TIBCO ActiveMatrix BusinessWorks™ Plug-in for AMQP

User Guide

Version 6.5.1 | March 2025

Contents

Overview

Advanced Message Queuing Protocol (AMQP) is an open standard for passing business messages between applications or organizations. This protocol is used to connect systems, provide business processes with the information they require, and reliably transmit instructions to achieve the goals of applications.

By using this plug-in, you can send messages to or receive messages from an AMQP broker queue. In addition, you can also use this plug-in to subscribe to messages from an AMQP broker topic except for the RabbitMQ broker.

You can also publish messages and subscribe to the messages sent to the topic entity for Azure Service Bus broker.

The following figure describes the relationship among an AMQP broker, TIBCO ActiveMatrix BusinessWorks™ Plug-in for AMQP, and TIBCO ActiveMatrix BusinessWorks™.
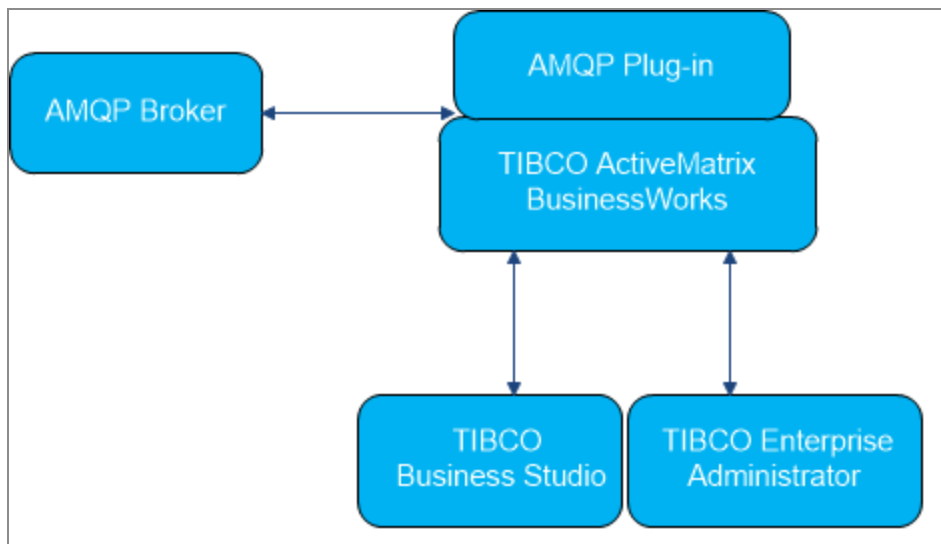
The following list describes each item and the relationship between them in the previous figure.

- The AMQP broker is a server. You can use ActiveMatrix BusinessWorks™ Plug-in for AMQP to communicate with the AMQP broker.

- ActiveMatrix BusinessWorks Plug-in for AMQP plugs into ActiveMatrix BusinessWorks™ and is connected to an AMQP broker.

- ActiveMatrix BusinessWorks is an easy-to-use integration product suite for enterprise applications.

- TIBCO Business Studio™ consists of a graphical user interface (GUI) used by ActiveMatrix BusinessWorks with the plug-ins to design business processes, and the process engine used to execute those processes.

- TIBCO® Enterprise Administrator provides a centralized administrative interface to manage and monitor the plug-in applications deployed in an enterprise.

Getting Started

This tutorial is designed for beginners who want to use ActiveMatrix BusinessWorks Plug-in for AMQP.

A basic procedure of using ActiveMatrix BusinessWorks Plug-in for AMQP includes:

1. Creating a Project
2. Creating an AMQP Connection
3. Configuring a Process
4. Testing a Process
5. Deploying an Application

Overview of TIBCO Business Studio for BusinessWorks

TIBCO Business Studio™ for BusinessWorks™ is an Eclipse-based integration development environment that is used to design, develop, and test ActiveMatrix BusinessWorks applications. The studio provides a workbench in which you can create, manage, and navigate resources in your workspace. A *workspace* is the central location on your computer where all data files are stored.

The following table introduces the workbench UI elements:

UI Element	Description
Menu	Contains menu items such as File, Edit, Navigate, Search, Project, Run, Window, and Help.
Toolbar	Contains the following buttons for frequently used commands:

UI Element	Description
	• **New** • **Save** • **Enable/Disable Business Studio capabilities** • **Create a new BusinessWorks Application Module** • **Debug As** • **Run As**
Perspectives	Contains an initial set and layout of views that are required to perform a certain task. TIBCO Business Studio for BusinessWorks launches the Design perspective by default. Use the Design perspective when designing a process and the Debug perspective when testing and debugging a process. To change the perspective, select **Window > Open Perspective > *perspective_name*** from the main menu. Alternatively, click the **Open Perspective** button from the top upper right of the workbench and select the perspective.
Views	Lists the resources and helps you navigate within the workbench. For example, the Project Explorer view displays the ActiveMatrix BusinessWorks applications, modules, and other resources in your workspace, and the Properties view displays the properties for the selected resource. To open a view, select **Window > Show View > *view_nameview_name*** from the main menu.
Editors	Provides a canvas to configure, edit, or browse a resource. Double-click a resource in a view to open the appropriate editor for the selected resource. For example, double-click on a process (`MortgageAppConsumer.bwp`) in the Project Explorer view to open the process in the editor.
Palette	Contains a set of widgets and a palette library. A *palette* groups activities that perform similar tasks, and provides quick access to activities when configuring a process.

