

RabbitMQ/Erlang OTP Installation

Installation Overview

RabbitMQ is an open-source message broker that provides extensible message queuing support. It is implemented using the Erlang OTP language.

The following steps outline the process of setting up RabbitMQ to use with Aras Innovator:

1. Set up the RabbitMQ server.
2. Install Erlang OTP.
3. Install RabbitMQ.
4. Configure Aras Innovator server nodes.

RabbitMQ should only be used in Aras Innovator clustered environments, as synchronizing cache across nodes is necessary only then.

RemoteMQ Server Setup

A Windows Server 2019 server must be allocated to run the RabbitMQ services. The resources needed for this service will vary from customer to customer depending on the workload required to support the Aras Innovator cluster. The following section provides more guidance for determining the resource needs for a given RabbitMQ server: [RabbitMQ Cluster Sizing and Other Considerations](#) .

Alternatively, a RabbitMQ consultant can be consulted to provide guidance.

A dedicated RabbitMQ server is recommended. RabbitMQ must be installed on a non-HA node like an independent service machine. Do not install it on an Aras Innovator node.

Erlang/OTP Installation

Erlang/OTP is required to run RabbitMQ.

The following steps outline the process of installing the Erlang/OTP:

1. Download Erlang / OTP 26.0.2 Windows 64-bit installer from the following site: { <https://www.erlang.org/downloads> }.



Important

Various versions of Erlang OTP may be available, and the one referenced above may not be the most recent version. However, use the one referenced above to reduce potential issues and restrict the configuration to one explicitly tested and/or used successfully by Aras.

2. Install Erlang/OTP 23 by following the instructions provided on the site.

RabbitMQ Installation and Setup

The following steps outline the process of installing and setting up the RabbitMQ:

1. Install and configure RabbitMQ using the RabbitMQ documentation{
<https://www.rabbitmq.com/docs/install-windows> }.
2. Create an administrative user which will be used later in configuration.
Go to <http://localhost:15672> on the server where RabbitMQ is installed and log in with the Default credentials:
 - **Username:** guest
 - **Password:** guest
3. Create another administrative user account as follows:
 - Go to the Admin tab
 - Add a user
 - Enter any specific name
 - Enter any password & confirm password
 - Set Admin from Tags
 - Click on Add user
 - Click on the Name of the added user
 - Click on Set Permissions
 - Click on the Admin tab & verify that the administrative user successfully added
 - Record username and password for a later step

RabbitMQ should now be installed and verified.

Aras Innovator Configuration

The following steps outline the process of configuring Aras Innovator to use RabbitMQ in place of the default SQL Server invalidation broker:



1. Open the InnovatorServerConfig.xml on the current Aras Innovator server node.
2. Make the following changes:

Turn off the current invalidation broker by commenting out its configuration entry:

- **Original entry:** `<Cache invalidation_broker_type="DatabaseDependency"/>`
- **Deactivated ("commented out") entry:** `<!-- Cache invalidation_broker_type="DatabaseDependency"/-->`

Create a cache invalidation broker entry for the newly configured RabbitMQ server by adding the following entry (at the same XML level as the original entry):

```
<Cache
RabbitServiceHost="%ARAS_RABBIT_HOST%"
RabbitUser="%ARAS_RABBIT_USER%"
RabbitPassword="%ARAS_RABBIT_PASSWORD%"
RabbitExchange="%ARAS_RABBIT_EXCHANGE%"
RabbitHostPort="%ARAS_RABBIT_HOST_PORT%"
invalidation_broker_type="%ARAS_CACHE_DEPENDENCY%" />
```

Substitute in the appropriate values:

- %ARAS_RABBIT_HOST% - Host server that is hosting RabbitMQ
- %ARAS_RABBIT_USER% - Administrative RabbitMQ user (recently added administrative user)
- %ARAS_RABBIT_PASSWORD% - Administrative RabbitMQ password
- %ARAS_RABBIT_EXCHANGE% - Rabbit MQ Exchange on which messages will be sent
- %ARAS_RABBIT_HOST_PORT% - Port for messaging on RabbitMQ (Default is 5672)
- %ARAS_CACHE_DEPENDENCY% - Should be set to 'MessageQueueDependency'

For example:

```
<Cache RabbitServiceHost="192.168.201.71" RabbitUser="admin_44" RabbitPassword="admin_44" RabbitExchange=
```

3. Repeat steps 1 and 2 for all Aras Innovator server nodes.
4. Once all nodes have been updated, restart IIS on all nodes. This will cause each node to be restarted with the updated configuration settings.



5. Validate connections within RabbitMQ. Login to RabbitMQ server from one of the Aras Innovator nodes and go to `http://{ip_of_server_where_RabbitMQ_is_installed}:15672` Login with the following credentials for the new administrative user:

User Name: %ARAS_RABBIT_USER%

Password: %ARAS_RABBIT_PASSWORD%

- new Exchange ('%ARAS_RABBIT_EXCHANGE%') should be created.
- Queues, Channels, and Connections should be created on the RabbitMQ page.

Connecting SSL Support to Rabbit MQ

The following steps outline the process of connecting SSL Support to RabbitMQ:

1. Edit the `<Cache . . . />` string in **InnovatorServerConfig.xml** on Innovator Server. Add the following attributes:

```
RabbitSslEnabled="true" - parameter turns TLS support on or off. It is off by default.  
RabbitSslCertPath="/path/to/client_key.p12" - path to the client's certificate in PKCS#12 format.  
RabbitSslCertPassphrase="yourPassword" - If your certificate has a password, specify it here. If your ce  
RabbitSslServerName="SAN_RABBIT_HOST" - expects this to match the Subject Alternative Name (SAN) or Commo  
Example blow:  
<Cache RabbitServiceHost="10.17.144.7" RabbitUser="innovator1" RabbitPassword="innovator" RabbitExchange=
```

2. Restart RabbitMQ.
3. Restart IIS on Innovator server.

Checking the Connection Between Aras Innovator and RabbitMQ

1. Launch RabbitMQ UI client.
2. Log into the system.
3. Confirm that the new exchange is created with SSL / TLS similar to the image below.



Connections

▼ All connections (2)

Pagination

Page 1 of 1 - Filter: ☐ Regex ?

Overview			Details			Network		+/-
Name	User name	State	SSL / TLS	Protocol	Channels	From client	To client	
10.17.144.6:61741 undefined	innovator1	running	•	AMQP 0-9-1	0	0 B/s	7 B/s	
10.17.144.6:61742 undefined	innovator1	running	•	AMQP 0-9-1	2	0 B/s	0 B/s	

