can produce or consume the document types in the can-publish and can-subscribe lists, and various logs controlled by the client group.
The following figure illustrates how client group ACLs work.
How a Client Group Access Control List Works with SSL Clients

In the figure, three SSL-configured clients are shown attempting to join the client group "payroll," which has access to company payroll information. Each of these clients is identified both by a unique user DN and the DN of its certification authority.

**admin Client Group ACLs**

The admin client group is a system-defined client group with full access permissions for a given Broker. Each Broker has one admin client group, and you can enable ACLs for that client group, the same as with any other client group. A client belonging to the admin client group has administrative access to a given Broker. Members of the admin client group can modify any Broker setting, and reconfigure the territories or local gateways to which the Broker belongs.

**Broker Server ACLs**

By attaching ACLs to a Broker Server, it is possible to limit administrative access to a Broker Server and the Brokers it controls. Clients without ACL access to a Broker Server can only view certain types of information about Broker Server, such as the names of its Brokers, including the default Broker, statistical information, and whether basic authentication or SSL is configured.

**Territory ACLs**

You use territory ACLs to control which Brokers can join a territory. When a territory has an ACL attached, the basic authentication identity or the SSL identity of any Broker attempting to join must be listed on the territory ACL. If not, the Broker will be denied permission.

**How a Territory Access Control List Works**

In the figure, territory X consists of three Brokers, A, B, and C, that are controlled by their respective Broker Servers A, B, and C. The basic authentication identities or SSL identities for each of the Broker Server's controlling Brokers A, B, and C are listed in the ACL for territory X. For Broker Z (belonging to Broker Server Z) to join territory X, its identity (shaded DN in the ACL), which is the same as that of Broker Server Z, must be added to the territory ACL.
Cluster ACLs
You use cluster ACLs to control which Brokers can join a cluster. When a cluster has an ACL attached, the SSL identity or the basic authentication identity of any Broker attempting to join must be listed on the cluster ACL. If not, the Broker will be denied permission.

Territory Gateway ACLs
A territory gateway allows a Broker in one territory to retrieve information from Brokers in another territory. You can set up ACLs on a territory gateway so that the Broker requesting information from a territory other than its own must be granted authorization to receive data.

When you set up a territory gateway ACL, the user name ACL on the local gateway contains one entry: the basic authentication identity or SSL identity of the Broker Server on which the remote gateway resides. After permission is granted, information flows from the remote Brokers on the other side of the gateway. A territory gateway ACL is not shared between the two sides of a gateway connection. Each gateway requires its own ACL.

The following figure shows the setup of a territory gateway ACL for a one-way flow of information. In the figure, an ACL containing the identity of the Brokers in territory Y is attached to a requesting Broker from territory X.

How a Territory Gateway ACL Works
Cluster Gateway ACLs

A cluster gateway allows a Broker in one cluster to retrieve information from Brokers in another cluster. You can set up ACLs on a cluster gateway so that the Broker requesting information from a cluster other than its own must be granted authorization to receive data. A cluster gateway ACL is not shared between the two sides of a gateway connection. Each gateway requires its own ACL.

About Configuring Broker Server ACLs

The point of using Broker Server ACLs is to control access to the administrative functions of the Broker Server. Once basic authentication or SSL is enabled for a Broker Server, you can use the Broker user interface to configure ACLs that limit administrative access to that server.

To add authenticator names to a Broker Server ACL

1 In My webMethods: Messaging > Broker Servers > Servers.
2 In the Broker Servers List, click the server on which to attach the ACL. If the server does not appear in the list, use the Search tab to locate it.
3 On the Broker Server Details page, click the ACL tab, then click Authenticator Names.
4 Click Add Authenticator Names.
5 To add an authenticator name for basic authentication:
   - Click the Enter Authenticator Name tab, type the basic authenticator alias in the Authenticator Name box, and click Add Authenticator Name. Repeat for each authenticator alias you want to add.
6 To add an authenticator name for SSL authentication, do one of the following:
   - To add an authenticator name from the Broker Server keystore:
     - Click the Select Authenticator Name tab, check the Authenticator Names for issuers for whom to grant permission, and click Add Authenticator Names.
   - To add an authenticator name from a different keystore:
Click the Enter Authenticator Name tab, type the DN of the authenticator in the Authenticator Name box, and click Add Authenticator Name. Repeat for each authenticator DN you want to add. An ACL is automatically enabled after you add entries; no additional actions are needed.

**To add user names to a Broker Server ACL**
2. In the Broker Servers List, click the server on which to attach the ACL. If the server does not appear in the list, use the Search tab to locate it.
3. On the Broker Server Details page, click the ACL tab, then click User Names.
4. Click Add User Names.
5. To add user names for basic authentication:
   - Click the Enter User Name tab, type the basic authentication user name in the User Name box, and click Add User Name. Repeat for each user you want to add.
6. To add user names for SSL authentication, do one of the following:
   - To add user names from the Broker Server keystore:
     - Click the Select User Names tab, check those User Names for whom to grant permission, and click Add User Names.
   - To add user names from a different keystore:
     - Click the Enter User Name tab, type the user's DN in the User Name box, and click Add User Name. Repeat for each user you want to add.
   An ACL is automatically enabled after you add entries; no additional actions are needed.

**To disable a Broker Server ACL**
2. In the Broker Servers List, click the server with the ACL. If the server does not appear in the list, use the Search tab to locate it.
3. On the Broker Server Details page, click the ACL tab, then click either:
   - Authenticator Names, to display the list of authenticators belonging to the Broker Server authenticator name ACL.
   - User Names, to display the list of users belonging to the Broker Server user name ACL.
4. Click Disable Authenticator Names or Disable User Names.

**To delete authenticator and user names from Broker Server ACLs**
2. In the Broker Servers List, click the server with the ACL(s). If the server does not appear in the list, use the Search tab to locate it.
3. On the Broker Server Details page, click the ACL tab.
4. Do one of the following:
   - **To delete authenticator names**:
     1. Click Authenticator Names.
     2. Check the boxes next to the Authenticator Names you want to remove from the authenticator ACL.
   - **To delete user names**:
     1. Click User Names.
     2. Check the boxes next to the User Names you want to remove from the user ACL.
5. Click Delete.

**To set access permissions to the Broker Server for a new SSL client**
1. Add the client's trusted root to the truststore file used by the Broker Server and its signed certificate to a new Broker Server keystore.
3 In the Broker Server List, click the server with the ACL(s) you want to update. If the server does not appear in the list, use the Search tab to locate it.
4 To add the client's user DN to the ACL:
   a On the Broker Server Details page, click the ACL tab, then click User Names.
   b Click Add User Names.
   c Click the Select User Names tab, check the User Name (DN) for the new client, and click Add.
5 To add the client's issuer DN to the ACL:
   a On the Broker Server Details page, click the ACL tab, then click Authenticator Names.
   b Click Add Authenticator Names.
   c Click the Select Authenticator Names tab, check the Authenticator Name (DN) for the new client, and click Add.

To register user names on a client group ACL
1 In My webMethods: Messaging > Broker Servers > Client Groups.
2 Select the Broker Server and Broker of the client group whose ACL you want to configure.
3 Click Go. If the client group does not appear in the list, use the Search tab to locate it.
4 Under Client Groups, click the client group whose ACL you want to configure.
View the Configure page for the client group. If the Access Control status reads Broker Administrator identity required, you need to configure the Broker user interface identity settings before continuing.
5 Click the ACL tab for the client group.
6 Click Add User Names.
7 To add user names for basic authentication:
   Click the Enter User Name tab, type the basic authentication user name in the User Name box, and click Add. Repeat for each user you want to add.
8 To add user names for SSL authentication, do one of the following:
   To add the user name in the Broker Server SSL keystore:
   Click the Select User Name tab, click the User Name for whom to grant permission, and click Add.
   To add user names from a different keystore:
   Click the Enter User Name tab, type the user DN in the User Name box, and click Add.
   Repeat for each user you want to add.

To register authenticator names on a client group ACL
1 In My webMethods: Messaging > Broker Servers > Client Groups.
2 Select the Broker Server and Broker of the client group whose ACL you want to configure.
3 Click Go. If the client group does not appear in the list, use the Search tab to locate it.
4 Under Client Groups, click the client group whose ACL you want to configure. View the Configure page for the client group. If the Access Control status reads Broker Administrator identity required, you need to configure the Broker user interface component identity settings before continuing.
5 Click the ACL tab for the client group.
6 Click Add Authenticator Names.
7 To add an authenticator name for basic authentication:
   Click the Enter Authenticator Name tab, type the alias of the authenticator in the Authenticator Name box, and click Add. Repeat for each authenticator alias you want to add.
8 To add an authenticator name for SSL authentication, do one of the following:
   To add an authenticator name from the Broker Server keystore:
   Click the Select Authenticator Name tab, check the Authenticator Names for issuers for whom to grant permission, and click Add.
   To add an authenticator name from a different keystore:
   Click the Enter Authenticator Name tab, type the DN of the authenticator in the Authenticator Name box, and click Add. Repeat for each authenticator DN you want to add.
The status message Authenticator name(s) added to access control list appears on the Client Group Details page. The ACL is automatically enabled after you add entries; no additional actions are needed.
To specify which Brokers can join a territory (using Broker user name DNs)
1 In My webMethods: Messaging > Broker Territories > Territory Brokers.
2 In the Territories List, click the territory on which to create an ACL. If the territory is not in the list, use the Search tab to locate it.
3 On the Territory Broker Details page, click the Territory link.
4 Click the ACL tab.
5 Click User Names (if not already selected), then click Add User Names.
6 On the Add User Names to Territory ACL page, do the following:
   - To add user names for basic authentication:
     Click the Enter User Name tab, and in the User Name box, type the basic authentication user name of the Broker Server whose Broker or Brokers are permitted to join and participate in the territory. Click Add. Repeat for any additional Broker Servers whose Broker or Brokers you also want to join.
   - To add user names from the Broker Server SSL keystore:
     Click the Select User Names tab, check the User Names belonging to the Broker Servers whose Brokers are permitted to join and participate in the territory, and click Add.
   - To add user names from a different keystore:
     Click the Enter User Name tab, and in the User Name box, type the DN of the Broker Server whose Broker or Brokers are permitted to join and participate in the territory. Click Add. Repeat for any additional Broker Servers whose Broker or Brokers you also want to join.
   The status message User name(s) added to access control list appears on the Territory Details page.