Creating a Project

The first task for using the plug-in is creating a project. After creating a project, you can add resources and processes.

An Eclipse project is an application module configured for TIBCO ActiveMatrix BusinessWorks. An application module is the smallest unit of resources that is named, versioned, and packaged as part of an application.

Procedure

1. Start TIBCO Business Studio for BusinessWorks by using one of the following ways:

   - Microsoft Windows: click **Start > All Programs > TIBCO > *TIBCO_HOME* > TIBCO Business Studio *version_number* > Studio for Designers**.

   - macOS and Linux: run the TIBCO Business Studio for BusinessWorks executable file located in the *TIBCO_HOME*/studio/*version_number*/eclipse directory.

2. From the menu, click **File > New > BusinessWorks Resources** to open the BusinessWorks Resource wizard.

3. In the Select a wizard dialog, click **BusinessWorks Application Module** and click **Next** to open the New BusinessWorks Application Module wizard.

4. In the Project dialog, configure the project that you want to create:

   a. In the **Project name** field, enter a project name.

   b. By default, the created project is located in the workspace currently in use. If you do not want to use the default location for the project, clear the **Use default location** checkbox and click **Browse** to select a new location.

   c. Use the default version of the application module, or enter a new version in the **Version** field.

   d. Keep the **Create empty process** and **Create Application** checkboxes selected to automatically create an empty process and an application when creating the project.

   e. Select the **Use Java configuration** checkbox if you want to create a Java module.

      A Java module provides the Java tooling capabilities.

   f. Click **Finish** to create the project.

Result

The project with the specified settings is displayed in the Project Explorer view.
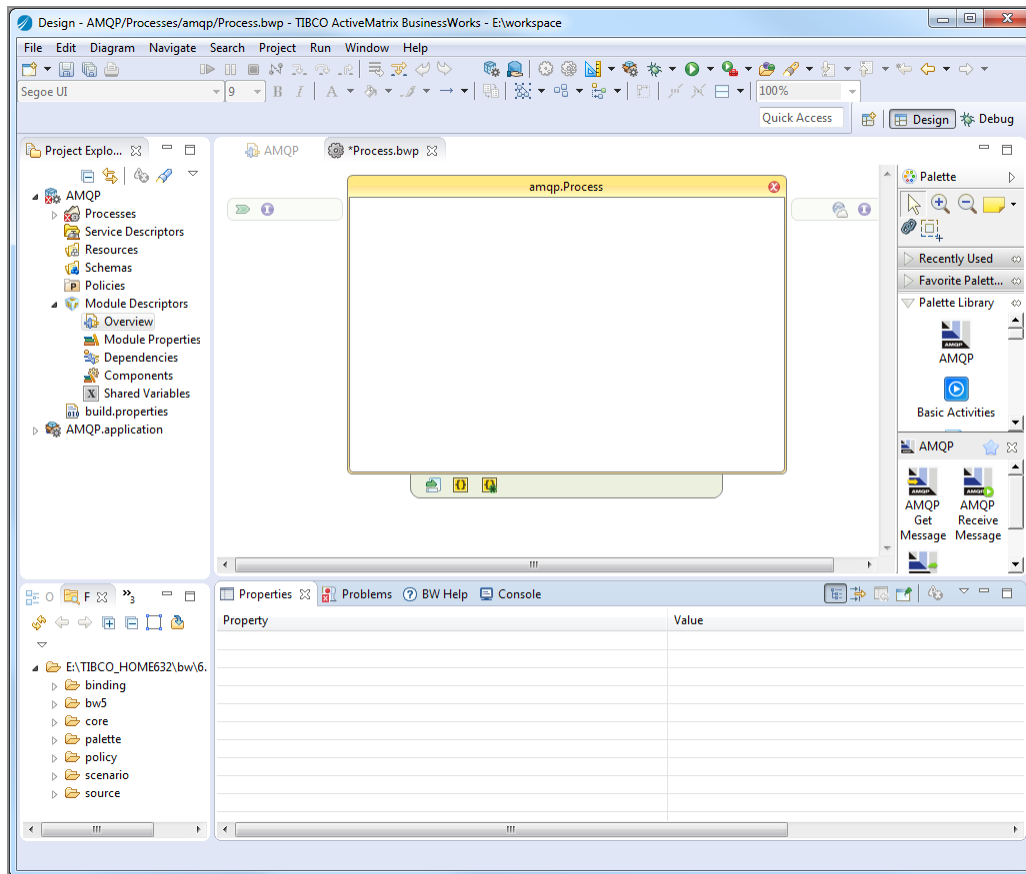
Creating an AMQP Connection

After creating a project, you have to create an AMQP Connection shared resource to create a connection between the plug-in and the AMQP server. With this shared resource, you can share the AMQP connection at runtime between different AMQP activities within a process.

