



# SAS<sup>®</sup> Viya<sup>™</sup> 3.2

## Administration: SAS<sup>®</sup> Message Broker

---

### SAS Message Broker Overview

SAS uses a set of event APIs that are dependent on Spring Integration and Spring AMQP to interact with the message broker. The AMQP-compliant message broker that SAS uses is Pivotal's RabbitMQ, version 3. RabbitMQ includes the Erlang platform, version 19.

**Note:** A [programming-only deployment](#) does not use SAS Message Broker.

---

### SAS Message Broker: How To

#### Operate

SAS Viya uses the operating system's default init system or systemd command to launch a script that can stop, start, restart, and check the status of SAS Message Broker. This script, `sas-viya-rabbitmq-server-default`, resides in `/etc/init.d`.

**Note:** You must be signed in to the machine where message broker resides with sudo privileges to run this script.

To operate the message broker, run:

```
sas-viya-rabbitmq-server-default status | stop | start | restart
```

**Note:** There is a script with which you can manage and view the running state of all SAS Viya servers and services. For more information, see ["All Servers and Services"](#) in [SAS Viya Administration: General Servers and Services](#).

For your convenience, here are a few examples:

## 2

- checking status of the message broker using a direct call:

```
sudo /etc/init.d/sas-viya-rabbitmq-server-default status
```

- stopping the message broker using the Red Hat Linux version 6 init system command:

```
sudo service sas-viya-rabbitmq-server-default stop
```

- starting the message broker using the Red Hat Linux version 7 systemd command:

```
sudo systemctl start sas-viya-rabbitmq-server-default
```

- restarting the message broker using a direct call:

```
sudo /etc/init.d/sas-viya-rabbitmq-server-default restart
```

## Locate Log Files

SAS Message Broker log files are located in `/opt/sas/viya/config/var/log/rabbitmq-server`.

---

## SAS Message Broker: Concepts

### What Is SAS Message Broker?

SAS Message Broker is an integral piece of the event-driven architecture in which SAS Viya services participate. SAS uses a set of event APIs that are dependent on Spring Integration and Spring AMQP to interact with the message broker. The AMQP-compliant message broker that SAS uses is Pivotal's RabbitMQ. The SAS event APIs provide an abstraction layer between the message broker and its clients, and prevents any breaking changes if SAS decides to use a different third-party message broker other than RabbitMQ in the future.

### How Does Message Broker Work?

SAS Message Broker accepts messages in a standard format and routes them through exchanges and queues, offering configurable degrees of transaction acknowledgment, message persistence, and redundancy. Message broker exchanges accept messages from publishers and route them to zero or more queues. The exchange type controls whether messages are sent to a specific queue, to all associated queues, or only to queues that accept a particular message routing key or match a key pattern.

---

## SAS Message Broker Reference

### Exchanges

- `sas.application.backup`
- `sas.application.topic`
- `sas.event.exchange`
- `sas.search.schema.topic`

## Configuration Files

**Note:** Change these configuration files only when instructed to do so by SAS Technical Support.

- `/opt/sas/viya/config/etc/rabbitmq-server/rabbitmq.config`
- `/opt/sas/viya/config/etc/rabbitmq-server/rabbitmq-env.conf`

## Log Files

SAS Message Broker log files are located in `/opt/sas/viya/config/var/log/consul/default/rabbitmq-server`.