To specify which Brokers can join a territory (using Broker authenticator names)
1 In My webMethods: Messaging > Broker Territories > Territory Brokers.
2 In the Territories List, click the territory containing the ACL you want to configure. If the territory is not in the list, use the Search tab to locate it.
3 On the Territory Details page, click the ACL tab.
4 Click Authenticator Names, then click Add Authenticator Names.
5 On the Add Authenticator Names to Territory ACL page, do the following:
   - To add authenticator names for basic authentication:
     Click the Enter Authenticator Name tab, type the authenticator's alias in the Authenticator Name box, and click Add. Repeat for each authenticator for whom to grant permission.
     For SSL authentication, do one of the following:
     - To add authenticator names from the Broker Server SSL truststore:
       Click the Select Authenticator Names tab, check the Authenticator Names for whom to grant permission, and click Add.
     - To add authenticator names from a different truststore:
       Click the Enter Authenticator Name tab, type the authenticator's DN in the Authenticator Name box, and click Add. Repeat for each authenticator for whom to grant permission.
   The status message Authenticator name(s) added to access control list appears on the Territory Details page.

To delete Brokers from a territory ACL
1 In My webMethods: Messaging > Broker Territories > Territory Brokers.
2 In the Territories List, click the territory, click the territory containing the ACL from which you want to delete Brokers.
3 On the Territory Details page, click the ACL tab.
4 Do one of the following:
   - To delete authenticator names:
     1 Click Authenticator Names.
     2 Check the boxes next to the Authenticator Names you want to remove from the authenticator ACL.
   To delete user names:
1. Click User Names.
2. Check the boxes next to the User Names (for the Broker Servers whose Brokers are participating in the territory) that you want to remove from the user ACL.
3. Click Delete.

**To specify which Brokers can join a cluster (using Broker user names)**

1. In My webMethods: Messaging > Broker Clusters > Cluster Brokers.
2. In the clusters list, click the cluster on which to create an ACL. If the cluster is not in the list, use the Search tab to locate it.
3. On the Cluster Details page, click the ACL tab.
4. Click User Names (if not already selected), then click Add User Names.
5. On the Add User Names to Cluster ACL page, do one of the following:
   - To add user names for basic authentication:
     1. Click the Enter User Name tab.
     2. In the User Name box, type the basic authentication user name of the Broker Server whose Broker or Brokers are permitted to join and participate in the cluster.
     3. Click Add.
     4. Repeat for any additional Broker Servers whose Broker or Brokers you also want to join.
   - For SSL authentication, to add user names from the Broker Server SSL keystore:
     1. Click the Select User Names tab.
     2. Check the User Names belonging to the Broker Servers whose Brokers are permitted to join and participate in the cluster.
     3. Click Add.
     4. Repeat for any additional Broker Servers whose Broker or Brokers you also want to join.

The status message User name(s) added to access control list appears on the Cluster Details page.

**To specify which Brokers can join a cluster (using Broker authenticator names)**

1. In My webMethods: Messaging > Broker Clusters > Cluster Brokers.
2. In the clusters list, click the cluster containing the ACL you want to configure. If the cluster is not in the list, use the Search tab to locate it.
3. On the Cluster Details page, click the Cluster link.
4. Click the ACL tab.
5. Click Authenticator Names, then click Add Authenticator Names.
6. On the Add Authenticator Names to Cluster ACL page, do one of the following:
   - To add authenticator names from a basic authentication configuration file:
     1. Click the Enter Authenticator Name tab, type the authenticator alias in the Authenticator Name box, and click Add. Repeat for each authenticator for whom to grant permission.
   - For SSL authentication, to add authenticator names from the Broker Server SSL truststore:
     1. Click the Select Authenticator Names tab, check the Authenticator Names for whom to grant permission, and click Add.
   - For SSL authentication, to add authenticator names from a different truststore:
     1. Click the Enter Authenticator Name tab, type the authenticator’s DN in the Authenticator Name box, and click Add. Repeat for each authenticator for whom to grant permission.

The status message Authenticator name(s) added to access control list appears on the Cluster Details page.

**To delete Brokers from a cluster ACL**

1. In My webMethods: Messaging > Broker Clusters > Cluster Brokers.
2 In the clusters list, click the cluster, and then click the cluster containing the ACL from which you want to delete Brokers.
3 On the Cluster Details page, click the ACL tab.
4 Do one of the following:
   - To delete authenticator names:
     1 Click Authenticator Names.
     2 Check the boxes next to the Authenticator Names you want to remove from the authenticator ACL.
   - To delete user names:
     1 Click User Names.
     2 Check the boxes next to the User Names (for the Broker Servers whose Brokers are participating in the cluster) that you want to remove from the user ACL.
     3 Click Delete.

**To configure a Broker gateway ACL (using Broker user names)**
1 In My webMethods: Messaging > Broker Territories > Broker.
2 Under Gateway, click the gateway whose ACL you want to configure.
3 On the Territory Gateway Details page, click the ACL tab.
4 Click Add User Names if not already selected.
5 Type the basic authentication user name or SSL user DN in the Remote Broker's User Name box.
6 Click the Enable user name box.
7 Click Apply.

**To configure a Broker gateway ACL (using Broker authenticator names)**
1 In My webMethods: Messaging > Broker Territories > Broker.
2 Under Gateway, click the gateway whose ACL you want to configure.
3 On the Territory Gateway Details page, click the ACL tab.
4 Click Add Authenticator Names, if not already selected.
5 Type the authenticator alias or DN in the Remote Broker's User Name box.
6 Click Apply.

**To configure a cluster gateway ACL (using Broker user names)**
1 In My webMethods: Messaging > Broker Clusters > Cluster Brokers.
2 Under Gateway, click the gateway whose ACL you want to configure.
3 On the Cluster Gateway Details page, click the ACL tab.
4 Click Add User Names if not already selected.
5 Type the basic authentication user name or SSL user DN in the Remote Broker's User Name box.
6 Click the Enable user name box.
7 Click Apply.

**To configure a cluster gateway ACL (using Broker authenticator names)**
1 In My webMethods: Messaging > Broker Clusters > Cluster Brokers.
2 Under Gateway, click the gateway whose ACL you want to configure.
3 On the Cluster Gateway Details page, click the ACL tab.
4 Click Add Authenticator Names if not already selected.
5 Type the authenticator alias or DN in the Remote Broker's User Name box.
6 Click Apply.

**ACLs and Migration Issues**
Because of the changes to certificate files and their usage, ACLs cannot be migrated from earlier versions of webMethods Broker to 7.1. You cannot import version 6.x ACL information into 7.1; you need to reconstruct the ACLs from scratch.
Broker Security and My webMethods Server

The following figure shows the Broker security model within the context of My webMethods applications. In the figure, the lines represent connections, the shaded padlocks illustrate Broker connections that can be SSL-enabled, and the white padlock illustrates the connection between the local machine and the My webMethods Server machine, which can be SSL-enabled.

Broker SSL authentication and My webMethods Server authentication

The connection between the local machine hosting your browser and the My webMethods Server machine is established with user name and password authentication. This connection (which can be made with HTTPS, or HTTP over SSL) is under the control of My webMethods security, and is not part of the Broker security model.

Basic authentication or SSL enabled connections under Broker control can exist between:
- A Broker client and the Broker Server.
- The Broker user interface and the Broker Server. (To the Broker Server, the Broker user interface is simply another client.)

You set either a basic authentication identity or an SSL Identity for the Broker Server in My webMethods.

**To assign a basic authentication identity to the Broker user interface component**

1. In My webMethods: Messaging > Settings > Broker Administrator Identity.
2. If no identity is set, click No identity has been set. Click here to set the identity.
3. In the Change Identity Setting page, select Username/Password from Client Authentication list box.
4. In the Username text field, type your basic authentication user name.
5. In the Password text field, type your basic authentication password.
6. Optionally, you can configure encryption.
   a. Select the SSL Truststore. The truststore you use for the Broker user interface must be located on the machine hosting your My webMethods applications.
   b. Select the SSL truststore's File Type (certificate file format) from the SSL Truststore Type list.
   c. Click the Yes option for Encryption.
7. Click Connect.