Before you begin

The AMQP Connection shared resource is available at the **Resources** level. Ensure that you have created a project, as described in Creating a Project.

Procedure

1. Expand the created project in the Project Explorer view.

2. Right-click the **Resources** folder and click **New > AMQP Connection** to open the AMQP Connection wizard.

3. The resource folder, package name, and resource name of the AMQP connection are provided by default. If you do not want to use the default configurations, change them accordingly. Click **Finish** to open the AMQP Connection editor.

4. Configure the AMQP Connection shared resource in the AMQP Connection editor.

   For information about the configuration fields, see AMQP Connection.

5. Click **Test Connection** to validate the connection.

Configuring a Process

After creating a project, an empty process is created. You can add activities to the empty process to complete a task, such as sending a message.

Before you begin

Ensure that you have created an empty process when creating a project. For more details, see Creating a Project.

Procedure

1. In the **Project Explorer** view, click the created project and open the empty process from the **Processes** folder.

2. Select activities from the **Palette** view and drop them in the **Process** editor.

   For example, select and drop the Timer activity from the General Activities palette, and the AMQPSendMessage activity from the AMQP palette.
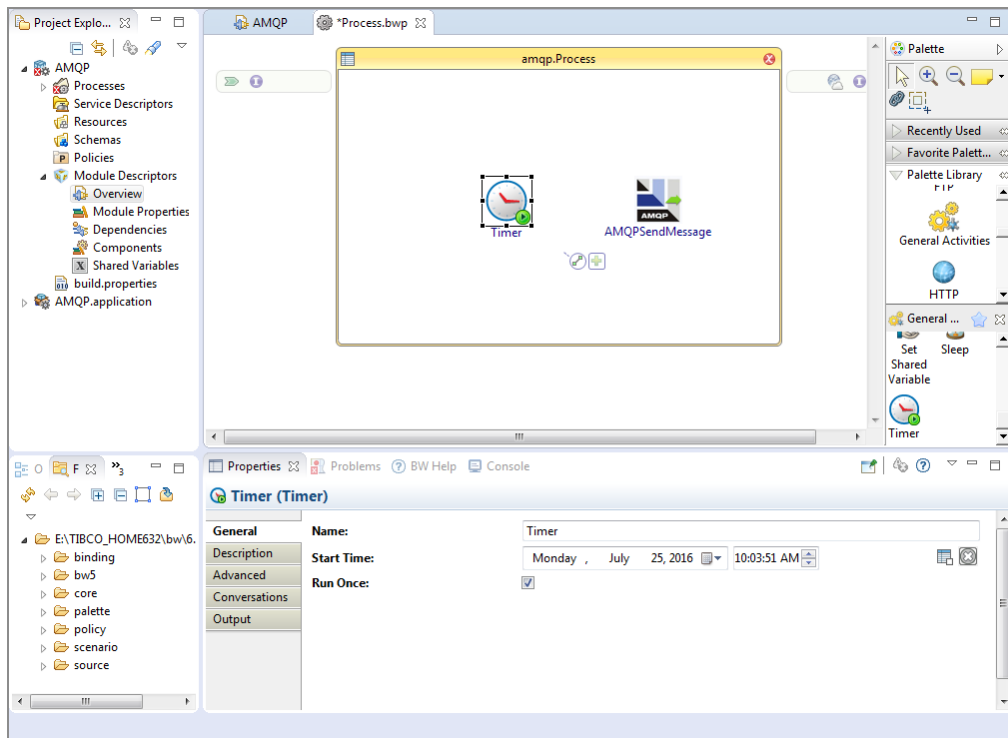
3. Drag the ⊘ icon to create a transition between the added activities.

4. Configure the added AMQP activity, as described in AMQP Palette.

> ℹ **Note:** An AMQP connection is required when configuring the AMQP activities. For more details about how to create an AMQP connection, see Creating an AMQP Connection.

5. Click **File > Save** to save the process.

Testing a Process

After configuring a process, you can test the process to check whether the process completes your task.

Before you begin

Ensure that you have configured a process, as described in Configuring a Process.

Procedure

1. On the toolbar, click **Debug > Debug Configurations**.

2. Click **BusinessWorks Application > BWApplication** in the left pane.

   By default, all the applications in the current workspace are selected on the **Applications** tab. Ensure that only the application you want to debug is selected on the **Applications** tab in the right pane.

3. Click **Debug** to test the process in the selected application.

   TIBCO Business Studio for BusinessWorks changes to the Debug perspective. The debug information is displayed in the Console view.

4. On the **Debug** tab, expand the running process and click an activity.

5. In the upper-right corner, click the **Job Data** tab, and then click the **Output** tab to check the activity output.
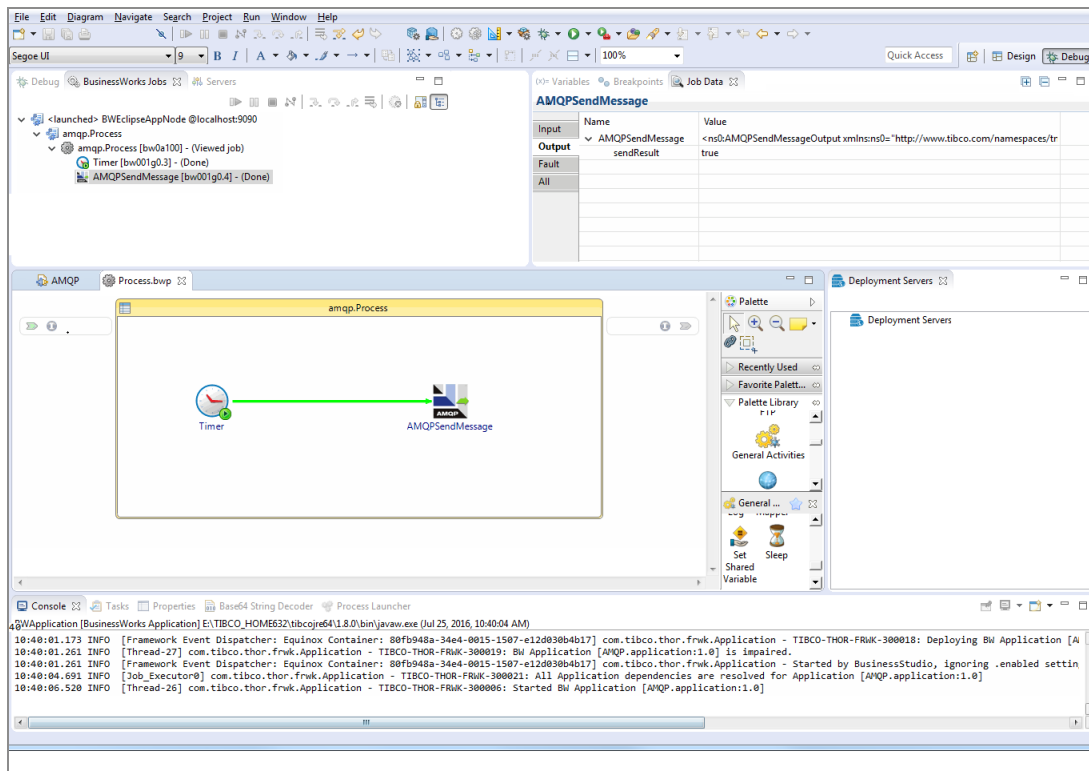
Deploying an Application

After testing, if the configured process works as expected, you can deploy the application that contains the configured process into a runtime environment, and then use the bwadmin utility to manage the deployed application.

Deploying an application involves the following tasks:

1. Uploading an application archive

2. Deploying an application archive

3. Starting an application

For more details about deploying an application, see the *TIBCO ActiveMatrix BusinessWorks™ Administration* guide.

AMQP Connection

The AMQP Connection shared resource contains all the necessary parameters that have to be configured when you use the TIBCO ActiveMatrix BusinessWorks™ Plug-in for AMQP to connect to the AMQP server. The AMQP Connection shared resource is used by all the activities in the AMQP palette.

General

The following table lists the fields in the General panel:

Field	Module Property?	Description
Package	No	The name of the package where the shared resource is added.
Name	No	The name to be displayed as the label for the shared resource in the process.
Description	No	A short description for the shared resource.

AMQP Connection Resource Configuration

The following table lists the fields in the AMQP Connection Resource Configuration panel:

Field	Module Property?	Description
BrokerType	No	Specifies the broker that is used for the connection. The following brokers are available: • **Qpid-1-0** • **RabbitMQ-0-9**

Field	Module Property?	Description
		• **ActiveMQ-1-0** • **AzureSB-1-0** • **AMQ-1-0** • **ActiveMQ-Artemis-1-0** The default broker is **Qpid-1-0**. For information of the latest versions of the available brokers, see the [Readme]().
HostPort	Yes	Specifies the host name or IP address. **Note:** This field is available only when **Qpid-1-0**, **RabbitMQ-0-9**, **AMQ-1-0**, **ActiveMQ-1-0**, or **ActiveMQ-Artemis-1-0** is selected from the **BrokerType** list.
VirtualHost	Yes	Specifies the namespace for entities, such as exchanges and queues. **Note:** This field is available only when **Qpid-1-0** or **RabbitMQ-0-9** is selected from the **BrokerType** list.
Username	Yes	Specifies the username that you use to get access to the AMQP server. **Note:** This field is available only when **Qpid-1-0**, **RabbitMQ-0-9**, **AMQ-1-0**, **ActiveMQ-1-0**, or **ActiveMQ-Artemis-1-0** is selected from the **BrokerType** list.
Password	Yes	Specifies the password that you use to get access to the AMQP server.