**Configuring an SSL Identity for the Broker User Interface Component**

To Broker Server, the Broker user interface component is a just another client (for example, a Java program) requesting a secure connection. Thus, you assign an SSL identity to the Broker user interface and configure SSL as you would with any SSL client. That allows the Broker Server to authenticate the
Broker user interface at connection time. Establishing an SSL identity for the Broker user interface and connecting to the Broker Server (with its SSL identity configured) is a requirement for enabling Broker security.

**To assign an SSL identity to the Broker user interface component**

**Note:** If you configure SSL identity when basic authentication identity has already been configured, the basic authentication identity settings will be disabled.

1. Make sure you have the file location and password of the SSL keystore for the Broker user interface before starting this procedure.
3. If no identity is set, click No identity has been set. Click here to set the identity.
4. On the Change Identity Setting page, select SSL from the Client Authentication list box.
5. Select the SSL keystore file containing the identity you want to use from the SSL Keystore list. The keystore you use for the Broker user interface must be located on the machine hosting your My webMethods applications.
6. Select the SSL keystore’s file type (certificate file format) from the SSL Keystore Type list.
7. Type the SSL Keystore Password, then click Get User Names to retrieve the DN from the keystore. The user DN from the keystore file displays next to User Name.
8. Select the SSL Truststore that contains the trusted root of the certificate in the selected keystore. The truststore you use for the Broker user interface must be located on the machine hosting your My webMethods applications.
9. Select the SSL truststore’s File Type (certificate file format) from the SSL Truststore Type list.
10. If you want to enable encryption, click the Yes option for encryption.
11. Click Connect.

**Configuring for One-way SSL Authentication**

When you configure the client for one-way SSL authentication, you only need to specify a truststore and not a keystore. The identity of the Broker Server is authenticated by the client, and must be guaranteed through the Broker Server’s SSL certificate before a connection is made. You can optionally encrypt this connection.

**To configure one-way SSL authentication for the Broker user interface component**

1. In My webMethods: Messaging > Settings > Broker Administrator Identity.
2. On the Change Identity Setting page, select None from the Client Authentication list box.
3. Select the SSL Truststore that contains the trusted root of the certificate in the selected keystore. The truststore you use for the Broker user interface must be located on the machine hosting your My webMethods applications.
4. Select the SSL truststore’s File Type (certificate file format) from the SSL Truststore Type list.
5. If you want to enable encryption, click the Yes option for encryption.
6. Click Connect.

**About Restoring the Identity of the Broker User Interface Component’s Identity**

For security reasons, the identity of the Broker user interface is persisted only for the duration of a session. Each time you logon to Broker, you must restore the identity (SSL or basic authentication) of the Broker user interface component. If you created an identity for the Broker user interface in a previous session, you can restore it.

**To restore the Broker user interface component’s identity**

1. In My webMethods: Messaging > Settings > Broker Administrator Identity.
2. Click Use Current Identity.
3. Restore the identity, as follows:
   - If you are restoring a two-way SSL identity, type the Password for the listed SSL Keystore.
   - If you are restoring a basic authentication identity, type the Password for the user name displayed.
4. Click Connect.
Disabling the Identity of the Broker User Interface Component

While being in an administrative session on the Broker user interface component, you can disable the identity that is currently set for the Broker user interface component. After you disable the identity, all the Broker Server operations such as adding a Broker, changing SSL settings, and changing basic authentication settings, that are protected by Broker Server ACL will no longer be available for anonymous logons.

To disable the Broker user interface component's SSL identity or basic identity:
1. In My webMethods: Messaging > Settings > Broker Administrator Identity.
2. Click Change Identity.
3. On the Change Identity Setting page, select None from the Client Authentication list box.
4. Click Connect.

Converting Certificate Files

Keystore files created for Broker version prior to 7.1 will not work with the current version of Broker Servers. For those keystore files to work, they must be modified to a format accessible by Broker 7.1. The Broker Certificate Conversion utility, a command-line utility helps automate this process. It converts a pre-version 7.1 keystore file to the new format required by Broker.

Broker Certificate Conversion Utility

The Broker Certificate Conversion utility ConvertCert.jar is a command-line tool that allows you to upgrade pre-version 7.1 keystore files to the new keystore file format.

Single Certificate Format

Keystores contain only a single certificate or certificate chain, whereas keystores older than version 7.1 may contain multiple certificates. If there are multiple certificates (or certificate chains), the Conversion utility generates a directory that contains as many new keystore files as there are certificates in the pre-7.1 keystore.

Trusted Roots and Truststore Files

Unlike pre-version 7.1 keystores, keystores used by Broker 7.1 and later do not contain trusted roots. The Conversion utility removes the trusted roots from the pre-7.1 keystore files upon creating the new keystore file(s), but does not generate a truststore file. These truststore files must be created manually using a certificate editing utility.

Downloading the Broker Certificate Conversion Utility

The utility can be downloaded from the Empower Product Support Web site. The file name of the utility is ConvertCert.jar.

Important! You must include the file enttoolkit.jar in the CLASSPATH for the Certificate Conversion Utility to work.

Command Usage

The following shows the syntax and usage for ConvertCert.jar:

```
java -jar ConvertCert.jar { export | list } old-keystore-file password output-dir
```

<table>
<thead>
<tr>
<th>Part</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>export or list</td>
<td>Whether to export the certificates in the old keystore file to newly created keystore(s), or display a list of certificates in the old keystore file in the command window.</td>
</tr>
<tr>
<td>old-keystore-file</td>
<td>Name of the keystore file to be migrated.</td>
</tr>
<tr>
<td>password</td>
<td>Password for the old keystore file.</td>
</tr>
<tr>
<td>output-dir</td>
<td>Name of the directory in which the new keystore(s) will be created.</td>
</tr>
</tbody>
</table>
Managing Territories

Introduction to Territories

A territory is a group of inter-connected Brokers that functions as one logical Broker. All Brokers in a territory maintain the same set of document types and client groups, however, they each support their own set of clients.

A client can connect to any Broker in the territory and publish documents to or receive documents from any other client in the territory. The Broker to which a client publishes documents takes care of forwarding those documents to subscribers that are connected to other Brokers in the territory. To an individual publisher or subscriber, a territory appears as one large Broker.

An Example Territory Made Up of Four Brokers

Brokers that operate in a territory are peers. There is no "central" or "controlling" Broker. Each Broker in the territory is responsible for notifying its peers of changes (e.g., additions, deletions, or modifications) that occur to its document types or client groups. By dynamically propagating changes to all Brokers in this manner, document types and client groups remain synchronized across the entire territory.

Why Implement a Territory?

If you were to keep adding clients to a single Broker, that Broker would, at some point, begin running out of resources such as memory, disk space, or socket connections. A territory enables you to scale a Broker to handle large numbers of clients by spreading the client population across multiple host machines. For additional flexibility, you can create multiple territories and link them together using gateways. A gateway is a connection that allows specified documents to flow between two territories.

How Brokers in a Territory Communicate with Each Other

Each Broker in a territory is directly connected to each of its peers. Each Broker in a territory maintains a Remote Broker object for each of the Brokers to which it is connected. This object acts as the server-side state object for the Broker at the other end of the connection.

In this capacity, the Remote Broker object serves the same function as does a client state object for a regular Broker client program. Like a client state object, the Remote Broker object maintains a list of subscriptions and a queue for the remote Broker. However, unlike an ordinary client state object, a Remote Broker object also functions as a client of the remote Broker. In this capacity, the Remote Broker object continually retrieves documents for the local Broker from its queue on the remote Broker.
Each Broker in a territory maintains a Remote Broker object for each of its peers

 Territory EAST

The Queue Maintained by a Remote Broker

A Remote Broker object also maintains a queue, called the *forward queue*. The forward queue contains instances of documents that match subscriptions in the Remote Broker object's subscription list. When the remote Broker is connected to its Remote Broker object, it continually retrieves documents from this forward queue and distributes the documents to its own clients.

A retriever thread gets documents from the forward queue and distributes the documents to local clients

A Remote Broker object is similar to an explicit-destroy client-state object in that it remains instantiated even if the remote Broker disconnects or goes offline. When the remote Broker comes back online, it automatically reconnects to its peers and retrieves documents from its forward queues.

Certain conditions can cause a Remote Broker object to become *orphaned*, meaning that the Remote Broker object exists for a Broker that has been permanently deleted or removed from the territory. An orphaned Remote Broker object will continue to collect documents in its forward queue and can potentially consume a great deal of storage if it is not administratively removed.
How Delivered Documents Are Handled in a Territory

A Broker will not forward a delivered document to other Brokers in a territory unless the document is addressed to a client by its fully-qualified client ID (in the form /territoryName/BrokerName/ClientID). If a Broker receives a delivered document that is addressed to an unqualified client ID, it delivers the document locally. If the addressee does not exist on the local Broker, the document is sent to the dead-letter queue (if one is configured) or is discarded.

How Brokers Synchronize Metadata

To function as one logical Broker, all the Brokers in a territory must have identical document-type and client-group information. When a Broker initially joins a territory, its document types and client groups (including ACL settings) are synchronized with the other Brokers in the territory. The subscription lists in the Remote Broker objects are also populated at this time.

After a Broker becomes a member of a territory, it is responsible for notifying its peers of any changes that occur to its document types or client groups. When a Broker in a territory receives such notification, it immediately applies the change to its own metadata. Because these metadata changes are self-propagating, you can make a change to a document type or client group on any Broker and that change will automatically be applied to all Brokers in the territory.