Field	Module Property?	Description
		Note: This field is available only when **Qpid-1-0**, **RabbitMQ-0-9**, **AMQ-1-0**, **ActiveMQ-1-0**, or **ActiveMQ-Artemis-1-0** is selected from the **BrokerType** list.
Client ID	Yes	Set an explicit **Client ID** in the AMQP connection resource as a part of the connection details for the following brokers: - Qpid-1-0 - AMQ-1-0 - ActiveMQ-1-0 - ActiveMQ-Artemis-1-0
Authentication Type	Yes	The following authentication types are supported by the plug-in when **AzureSB-1-0** is selected from the **BrokerType** list: - SAS - OAuth - Managed Identity

Field	Module Property?	Description
		Important: • Register your application with Microsoft Azure Active Directory before you can use OAuth authentication to connect to the AMQP server. For more information, see [Registering an Application in Azure Active Directory](#). • Create and configure a User-Assigned Managed Identity in Microsoft Azure before you can use it for Managed Identity authentication to connect to Azure Service Bus. For more information, see [Assigning a User-Assigned Managed Identity to an Azure Virtual Machine](#). **Note:** • The plug-in supports the OAuth 2.0 authorization protocol for all the palette activities that are supported by AMQP. • The plug-in supports only User-Assigned Managed Identities and does not support System-Assigned Managed Identities.
Connection String	Yes	Specifies the value used to connect to the Azure Service Bus broker instance. The syntax is `Endpoint=sb://<Namespace>.servicebus.windows.net/` **Note:** This field is available only when **AzureSB-1-0** is selected from the **BrokerType** list.
Tenant ID	Yes	The unique ID of your Microsoft Azure Active Directory with which you have registered your

Field	Module Property?	Description
		application. This field is available only when **AzureSB-1-0** is selected from the **BrokerType** list and the **OAuth** authentication is selected.
		Tip: To know the Tenant ID, in Microsoft Azure, click App registrations and click Endpoints. In the fields that are displayed, the string value after the host name in the OAuth 2.0 TOKEN ENDPOINT field is known as Tenant ID. You can copy the OAuth 2.0 Token Endpoint field value by clicking the icon beside it.
Client ID	Yes	The unique ID of the application that you want to connect to the AMQP server. This is application-specific and is unique for each application. This field is available only when **AzureSB-1-0** is selected from the **BrokerType** list and the **OAuth** authentication is selected.
		Tip: To know the Client ID in Microsoft Azure, click **App registrations**, and in the list of registered applications, select a specific registered application. In the application pane, the value assigned under **Application ID** is known as Client ID.
Client Secret	Yes	The password that you want to use to connect to the AMQP server. This field is available only when **AzureSB-1-0** is selected from the **BrokerType** list and the **OAuth** authentication is selected. To generate a new password in Microsoft Azure, see Microsoft Azure documentation.
		Note: An application can have one or more passwords that can be used as a secret key while obtaining OAuth2.0 keys.

Field	Module Property?	Description
Shared Access Key Name	Yes	Specifies the name of the shared access policy created for an entity. **Note:** This field is available only when **AzureSB-1-0** is selected from the **BrokerType** list and the **SAS** authentication is selected.
Shared Access Key	Yes	Specifies the key for **Shared Access Key Name**. **Note:** This field is available only when **AzureSB-1-0** is selected from the **BrokerType** list and the **SAS** authentication is selected.
Entity Type	No	Specifies the type of the entity created. You can select an entity type from the following: • Queue • Topic **Note:** This field is available only when **AzureSB-1-0** is selected from the **BrokerType** list. The value specified for the **Entity Type** in the activity takes precedence over the value specified for this field.
Entity Name	Yes	Specifies the name of the queue or topic entity that is existing on the server. **Note:** This field is available only when **AzureSB-1-0** is selected from the **BrokerType** list.
Entity Subscriber Name	Yes	Specifies the name of the subscriber that is existing on the server for the topic specified in **Entity Name**.

Field	Module Property?	Description
		Note: This field is available only when **AzureSB-1-0** is selected from the **BrokerType** list and Topic is selected from the **Entity Type** list. This field is needed only when testing connectivity to the Azure Service Bus broker by using a "Listen only" **Shared Access Key**.
ConnectionTimeout (msec)	Yes	Specifies the time period (in milliseconds) the connection remains active if no action is performed. **Note:** For failover, increase the connection timeout to allow transition from one server to another server.
Session Count	Yes	Specifies the number of channels that are created on the server. The default value is 1. The minimum value is 1 and maximum value is 20. **Note:** Session Count is available when **RabbitMQ-0-9** is selected from the **Broker Type** list. Session count works with `Prefetch count` property. For more information about this property, see Setting Prefetch Count for RabbitMQ Broker.
Connection Recovery	No	If this checkbox is selected, the connection between the plug-in and the server is automatically made if the connection is interrupted.

Field	Module Property?	Description
		Note: For the RabbitMQ server, if you have given the details of a single **HostPort**, you need not select the **Connection Recovery** checkbox to achieve recovery of the connection.
		In the sender activity, when the ActiveMQ server is set up in leader-follower mode, when the master server is down, for master-subordinate transition, **ConnectionTimeout(msec)** must be configured to a value more than 30 seconds.
Retry Interval (msec)	Yes	Specifies the time period (in milliseconds) between successive attempts made to establish an interrupted connection again.
		The default value is `3000`.
		Note: This field is available when **ActiveMQ-1-0**, **Qpid-1-0**, **AMQ-1-0**, **AzureSB-1-0**, or **ActiveMQ-Artemis-1-0** is selected from the **BrokerType** list and the **Connection Recovery** checkbox is selected.
Retry Attempts	Yes	Specifies the number of attempts allowed to establish a connection after the connection was interrupted.
		The default value is `20`.
		Note: This field is available only when **ActiveMQ-1-0**, **Qpid-1-0**, **AMQ-1-0**, **AzureSB-1-0**, or **ActiveMQ-Artemis-1-0** is selected from the **BrokerType** list and the **Connection Recovery** checkbox is selected.
NetworkRecoveryInterval(msec)	Yes	Specifies the time period (in milliseconds) between successive attempts made to establish an

Field	Module Property?	Description
		interrupted connection again.
		The default value is `5000`.
		Note: This field is available only when **RabbitMQ-0-9** is selected from the **BrokerType** list and the **Connection Recovery** checkbox is selected.
Test Connection	No	Click this button to validate the connection.

SSL

SSL configuration is applicable for RabbitMQ-0-9, ActiveMQ-1-0, AMQ-1-0, ActiveMQ-Artemis-1-0, and Qpid-1-0 brokers. The following fields are available in the SSL panel:

Field	Module Property?	Description
Confidentiality	No	If you want to choose an existing SSL Client shared resource or create one, select this checkbox. This shared resource contains specified SSL information for connection to an AMQP broker. For more details, see the TIBCO ActiveMatrix BusinessWorks™ documentation.
SSL Client	No	Select an SSL Client shared resource to establish a connection between the plug-in and the specified AMQP broker.

Connection Recovery and Fault Tolerance

Fault tolerance is the ability of the system to continue processing requests even if an unexpected failure occurs in the system. Select the **Connection Recovery** checkbox to

achieve fault tolerance. Give comma-separated values in the **HostPort** field.

RabbitMQ Broker

For RabbitMQ broker, you must configure the **NetworkRecoveryInterval(msec)** field to achieve connection recovery. If the connection recovery is enabled, the connection recovery waits for the time mentioned in the **NetworkRecoveryInterval(msec)** field and then retry.

Apache ActiveMQ, Apache Qpid, Red Hat AMQ, ActiveMQ Artemis, and Microsoft Azure Service Bus Brokers

For Apache ActiveMQ (ActiveMQ), Apache Qpid (QPID), Microsoft Azure Service Bus (AzureSB), Red Hat AMQ (AMQ), and ActiveMQ Artemis (Artemis) brokers, you must configure **Retry Interval (msec)** and **Retry Attempts** to achieve connection recovery.

The AMQPSendMessage activity timeout is governed by the TIBCO ActiveMatrix BusinessWorks timeout, whose default value is 3 minutes for all the brokers supported. If a reconnection attempt is made within this time limit, the connection is reestablished, otherwise the TIBCO ActiveMatrix BusinessWorks displays an activity timeout error. The default behavior can be overridden by setting one of the following:

File Name	Property Name	Location
bwappnode-<AppNodeName>.tra	`java.property.bw.engine.activity.async.waitTime=n(millisec)`	<TIBCO_HOME>/bw/6.x/domains/<Domain_Name>/appnodes/<AppSpaceName>/<AppNodeName>/bin/
config.ini	`bw.engine.activity.async.waitTime=n(millisec)`	<TIBCO_HOME>/bw/6.x/domains/<Domain_Name>/appnodes/<AppSpaceName>/<AppNodeName>

Setting Retry Properties for Microsoft Azure Service Bus broker

The plug-in provides an option to set `maxReconnectAttempts` and `maxReconnectDelay` properties as system properties when attempting to connect to the Microsoft Azure Service Bus broker.