Securing a Territory

Within a territory, either all Brokers use SSL or no Brokers use SSL. You cannot mix SSL-enabled Broker Servers and non-SSL-enabled Broker Servers in the same territory. Similarly, encryption, when enabled, is enabled for all connections between Brokers in the territory. You cannot selectively enable encryption between certain Brokers. When the Broker Servers in a territory are SSL-enabled, you can use an access control list (ACL) to identify which Brokers are authorized members of the territory. Brokers that are not listed in the ACL, are not allowed to join or participate in the territory.
To create a territory
1 In My webMethods: Messaging > BrokerTerritories > Brokers.
2 Click Add Territory.
3 In Territory Name, type a name for the territory.
   The name you choose must be unique among all territories that will be connected using gateways.
4 Use the following steps to select one of the Brokers that will participate in this territory:
   a In the Broker Server field, select the Broker Server that hosts the Broker.
   b Use the Selected Brokers controls to specify the Brokers that you want to add. This list, check whether it
      is already a member of another territory.
5 Click OK.

Adding Brokers to a Territory
Use the following procedure to add one or more Brokers to an existing territory. (This action is also
known as “joining” a territory.)
To add a Broker to a territory
1 In My webMethods: Messaging > BrokerTerritories > Brokers.
2 In the Territory Brokers List, click Add Broker.
3 From the Territory to Join list on the Add Territory Broker page, select the territory to which you want to
   add a Broker. (If a territory is listed multiple times, you can select any of the entries for it.)
4 Use the following steps to select the Brokers to add to the territory.
   a In the Broker Server field, select the Broker Server that hosts the Brokers that you
      want to add.
   b Use the Selected Brokers controls to specify the Brokers that you want to add to
territory.
5 Click Join.

To remove a Broker from a territory
1 In My webMethods: Messaging > Broker Territories > Brokers.
2 In the Territory Brokers List, select the check box beside the Broker that you want to remove from a
   territory.
3 Click Leave Territory.

Deleting a Territory
You can dissolve a territory by simply removing every Broker from the territory. When the last Broker
has been removed from the territory, the territory will cease to exist.

Viewing Information about a Remote Broker Object
Use the following procedure to view the Territory Broker Details tab, which displays the list
of remote Broker objects that a Broker maintains. The information on this tab indicates:
- Whether the remote Broker is currently connected to the local Broker
- The number of documents that reside in the forward queue for the remote Broker and
  the time at which the Remote Broker last retrieved documents from this queue
- The number of documents that the local Broker has recently retrieved from its forward queue on the
  remote Broker and the time at which it last retrieved documents from that queue

To view information about a Remote Broker object
1 In My webMethods: Messaging > Broker Territories > Brokers.
2 In the Territory Brokers List, click the Broker on which the Remote Broker object resides.
The list at the bottom of the Territory Broker Details tab displays the following information for each Remote
Broker object on the Broker.
Remote Broker  The name of the Remote Broker object. This name identifies the remote Broker that the Remote Broker object represents. You can click this name to open the Territory Broker Information page for the remote Broker.

Linked  Indicates the state of the local Broker's connection to the remote Broker as follows:
- Yes.  The remote Broker is currently connected to the local Broker.
- No.  The remote Broker is not currently connected to the local Broker.

Forward Queue Length  The number of documents currently queued for the remote Broker.
Forward Queue Size  The size of the queue, in bytes.
Recent Local Receipts  The number of documents that the local Broker has retrieved from the remote Broker during the last statistics collection interval.
Last Local Receipt  The time at which the local Broker last retrieved documents from the remote Broker.
Recent Remote Deliveries  The number of documents that the remote Broker retrieved from its forward queue during the last statistics collection interval.
Last Remote Delivery  The time at which the remote Broker last retrieved documents from its forward queue.
Keep Alive  The frequency (in seconds) at which the local Broker issues a 'keep-alive' message to this remote Broker. Keep-alive messages are generally used to maintain a connection between Brokers that are separated by a firewall. They prevent a firewall from disconnecting what it considers to be an idle connection. Click the keep-alive value if you want to change it. A keep-alive interval of 0 seconds disables the keep-alive feature. When disabled, the Keep Alive column for the Remote Broker object will display "Disabled."

Deleting a Remote Broker Object
You can use the following procedure to delete a Remote Broker object from a Broker. If the remote Broker that the Remote Broker object represents is online, this procedure will remove that Broker from the territory. If the remote Broker is not online, this procedure removes the Remote Broker object from the local Broker and from every other Broker in the territory.

To delete a Remote Broker object
1 In My webMethods: Messaging > Broker Territories > Brokers.
2 In the Territory Brokers List, click the Broker on which the Remote Broker object resides.
3 In the Territory Broker Details tab, select the check box beside the Remote Broker object that you want to delete.
4 Click Remove from Territory.

Managing Territory Gateways
A territory gateway connection enables two territories to exchange documents. Unlike territories, whose constituent Brokers automatically share their document types and client groups, a territory gateway connection requires an administrator to explicitly specify which document types the territories are permitted to send and receive. Moreover, the territories never share client group information.

Territory Gateway Connections, Territory Gateways, and Territory Gateway Pairs
The term territory gateway connection refers to the link that enables two territories to exchange documents. To create this link, two Brokers, one from each territory, are designated to communicate with one another. To form a territory gateway connection, you must create and configure a gateway on each of these Brokers. A territory gateway performs functions similar to those performed by a Remote Broker object in a territory. In this capacity, the territory gateway:
- Establishes and maintains a connection with its counterpart in the remote territory
- Maintains the subscription list and forward queue for the remote territory
- Retrieves documents for the local territory from the remote gateway.
Collectively, the two territory gateways that operate together to form a territory gateway connection are known as a **gateway pair**. A territory gateway connection will not function until both Brokers have been equipped with a territory gateway and both territory gateways in the pair are configured correctly.

**Territory Gateway Names**

In the Broker user interface, a territory gateway is identified by a name. As shown below, the name of the territory gateway is composed from the names of the two territories that the territory gateway connects: `localTerritory > remoteTerritory`

Where `localTerritory` is the name of the territory where the territory gateway resides and `remoteTerritory` is the name of the territory to which the territory gateway connects. The gateway pair shown in the diagram above consists of a gateway named "AAA \> BBB" and gateway named "BBB \> AAA."

**Connecting Multiple Territories with Territory Gateway Connections**

If you use territory gateway connections to link multiple territories, be aware that the set of connected territories must form a graph that has no cycles (that is, a path that traverses the graph should not be able to return to its beginning). Visually, an acceptable graph looks like a tree.

The following figure shows a graph that crosses the boundary between two administrative domains. With the correct permissions set at each territory gateway, client 1 and client A can exchange documents with each other.
Creating a Territory Gateway Connection

To link territories with a territory gateways connection, you must perform the following high-level steps:

1. Identify the list of document types that each territory needs to receive from the other and make sure that these document types exist in both territories. If a required document type does not exist in one of the territories, the administrator of that territory must add it.

2. Identify the two Brokers that will host the territory gateways. The domain name service (DNS) used by the two host machines must be capable of bi-directional name resolution, meaning that the DNS must be able to resolve the name of the host machine to the correct IP address and also be able to resolve the machine's IP address back to the correct name.

3. Determine what type of authentication and encryption is required between the territories and make sure that the Broker Servers that host the two gateways support these requirements. For example, if you want to secure each gateway with an access control list (ACL), both Broker Servers must be SSL-enabled (that is, have an SSL identity) and their keystores must be equipped with the trusted root and signed digital certificate for the Broker Server that hosts their counterpart in the gateway pair.

4. Create and configure each territory gateway in the gateway pair.

When you configure a territory gateway, you must do the following:

a. Specify the name and location of the Broker that hosts the territory gateway in the remote territory.

b. Assign "Allow Receive" permission to each document type that the local territory needs to receive.

c. Assign "Allow Forward" permission to each document type that the remote gateway’s territory needs to receive. (If necessary, you can include a filter that permits only specified instances of the document type to pass to the remote territory.)

Creating a "One-Way" Territory Gateway

Organizations often want to create what is known informally as a "one-way" territory gateway, meaning that they want documents to flow only in one direction. For example, a territory that handles employee-related documents might want to broadcast certain documents to, but never receive documents from, a territory that handles customer support documents.

To create this type of territory gateway, you must create a complete gateway connection (that is, you must create and configure both gateways in the gateway pair). Creating a one-way territory gateway does not mean that you create only one side of a territory gateway connection.

Creating a Territory Gateway Pair if You Control Both Brokers

It is easiest to configure a territory gateway if you control both sides of the gateway. In this case, My webMethods can do much of the work for you. You only have to create one territory gateway. My webMethods automatically creates and configures its counterpart in the remote territory.

There are two major steps in creating a territory gateway pair using this procedure:

1. First, you must create the territory gateway pair.

2. Next, you must configure the list of document types that the territories are permitted to exchange.

To define a territory gateway pair:

1. In My webMethods: Messaging > Broker Territories > Territory Gateways.

2. Click Add.

3. On the Create Gateway Pair tab, select the two Brokers that will form the territory gateway connection.

4. Click Create.

Specifying Permissions for Receiving Individual Document Types

Use the following procedure to set document types that a territory gateway can receive from another territory.