Setting Retry Properties for Microsoft Azure Service Bus Broker in TIBCO Business Studio™ at Run Time

1. From the TIBCO Business Studio™ menu, click **Run** > **Run Configurations**.

2. In the Run Configurations dialog, perform the following tasks:

   a. In the left pane, click **BusinessWorks Application** > **BWApplication**.

   b. In the right pane, click the **Arguments** tab.

   c. In the **VM arguments** field, add the following parameters:

      - `Dcom.tibco.plugin.amqp.maxReconnectAttempts=<maxReconnectAttempts>`
      - `Dcom.tibco.plugin.amqp.maxReconnectDelay=<maxReconnectDelay>`

Setting Retry Properties for Microsoft Azure Service Bus Broker in TIBCO® Enterprise Administrator at Run Time

1. Navigate to the `TIBCO_HOME/bw/version_number/bin` directory and open the `bwcommon.tra` file.

2. Add the following parameters to the `java.extended.properties` property:

   - `com.tibco.plugin.amqp.maxReconnectAttempts=<maxReconnectAttempts>`
   - `com.tibco.plugin.amqp.maxReconnectDelay=<maxReconnectDelay>`

> **ⓘ** **Note:** Setting these system properties values takes precedence over the values set on AMQP Shared Resource Connection for **Retry Interval (msec)** and **Retry Attempts** fields.

Registering an Application in Azure Active Directory

To connect to Microsoft Azure from an AMQP Connection shared resource, you must do the following:

- Register an AMQP application in Microsoft Azure Active Directory

- Provide the necessary permissions to this application in the Azure Active Directory's App Registration page

For more information, see the Microsoft Azure documentation.

Assigning a User-Assigned Managed Identity to an Azure Virtual Machine

To assign a User-Assigned Managed Identity (UMI) to an Azure Virtual Machine, you must do the following:

- Create a User-Assigned Managed Identity (UMI).

- Assign the UMI to the Azure Virtual Machine.

- Assign Roles and Permissions on the Service Bus.

For more information, see Microsoft Azure documentation.

AMQP Palette

A palette groups the activities that connect to the same external applications together. The AMQP palette is added to TIBCO Business Studio™ for BusinessWorks™ after installing the TIBCO ActiveMatrix BusinessWorks™ Plug-in for AMQP.

The AMQP palette contains the following activities:

- AMQPGetMessage
- AMQPReceiveMessage
- AMQPSendMessage

AMQPGetMessage

You can use the AMQPGetMessage activity to retrieve messages from a specified queue.

General

The following table describes the configurations on the **General** tab of the AMQPGetMessage activity.

Field	Module Property?	Description
Name	No	Specifies a name of the activity displayed in the process. The default value is `AMQPGetMessage`.
Connection	Yes	Click 🔍 to select an AMQP Connection shared resource. The AMQP Connection shared resource creates a connection between the plug-in and an AMQP server. If no matching AMQP Connection shared resources are found, click **Create Shared Resource** to create one. For more details, see Creating an AMQP Connection.

Field	Module Property?	Description
Queue Name	Yes	Enter the name of the queue from where the message is received. **Note:** This field is available when **RabbitMQ-1-0** is selected from the **Broker Type** list of the used AMQP Connection shared resource.
Entity Type	No	Select an entity type from the following: • Queue • Topic **Note:** This field is not available when **RabbitMQ-1-0** is selected from the **Broker Type** list of the used AMQP Connection shared resource.
Entity Name	Yes	Enter the name of the queue or topic entity from where the message is received. **Note:** This field is not available when **RabbitMQ-1-0** is selected from the **Broker Type** list of the used AMQP Connection shared resource.
Subscription Name	Yes	Enter the name of the subscription from where the message is received. **Note:** • This field is not available when **RabbitMQ-1-0** is selected from the **Broker Type** list of the used AMQP Connection shared resource. • This field is available only when **Topic** is selected as the **Entity type**.
Durable	No	Select this checkbox to create a durable subscription.

Field	Module Property?	Description
Subscription		**Note:** • This field is not available when **RabbitMQ-1-0** or **AzureSB-1-0** is selected from the **Broker Type** list of the used AMQP Connection shared resource. • This field is available only when **Topic** is selected as the **Entity type**.
Shared Subscription	No	Select this checkbox to create a shared subscription. **Note:** • A shared subscription is not supported with **ActiveMQ-1-0**, **RabbitMQ-1-0** or **AzureSB-1-0** from the **Broker Type** dropdown list of the used AMQP Connection resource. • This field is available only when **Topic** is selected as the **Entity type**.
Message Type	No	Select one of the following message types: • **TextMessage**: the message is in the String format. • **BytesMessage**: the message is in the Byte stream format. • **Simple**: the message does not have a message body.
Acknowledge Mode	No	Select one of the following acknowledge modes: • **Auto**: the message is automatically acknowledged when being received. • **Client**: the message is acknowledged after being received by the Confirm activity.

Field	Module Property?	Description
		Note: Ensure that the message can be confirmed when you use the Client mode. If the message is not confirmed before the process instance ends, the message is kept as Unacked status and is still in the queue.
		The default value is Auto.

Description

On the **Description** tab, you can add a short description for the AMQPGetMessage activity.