To select individual document types that a territory gateway can receive from the other territory:

1. In My webMethods: Messaging > Broker Territories > Territory Gateways.

2. In the Territory Gateways List, locate the territory gateway that you want to configure and click Details.

3. Click the Shared Document Types tab.
To select the document types that this territory gateway Broker can receive from the other territory, do the following:

a. Click Allow Receive and then click Add Receive Document Types.
b. On the Allow Receive Document Types panel, select the document types that this territory gateway will receive from the other territory.
c. Click Add.

To select individual document types that a territory gateway can forward to the other territory:

1. In My webMethods: Messaging > Broker Territories > Territory Gateways.
2. In the Territory Gateways List, locate the territory gateway that you want to configure and click Details.
3. Click the Shared Document Types tab.
4. To select the document types that this territory gateway can forward to the other territory, do the following:
   a. Click Add Forward Document Types.
   b. On the Allow Forward Document Types panel, select the document types that this territory gateway will be allowed to forward to the other territory.
   c. Click Add.

To use client groups to specify which document types can flow to and from a territory gateway:

1. In My webMethods: Messaging > Broker Territories > Territory Gateways.
2. In the Territory Gateways List, locate the territory gateway that you want to configure and click Details.
3. Click the Shared Document Types tab and then click Hook Up Client Groups.
4. Use the Available Client Group controls to select the client group(s) that contain the document type definitions and permissions that you want to apply to this territory gateway Broker.
5. Select one of the following options and click Update.

Creating a Territory Gateway Pair if You Control Only One Broker

Following steps:
1. Create the territory gateway for your territory.
   You will need the following information from the other administrator.
   - The exact name of the remote territory
   - The name of the Broker that will host the territory gateway in the remote territory
   - The host name and port number of the Broker Server on which that Broker resides
2. Configure the list of document types that your territory gateway can exchange with the remote territory.

To define one side of the territory gateway pair:
1. In My webMethods: Messaging > Broker Territories > Territory Gateways.
2. Click Add.
3. Click the Create Gateway tab, and complete the following fields:
   - Local Territory, Remote Territory, Remote Broker Server, Broker Server
4. Click Create.

Specifying which Documents the Territory Gateway Can Forward and Receive

To place a filter on a document type:
1. In My webMethods: Messaging > Broker Territories > Territory Gateways.
2. In the Territory Gateways List, locate the gateway that you want to configure and click Details.
3. Click the Shared Document Types tab.
4. In the list of the document types that the territory gateway is permitted to receive, click Edit Filter for the document type that you want to filter.
5. In the Filter text box, type a filter string that specifies the criteria that identifies which documents this territory gateway can forward to the remote territory.
6. Click OK.
   The filter appears in the Filter column in the Allow Receive table.
Viewing Information about a Territory Gateway

The Territory Gateways page displays a list of territory gateways that are known to the Broker Servers in your Broker Server List.

To view information about a territory gateway

1. In My webMethods: Messaging > Broker Territories > Territory Gateways.

The Territory Gateways List displays the following information about each territory gateway that resides on a Broker Server that is part of your Broker Server List.

Viewing Information about Documents Flowing between the Gateways

To view information about documents flowing through a territory gateway

1. In My webMethods: Messaging > Broker Territories > Territory Gateways.
2. Click Details to display the Territory Gateway Details page.

The Queue Length field on the Configuration tab reports the number of documents currently in the forward queue for the remote territory gateway.
3. If you want information about the number of documents that the remote territory has received from the local territory gateway, click the Remote Broker tab and examine the following statistics:

To delete a territory gateway connection

1. In My webMethods: Messaging > Broker Territories > Territory Gateways.
2. In the Territory Gateways List, select the check box beside the territory gateway(s) that you want to delete.
3. Click Delete.

Removing an Allowed Document Type from a Territory Gateway

To remove a document type from a territory gateway

1. In My webMethods: Messaging > Broker Territories > Territory Gateways.
2. Click Details to display the Territory Gateway Details page and then click the Shared Document Types tab.
3. If you want to remove a document type from the list of types that this territory gateway can receive, click Allow Receive.
4. Select the check box beside the document type that you want to remove from the allowed list.
5. Click Delete.
6. Select options and click Update.

Pausing and Resuming a Territory Gateway

You can temporarily halt the flow of documents from a territory gateway to a remote territory by pausing the territory gateway. When you pause a territory gateway, the gateway locks the forward queue that it maintains for the remote territory. This action prevents the remote territory from retrieving documents from this territory gateway.

No documents are lost when you pause a territory gateway. The forward queue on the territory gateway continues to accumulate documents for the remote territory, but these documents will not flow to the other territory until the territory gateway is resumed. Be aware that pausing a territory gateway halts the flow of outbound documents (documents that originated from the territory on which the territory gateway resides and are being forwarded to subscribers in the remote territory). It does not prevent that territory gateway from receiving documents from the remote territory. To halt the flow of inbound documents, you must pause the territory gateway in the remote territory, too.

To pause a territory gateway

1. In My webMethods: Messaging > Broker Territories > Territory Gateways.
2. Click Details for the territory gateway whose outbound flow of documents you want to pause.
3. On the Configuration tab, click Pause Forwarding.

The Status field switches to Paused and indicates the ID of the client that placed the pause on the territory gateway.
To resume a territory gateway
1 In My webMethods: Messaging > Broker Territories > Territory Gateways.
2 Click Details for the territory gateway whose outbound flow of documents you want to restore.
3 On the Configuration tab, click Resume Forwarding.
The Status field switches to Active, indicating the outbound documents are permitted to flow to the other territory.

Preventing Connection Timeouts between Territories
If your territory gateway connects to a remote territory through a firewall, you might need to activate the territory gateway's "keep-alive" feature to prevent the firewall from dropping the connection after a period of inactivity. When you enable the keep-alive feature, the territory gateway publishes "keep alive events" to the remote territory at a specified frequency. The keep-alive events prevent the connection from appearing idle to the firewall.

Viewing and Modifying the Keep-Alive Setting
Use the following procedure to view or modify the keep-alive setting.
To view or modify the keep-alive setting
1 In My webMethods, Messaging > Broker Territories > Territory Gateways.
2 Click Details for the territory gateway whose keep-alive setting you want to configure.
3 On the Configuration tab, examine the Keep Alive Interval field.

To view or configure the forwarding mode
1 In My webMethods: Messaging > Broker Territories > Territory Gateways.
2 Click Details for the territory gateway whose forwarding option you want to view or configure.
3 On the Configuration tab, enable or disable the Static Gateway Forwarding option.

Refusing Document-Type Updates
Territories that are connected by a territory gateway connection automatically exchange and apply updates to the document types that they are configured to exchange. For example, if territory A is configured to receive a document type "CancelOrder" from territory B, and an administrator in either territory modifies that document type, the change is automatically propagated to the other territory.

Viewing or Configuring the Global Refuse Update Option
Use the following procedure to view or configure the global Refuse Update option.
To view or configure the global Refuse Update option
1 In My webMethods: Messaging > Broker Territories > Territory Gateways.
2 Click Details for the territory gateway whose update option you want to view or configure.
3 On the Configuration tab, enable or disable the Refuse all shared document type updates.
4 Click Apply.

To view or configure the Refuse Update option for individual document types
1 In My webMethods: Messaging > Broker Territories > Territory Gateways.
2 Click Details for the territory gateway whose document types you want to view or configure.
3 Click the Shared Document Types tab and use the Allow Receive and Allow Forward links to display the list in which the document type appears. If the document type appears in both lists, you can configure the update option on either list.
4 To determine whether the territory gateway currently accepts or rejects updates for the document type, examine the value in the Refuse Updates column.
5 If you want to change the document type's Refuse Update setting, do the following:
a Select the check box beside the document type whose update option you want to change.
b Click Refuse Update or Allow Update, depending on whether you want the territory gateway to reject or permit updates to the selected document type.
Managing Clusters

A cluster is a group of Brokers that functions as one logical Broker. All Brokers in a cluster maintain the same set of document types and client groups; however, they each support their own set of clients. Clients can connect to a cluster and publish documents to or receive documents from a cluster. The cluster to which a client publishes documents takes care of forwarding those documents to subscribers that are connected to the cluster. To an individual publisher or subscriber, a cluster appears as one large Broker.

Why Implement a Cluster?

When the publishing load increases, you scale up client-side message distribution by creating a cluster of Brokers to handle the increasing load. Clusters enable scalability and availability of your messaging system. A cluster is similar to a territory, except that document forwarding within a cluster is disabled. The JMS subscriber is connected to all the Brokers in a cluster, and receives the documents published to any of the Brokers in the cluster. Clusters facilitate client-side load balancing and failover when you use JMS cluster policies for client connections. For additional flexibility, you can create multiple clusters and link them together using cluster gateways. A cluster gateway is a connection that allows specified documents to flow between two clusters.

How Delivered Documents Are Handled in a Cluster

A Broker will not forward a delivered document to other Brokers in a cluster unless the document is addressed to a client by its fully-qualified client ID (in the form/clusterName/BrokerName/ClientID). If a Broker receives a delivered document that is addressed to an unqualified client ID, it delivers the document locally. If the addressee does not exist on the local Broker, the document is sent to the dead-letter queue (if one is configured) or is discarded.

Creating Clusters

This section explains how to create clusters.

Requirements Checklist

Before you can create a cluster, make sure the following requirements are met:

- The domain name service (DNS) used by the host machines that participate in the cluster must be capable of bi-directional name resolution. This means that your DNS must be able to resolve the name of the host machine to the correct IP address and also be able to resolve the machine's IP address back to the correct name.
- Each Broker must have a name that is unique within the cluster.
- Each cluster must have a unique name.
- A Broker must not belong to another cluster. To become part of the new cluster, it must first leave its current cluster.
- All Broker Servers that participate in the cluster must have the same SSL capabilities. That is, they must all be SSL-enabled or non-SSL enabled.

To create a cluster

1. In My webMethods: Messaging > Broker Clusters > Clusters.
2. Click Add Cluster.
3. In Cluster Name, type a name for the cluster.
4. Use the Select Brokers controls to specify the Brokers that you want to add.
5. Click OK.

To add a Broker to a cluster

1. In My webMethods: Messaging > Broker Clusters > Cluster Brokers.
2. In the Broker Clusters List, click Add Broker.
3. From the Cluster to Join list on the Add Cluster Broker page, select the cluster to which you want to add a Broker.
4. Use the following steps to select the Brokers to add to the cluster:
   a. In the Broker Server field, select the Broker Server that hosts the Brokers that you want to add.
b Use the Selected Brokers controls to specify the Brokers that you want to add to cluster.
5 Click Join.

**To remove a Broker from a cluster**
1 In My webMethods: Messaging > Broker Clusters > Cluster Brokers.
2 In the Cluster Brokers List page, click the cluster whose Brokers you want to remove.
3 In the Cluster Information page, select the check box beside the Broker you want to remove from the cluster.
4 Click Leave Cluster.

**Deleting a Cluster**
You can dissolve a cluster by simply removing every Broker from the cluster. When the last Broker has been removed from the cluster, the cluster will cease to exist.

**To view the list of known clusters**
- In My webMethods, select: Messaging > Broker Clusters > Cluster Brokers.
The Broker Clusters List displays a list of known clusters and their known members.

---

**Managing Cluster Gateways**

A *cluster gateway* enables two clusters to exchange documents.

**Cluster Gateway Connections, Cluster Gateways, Cluster Gateway Pairs, and Cluster Gateway Brokers**

The term *cluster gateway connection* refers to the link from a local Broker (in the local cluster) to a remote Broker (in the remote cluster) that enables two clusters to exchange documents. A cluster gateway connection refers to an uni-directional link from one cluster to another. To create this link, two Brokers, one from each cluster, are designated to communicate with one another.

A *cluster gateway* consists of multiple cluster gateway connections that allow message flow from the local cluster to a remote cluster. To form a cluster gateway, you must create and configure a cluster gateway on each of these Brokers.

**Note:** The Broker that hosts a cluster gateway is a regular member of its own cluster and can handle regular clients in addition to the cluster gateway.

A *cluster gateway pair* refers to a pair of cluster gateways that allow bi-directional message exchange between two clusters. A cluster gateway will not function properly until all the Brokers have at least one outgoing cluster gateway connection.
A *cluster gateway* Broker performs functions similar to those performed by a Remote Broker object in a cluster. In this capacity, the local cluster gateway Broker:
- Establishes and maintains a connection with its counterpart in the remote cluster
- Maintains the subscription list and forward *queue* for the remote cluster
- Retrieves documents for the local cluster from the remote cluster gateway

### Cluster Gateway Names

In the Broker user interface, a cluster gateway is identified by a name. The name of the cluster gateway is composed of the names of the two clusters that the cluster gateway connects:

localCluster > remoteCluster

where *localCluster* is the name of the cluster where the cluster gateway resides and *remoteCluster* is the name of the cluster to which the cluster gateway connects.

### Creating a Cluster Gateway

To link clusters with a cluster gateway, you must perform the following high-level steps:
1. Identify the list of document types that each cluster needs to receive from the other cluster and *make sure that these document types exist in both clusters*. If a required document type does not exist in one of the clusters, the administrator of that cluster must add it.
2. Identify the Broker pairs that will create cluster gateway connections. *Make sure that each Broker in the local cluster is connected to at least one Broker in the remote cluster*
3. Determine what type of authentication and encryption is required between the clusters.
4. Create and configure each gateway in the cluster gateway pair.

### Creating a Cluster Gateway when You Control Both Sides

It is easiest to configure a cluster gateway if you control both sides of the gateway. In this case, My webMethods can do much of the work for you. You only have to create one side of the cluster gateway. My webMethods automatically creates and configures its counterpart in the remote cluster.

There are two major steps in creating a cluster gateway pair using this procedure.
1. First, you must create the cluster gateway pair.
2. Next, you must configure the list of document types that the clusters are permitted to exchange.

**To define a cluster gateway pair**
1. In My webMethods: Messaging > Broker Clusters > Cluster Gateways.
2. Click Add.
3. On the Create Gateway Pair tab, select the local cluster and remote cluster that will form the cluster gateway connection.
4. Click Edit Selection button to add or remove Broker connections.
5. Click Apply.
6. Select the primary connection for each local Broker gateway.
7. Click Create.

### Specifying Individual Document Types a Cluster Gateway Can Receive

Use the following procedure to set permissions on individual document types for a cluster gateway.

**To select individual document types a cluster gateway can receive from other clusters**
1. In My webMethods: Messaging > Broker Clusters > Cluster Gateways.
2. In the Cluster Gateway List, locate the cluster gateway that you want to configure and click Details.
3. Click the Shared Document Types tab.
   a. Click Allow Receive and then click Add Receive Document Types.
   b. On the Allow Receive Document Types panel, select the document types that this cluster gateway will receive from the other cluster.
c Click Apply.
5 Select one of the following options and click Update.

**To select individual document types that a cluster gateway can forward to the other cluster**
1 In My webMethods: Messaging > Broker Clusters > Cluster Gateways.
2 In the Cluster Gateways List, locate the cluster gateway that you want to configure and click Details.
3 Click the Shared Document Types tab.
4 To select the document types that this cluster can forward to the other cluster across the gateway, do the following:
a Click Allow Forward and then click Add Forward Document Types.
b On the Allow Forward Document Types panel, select the document types that this cluster gateway will be allowed to forward to the other cluster.
c Click Add.
5 Select one of the following options and click Update.

**To use client groups to specify which document types can flow to and from a cluster gateway**
1 In My webMethods: Messaging > Broker Clusters > Cluster Gateways.
2 In the Cluster Gateways List, locate the cluster gateway that you want to configure and click Details.
3 Click the Shared Document Types tab and then click Hook Up Client Groups.
4 Use the Available Client Group controls to select the client groups that contain the document type definitions and permissions that you want to apply to this cluster gateway Broker.
5 Select one of the following options and click Hook Up.

**Defining One Side of a Cluster Gateway Pair**
Use the following procedure to create one side of a cluster gateway.
To define one side of the cluster gateway pair
1 In My webMethods: Messaging > Broker Clusters > Cluster Gateways.
2 Click Add.
3 Click the Create Gateway tab, and complete the fields.
4 Click Edit Selection to select Brokers in the remote cluster that will create gateway connections through the cluster gateway.
5 Click Apply.
6 Select the primary connection.
7 Click Create.

**Filtering Documents across a Cluster Gateway**
A filter string specifies criteria for the contents of a document. Broker uses the filter string to determine which documents match your requirements and allows only those documents to pass through the cluster gateway.
You apply filter strings to those document types that a cluster gateway *can receive* from a remote cluster.
To set a filter on a document type
1 In My webMethods: Messaging > Broker Clusters > Cluster Gateways.
2 In the Cluster Gateway List, locate the cluster gateway that you want to configure and click Details.
3 Click the Shared Document Types tab.
4 In the list of the document types that the cluster gateway is permitted to forward, click Edit Filter for the document type that you want to filter.

**To view information about a cluster gateway**
1 In My webMethods: Messaging > Broker Clusters > Cluster Gateways.
The Cluster Gateways List displays the following information about each cluster gateway you create.
To edit cluster gateway details
1 In My webMethods: Messaging > Broker Clusters > Cluster Gateways.
2 Click Details to display the Cluster Gateway Details page.
3 Click the Connections tab.
4 Click Edit Selection to add or remove a cluster gateway connection.
5 In the Select Remote Broker Connections dialog box, select Brokers in the remote cluster from the Available list to create gateway connections with this local Broker.
6 Click Apply.
7 In the Cluster Gateway Information page, under Primary, select the remote Broker with which you want this local Broker to create a new primary cluster gateway connection.
8 Click Apply.

Viewing Document Details across a Cluster Gateway
Use the following procedure to view information about the activity between the two clusters in the gateway pair.
To view information about documents flowing through a cluster gateway
1 In My webMethods: Messaging > Broker Clusters > Cluster Gateways.
2 Click Details to display the Cluster Gateway Details page.
The Queue Length field on the Configuration tab reports the number of documents currently in the forward queue for all gateway connections in the cluster gateway.

Deleting a Cluster Gateway
To remove a cluster gateway established between two clusters, you delete each member of the cluster gateway pair. If you delete a cluster gateway, all documents that are currently queued for the remote cluster gateway are deleted.
After one side of the cluster gateway is deleted, the other side should be deleted promptly. Otherwise, it will continue to accumulate documents in its forward queue and can eventually consume a considerable amount of memory and queue storage.
To delete a cluster gateway connection
1 In My webMethods: Messaging > Broker Clusters > Cluster Gateways.
2 In the Cluster Gateways List, select the check box beside the cluster gateways you want to delete.
3 Click Delete.
To remove a document type from a cluster gateway
1 In My webMethods: Messaging > Broker Clusters > Cluster Gateways.
2 Click Details to display the Cluster Gateway Details page and then click the Shared Document Types tab.
3 If you want to remove a document type from the list of document types that this cluster gateway can receive, click Allow Receive.
4 Select the check box beside the document type that you want to remove from the allowed list.
5 Click Delete.

Pausing and Resuming a Cluster Gateway
You can temporarily halt the flow of documents from a cluster gateway to a remote cluster by pausing the cluster gateway.
To pause a cluster gateway
1 In My webMethods: Messaging > Broker Clusters > Cluster Gateways.
2 Click Details for the cluster gateway whose outbound flow of documents you want to pause.
3 On the Configuration tab, click Pause Forwarding.
The Status field switches to Paused and indicates the ID of the client that placed the pause on the cluster gateway.
To resume a cluster gateway
1 In My webMethods: Messaging > Broker Territories > Cluster Gateways.
2 Click Details for the cluster gateway whose outbound flow of documents you want to restore.
3 On the Configuration tab, click Resume Forwarding.
The Status field switches to Active, indicating the outbound documents are permitted to flow to the other cluster.

Preventing Connection Timeouts between Clusters
If your cluster gateway connects to a remote cluster through a firewall, you might need to activate the cluster gateway's "keep-alive" feature to prevent the firewall from dropping the connection after a period of inactivity. When you enable the keep-alive feature, the cluster gateway publishes "keep alive events" to the remote cluster at a specified frequency. The keep-alive events prevent the connection from appearing idle to the firewall.

To view or modify the keep-alive setting
1 In My webMethods: Messaging > Broker Clusters > Cluster Gateways.
2 Click Details for the cluster gateway whose keep-alive setting you want to configure.
3 On the Configuration tab, examine the Keep Alive Interval field. It will display either the amount of time between keep-alive events, in seconds, or the word "Disabled" if the keep-alive feature is not activated.
4 If you want to modify or disable the keep-alive setting, do the following:
a Click the current Keep Alive Interval value to display the Change Cluster Gateway Keep Alive Interval page.
b In the Keep Alive Interval (seconds) field, specify the frequency at which you want the cluster gateway to publish keep-alive messages. Specify zero seconds if you want to disable the keep-alive feature.
c Click Apply.

Configuring the Forwarding Mode
A cluster gateway maintains a subscription list that identifies the types of documents the remote cluster wants to receive. This subscription list tells the cluster gateway which documents it must forward to the remote cluster. The forwarding mode determines how the cluster gateway populates and maintains this subscription list.

- In dynamic forwarding mode, a cluster gateway maintains the subscription list for a remote cluster based on subscription events that the remote cluster gateway issues when clients in its cluster add and drop subscriptions to a document type that the clusters share. Broker Servers version 6.5 and earlier use only dynamic forwarding mode.
- In static forwarding mode, a cluster gateway automatically populates the subscription list based on the document types that the cluster gateway is permitted to forward to the remote cluster. Static forwarding mode is the default mode in Broker Servers version 6.5 and later.

To view or configure the forwarding mode
1 In My webMethods: Messaging > Broker Clusters > Cluster Gateways.
2 Click Details for the cluster gateway whose forwarding option you want to view or configure.
3 On the Configuration tab, select or clear the Static Gateway Forwarding check box.

Blocking Document Types Globally
Use the following procedure to configure a cluster gateway to globally refuse document type update.

To view or configure a cluster gateway to globally refuse document type update
1 In My webMethods: Messaging > Broker Clusters > Cluster Gateways.
2 Click Details for the cluster gateway whose update settings you want to view or configure.
3 On the Configuration tab, enable or disable the Refuse all shared document type updates.
4 Click Apply.

**Blocking Document Types Individually**

To view or configure a cluster gateway to refuse individual document type update
1 In My webMethods: Messaging > Broker Clusters > Cluster Gateways.
2 Click Details for the cluster gateway whose document types you want to view or configure.
3 Click the Shared Document Types tab.
4 Click the Allow Forward tab.
5 Under Document Types, select the document types for which you want to configure the Refuse Update option, and then click the Refuse Update button.
6 To determine whether the cluster gateway currently accepts or rejects updates for the document type, examine the value (Yes or No) in the Refuse Updates column.
7 If you want to change the document type’s Refuse Update setting, do the following:
   a Select the check box beside the document type whose update option you want to change.
   b Click Refuse Update or Accept Update, depending on whether you want the cluster gateway to reject or accept updates made to the selected document type.

---

**Export and Import of Broker Metadata**

**Description of Functionality**

Broker includes capabilities that allow you to:
- Export a Broker object and its dependent objects to a file. Exporting the metadata to a file allows you save the configuration settings for an entire hierarchy of related Broker objects (territory, territory gateway, cluster, cluster gateway, Broker Servers, Brokers, client groups, document types, clients), and recreate them at a later time. You can select which objects to include or exclude when exporting metadata.
- Import a saved file of Broker metadata. You can import the saved Broker metadata, effectively re-creating the Broker objects and their relationships. You can import saved metadata to any valid location in your Broker network.

**Using My webMethods for Export and Import**

My webMethods provides a broad range of capabilities for Broker metadata export and import, allowing you to save entire hierarchies of Broker objects and their relationships in XML files. You can import that data in whole or part to other locations in your Broker network.

**XML Files**

My webMethods exports Broker objects to and imports Broker objects from XML files.

**Configuring Export and Import Operations**

In My webMethods, you configure the export and import operations in a table format. The Export and Import tables contain an expandable tree graph with selectable nodes in the first column. These nodes represent discrete elements in Broker object hierarchies, allowing the inclusion or exclusion of Broker objects and data.

**Broker Server and Broker Information**

When exporting or importing objects from the Broker Server level downwards, the hierarchy of Broker objects is as follows:

- Broker Server
- Broker1
Brokers

If you export information at the Broker Server level, you can save information about a specific Broker Server, as well as its Brokers, client groups, clients, and document types, to an XML file. When you select a Broker Server for export, metadata about the Broker Server, such as its description text, ACL settings, and SSL configuration data is included in the XML file.

Brokers

You can generate and relocate the Brokers, client groups, clients, and document types from an export file.

Example of an export display for a territory

Example of an import display for a territory

Using Command-Line Utilities for Export and Import

You can use the Broker command-line utilities to export and import Broker metadata. However, there are several differences between export and import with the command line utilities and export and import with My webMethods:

- The command-line utilities export objects to and import object from ADL-formatted files, whereas My webMethods uses XML files.
Each execution of a command-line utility allows the export and import of metadata for a single Broker; My webMethods allows you to export entire branches of Broker object data from the level of the territory downwards.

With the command-line utilities, you cannot select which dependent objects to include in the saved Broker metadata; you must include or exclude all of them. Using My webMethods, you can select which dependent objects to export or import.

**Exporting Broker Data**

Use the `broker_save` utility from the command line to save Broker configuration information for a specified Broker to an ADL file. The utility includes options that allow you to save configuration information about the territory and Broker Server to which the Broker belongs, the basic authentication/SSL information needed to make a connection, and the basic authentication/SSL information required by a client group Access Control List (ACL).

**Importing Broker Data**

Use the `broker_load` command-line utility to import Broker object configuration data from an ADL file generated by `broker_save`. Once the import has taken place, the Broker objects are re-created at the selected Broker location.

**Exporting a Broker Server and Its Dependent Objects**

You can export metadata for Broker Server and its dependent objects in the format of an XML file.

**To export a Broker Server, Broker, client group, client, or document type**

2. In the Broker Servers List, click the name of the Broker Server containing the Broker object that you want to export. If the Broker Server is not in the list, use the Search tab to locate it.
4. Click the plus (+) tab to expand the nodes for the Broker Server, displaying its dependent objects. Click to expand any dependent nodes.
5. Check the boxes for the objects that you want to include in the export to an XML file.
6. Click Export.
7. In the Export Broker Server page, click Include System Defined Data if you want that data in the export file.
   System-defined data includes information such as the system-supplied document types and client groups.
   You cannot import this data back into Broker; however, you may want to print out the export file to read the information.
8. Click Export.
9. The Export Confirmation Page displays information about the status of the export. Click Proceed.
10. Select whether to Open or Save the XML file in the dialog box and click OK.

**To import a Broker Server, Broker, client group, client, or document type**

2. In the Broker Servers List, click the name of the Broker Server that is the target of the import. If the Broker Server is not in the list, use the Search tab to locate it.
3. On the Broker Server Details page, select the Import tab.
4. Click Upload File.
5. Click Browse and select the XML file to import.
6. Click Upload.
7. In the Import tab, expand the node so as to display its dependent objects. Click to expand any dependent nodes.
   The Target column provides links allowing you to set or change the location of the imported metadata.
8 Select the check boxes for the objects that you want to import.
9 Optionally, click Change Target (if that option is available) to change where to import the data. Then, do one of the following in the Server Import Target panel:
   ▪ Click Select Broker and select a Target Broker Name from the list. My webMethods creates the Broker objects from the import file at the specified location.
   ▪ Click New Broker and type a new Broker name for Target Broker Name. My webMethods will create the Broker objects from the import file at the new Broker location.
10 On the Import tab, click Import.
11 The Import SSL Verification panel prompts you to enter the password for the keystore file. Do one of the following:
   ▪ If SSL authorization is required for the import, enter the Password for the import target Broker Server keystore and click Proceed.
   ▪ If SSL authorization is not required for the import, leave the Password field empty and click Skip SSL Configuration.

To export a territory
1 In My webMethods: Messaging > Broker Territories > Territories.
2 In the Territories List, click the territory that includes the metadata you want to export. If the territory is not in the list, use the Search tab to locate it.
3 On the Territory Details page, click the Export tab.
My webMethods displays the Territory Content in a tree view.
4 Select the check box next to the territory whose metadata you want to export.
5 Click the plus (+) tab to expand the nodes for the territory, and click to expand any dependent nodes.
6 Click Export.
7 On the Export Territory page, click Include System Defined Data if you want that data in the export file. System-defined data includes information such as the system-supplied document types and client groups. You cannot import this data back into Broker; however, you may want to print out the export file to read the information.
8 Click Export.
9 The Export Confirmation Page displays information about the status of the export. Click Proceed.
10 Select whether to Open or Save the XML file in the dialog box and click OK.

To import a territory
1 In My webMethods: Messaging > Broker Territories > Territories.
2 On the Territories page, click Import.
3 On the Territory Import page, check to see whether the export file content is listed. If the file is not listed, do the following:
   a Click Upload File.
   b On the Import Territory page, click Browse and select the XML file to import.
   c Click Upload.
4 In the Territory Import list, locate the territory whose metadata you want to import.
5 The Target Object column provides links that allow you to set or change the location of the imported metadata for the data specified by the row.
6 If you selected Change Target, do one of the following on the Territory Import Broker Target page:
   ▪ If the Broker Server and Broker to which you are importing the metadata already exist, select the Select Broker tab and select a Target Server Name and Target Broker Name from their respective lists. Click OK.
   ▪ If the Broker to which you are importing the metadata is new, select the New Broker tab, select a Target Server Name, and type the name of the new Target Broker Name. Click OK.
Then, on the Territory Import page, reselect the objects to import.
7 Click Import.
To export territory gateway information
1 In My webMethods: Messaging > Broker Territories > Territories.
2 In the Territories List, click the territory that includes the gateway configuration information that you want to export. If the territory is not in the list, use the Search tab to locate it.
3 On the Territory Details page, click the Export tab.
My webMethods displays the Territory Content in a tree view.
4 Click the plus (+) tab to expand the nodes for the territory. Expand the Gateways node to verify that the gateways whose metadata that you want to export is listed.
5 Click the box for the territory object, and for the gateways whose metadata you want to export.
Note: You must select the territory object for the selected gateway(s).
6 Click Export.
7 On the Export Territory page, you can optionally Include System Defined Data in the export file.
8 Click Export.
9 The Export Confirmation Page displays information about the status of the export. Click Proceed.
10 Select whether to Open or Save the XML file in the dialog box and click OK.

To import territory gateway information
1 In My webMethods: Messaging > Broker Territories > Territories.
2 On the Territories page, click Import.
3 On the Territory Import page, check to see whether the export file content is listed. If the file is not listed, do the following:
a Click Upload File.
b On the Import Territory page, click Browse and select the XML file to import.
4 In the Territory Import list, locate the territory that includes the gateway configuration information you want to import.
   If you want to select dependent objects to include or exclude, expand the node for the territory and check the boxes for the objects to include.
5 Make sure the box for Gateways, or one of its dependent nodes, is selected, as well as the territory object to which the gateway belongs.

To export a cluster
1 In My webMethods: Messaging > Broker Clusters > Clusters.
2 In the Clusters List, click the cluster that includes the metadata you want to export. If the cluster is not in the list, use the Search tab to locate it.
3 On the Cluster Details page, click the Export tab.
My webMethods displays the cluster content in a tree view.
4 Click the box next to the cluster whose metadata you want to export.
5 Click the plus (+) tab to expand the nodes for the cluster, and click to expand any dependent nodes.
6 Click Export.
7 The Export Confirmation Page displays information about the status of the export.
a If you want to include system-defined data in the export file, click Include System Defined Data.
System-defined data includes information such as the system-supplied document types and client groups. You cannot import this data back into Broker; however, you may want to print out the export file to read the information.
b Click Export.
8 Click Save in the File Download dialog and provide a name for the exported file. For example, FinanceClusterExport.zip.

To import a cluster
1 In My webMethods: Messaging > Broker Clusters > Clusters.
2 On the Clusters page, click Import.
3. On the Cluster Import page, check to see whether the export file content is listed. If the file is not listed, do the following:
   a. Click Upload File.
   b. On the Import Cluster page, click Browse and select the .zip or the XML file to import.
   c. Click Upload.
4. In the Cluster Import list, locate the cluster whose metadata you want to import. Click Import.

To export cluster gateway information
1. In My webMethods: Messaging > Broker Clusters > Clusters.
2. In the Clusters List, click the cluster that includes the cluster gateway configuration information that you want to export. If the cluster is not in the list, use the Search tab to locate it.
3. On the Clusters Details page, click the Export tab.

My webMethods displays the cluster content in a tree view.
4. Click the plus (+) tab to expand the nodes for the cluster. Expand the Gateways node to verify that the cluster gateways whose metadata that you want to export is listed.
5. Select the check box for the cluster object and for the cluster gateways whose metadata you want to export.
   Note: You must select the cluster object for the selected cluster gateway(s).
6. Click Export.
7. On the Export Cluster page, you can optionally Include System Defined Data in the export file.
8. Click Export.
9. Click Save in the File Download dialog and provide a name for the exported file.

To import cluster gateway information
1. In My webMethods: Messaging > Broker Clusters > Clusters.
2. On the Clusters page, click Import.
3. On the Cluster Import page, check to see whether the export file content is listed. If the file is not listed, then do the following:
   a. Click Upload File.
   b. On the Import Cluster page, click Browse and select the .zip file or the XML file to import.
4. In the Cluster Import list, locate the cluster that includes the cluster gateway configuration information you want to import. If you want to select dependent objects to include or exclude, expand the node for the cluster and select the check boxes for the objects to include.
5. Select the gateway and one of its dependent nodes, as well as the cluster object to which the gateway belongs.
6. Make sure the Target Objects are correct. To change the Target Object, click the corresponding Target Object link, and do the following on the Cluster Import Gateway Target page:
   a. Type the name of the Remote Cluster Name and the Remote Broker Name.
   b. Select the Local Broker Name.
   c. Click Apply.
7. Click Import.

Exporting And Deploying Assets

Using Deployer, you can deploy your Broker assets and JNDI assets to another webMethods Broker, and share the assets that you created in Broker with other applications.

webMethods Broker enables you to export the following assets to Deployer:
- Broker assets such as clients, client groups, and document types.
- JNDI assets such as JMS queues and JMS topics created by webMethods JNDI providers.
Selecting Broker Assets For Export

To select the Broker assets for export
1 In My webMethods: Administration > Messaging > Broker Servers > Brokers.
2 In the Brokers List, click the Broker from which you want to export the assets. If the Broker does not appear in the list, use the search functionality to locate it.
3 On the Broker Details page, click the Asset Management tab.
4 To select the document types for export, click the Document Types tab and perform the following steps:
   a Select the document types from the list. Use the search functionality to search and select the specific document types.
   b Click Add To Export. The selected document types are added to the list of assets selected for export.
   c Click Add All To Export if you want to select all the document types listed.
5 To select the client groups for export, click the Client Groups tab and perform the above steps:

To select the JMS queues and JMS topics for export
1 In My webMethods: Administration > Messaging > Naming Directories > Providers.
2 Click JNDI Export. The JNDI destinations belonging to the webMethods JNDI providers are displayed in the Providers > JNDI Export page.
3 In the JNDI destinations list, select the JNDI destinations you want to export. If a JNDI destination does not appear in the list, use the search functionality to locate it.
4 Click Add To Export. The selected JNDI destinations are added to Export JNDI Assets list in the right panel.
5 Click Add All To Export if you want to select all the JNDI destinations listed for export.
6 If you do not want to export a JNDI destination listed for export, click Remove to remove that JNDI destination from the list.
7 If you do not want to export any of the JNDI destinations listed for export, click Remove All. All the JNDI destinations selected for export will be cleared from the export summary. You must select the JNDI destinations again if you want to export later.
8 Click Export JNDI Assets.

To build assets
1 Specify the location of the source files of exported assets.
   a Extract the XML files of the exported assets from the .zip files in the My webMethods Server_directory\MWS\broker\acdl\user directory to a separate directory.
   b In the Software AG_directory\common\AssetBuildEnvironment\master_build\build.properties file, specify the location where you have saved the XML source files.
2 Make sure you have specified the properties in the Software AG_directory\common\AssetBuildEnvironment\master_build\build.properties file.
3 Execute the following command from the Software AG_directory\common\AssetBuildEnvironment\master_build directory:
   ant build
This command will store the .acdl file of the assets in the directory specified by the build.output.dir property. The .acdl file contains the exported assets and the dependencies. This metadata schema is defined by Deployer and is common across the webMethods product suite.

Deploying Assets
Broker also supports partial deployment of Broker assets and JNDI assets. When you perform partial deployment in Deployer, only the assets selected from the list of exported assets (available in .acdl files) will be deployed.