Input

The **Input** tab displays the input schema of the activity as a tree structure. The input values vary depending on the value selected for **Entity Type** on the **General** tab. The value specified for the **queueName** or **topicName** field on the **Input** tab takes precedence over the value specified for the **Entity Name** field on the **General** tab. The following table describes the input elements on the **Input** tab of the AMQPGetMessage activity.

Input Item	Data Type	Description
queueName	String	Enter the name of the queue from which you want to get the message. **Note:** The **queueName** field is only displayed when the **Entity Type** is selected as **Queue** on the **General** tab.
topicName	String	Enter the name of the topic from where the message is received.

Input Item	Data Type	Description
		Note: This field is not available when **RabbitMQ-1-0** is selected from the **Broker Type** list of the used AMQP Connection shared resource and the **Entity Type** field on the **General** tab is Topic.
subscriptionName	String	Enter the name of the subscription from where the message is received. **Note:** This field is not available when **RabbitMQ-1-0** is selected from the **Broker Type** list of the used AMQP Connection shared resource and the **Entity Type** field on the **General** tab is Topic.

Note: The **Input** tab of the AMQPGetMessage activity is not available on projects created before version 6.0.2 of ActiveMatrix BusinessWorks Plug-in for AMQP.

Output

The following table describes the output elements on the **Output** tab of the AMQPGetMessage activity.

Output Item	Data Type	Description
UserProperties	Complex	The message header consists of the following user properties: • name (a String value): the name of the user property. • type: the property data type. The valid data types include String, Number, and Boolean. • value (a String value): the value of the user property.

Output Item	Data Type	Description
		Note: If the property data type is Boolean, any other value except for `true` or `True` or `TRUE` is considered false.
`MessageProperties`	Complex	The message properties are listed as follows:

- `deliveryMode` (a Boolean value): the delivery mode of the messages. If the value is set to true, the messages are stored before forwarding. If the value is set to false, the messages are not stored before forwarding. In this case, if the transmission failed, the messages might be lost.

- `messageID` (a String value): the message ID of the received message.

- `timestamp` (an Integer value): the time when the message is sent to the queue.

- `expiration` (an Integer value): the time of the message remains active (in milliseconds).

  Note: For `Qpid-1-0`, `AMQ-1-0`, `AzureSB-1-0`, `ActiveMQ-1-0`, and `ActiveMQ-Artemis-1-0` displays the point in time of the expiration. For `RabbitMQ`, displays the active time length.

- `priority` (an Integer value): the delivery priority of the message.

- `type` (a String value): the event type or command type the message represents.

- `contentType` (a String value): value indicating the message content type.

- `correlationID` (a String value): value specifying the context for the message for correlation.

Output Item	Data Type	Description
		Note: This field is available only when **AzureSB-1-0** is selected from the **Broker Type** list of the used AMQP Connection shared resource.
		• `entityPath`: relative path of the Queue or Topic entity depending on whether the activity is getting the message from an Azure Service Bus Queue or Topic.
		Note: Displayed only when AzureSB-1-0 is chosen as the broker type in the selected shared resource for this activity. If Queue is chosen as the **Entity Type**, the value would be the Queue name. If Topic is chosen as the **Entity Type**, the value would be of the form `<topicname>/subscriptions/<subscriptionname>`
body	String/Byte	The message body of the received message: • If **TextMessage** is selected from the **Message Type** list of the **General** tab, the message is in the String format. • If **BytesMessage** is selected from the **Message Type** list of the **General** tab, the message is in the String format. **Note:** If **Simple** is selected from the **Message Type** list of the **General** tab, the `body` field is unavailable.

Fault

The following table describes the error schema elements on the **Fault** tab of the AMQPGetMessage activity.

Error Schema Element	Description
AMQPPluginException	This error message is displayed when the plug-in captures any exception related to the process of sending a message to an AMQP broker.
AMQPConnectionException	This error message is displayed when any exception about the connection problem occurs in the process of sending a message to an AMQP broker.

> **Note:** Connection Recovery is not applicable with AMQPGetMessage activity for the RabbitMQ broker.

AMQPReceiveMessage

You can start a process with the AMQPReceiveMessage activity. The process is triggered when the plug-in receives a message from a specified AMQP broker queue or topic.

General

The following table lists the configurations on the **General** tab of the AMQPReceiveMessage activity.

Field	Module Property?	Description
Name	No	The name of the activity displayed in the process. The default value is `AMQPReceiveMessage`.
Connection	Yes	Click 🔍 to select an AMQP Connection shared resource.
		The AMQP Connection shared resource creates a connection between the plug-in and an AMQP server.
		If no matching AMQP Connection shared resources are found, click **Create Shared Resource** to create one. For

Field	Module Property?	Description
		more details, see Creating an AMQP Connection.
Queue Name	Yes	Enter the name of the queue from where the message is received. **Note:** This field is available when **RabbitMQ-1-0** is selected from the Broker Type list of the used AMQP Connection shared resource.
Entity Type	No	Select an entity type from the following: • Queue • Topic **Note:** This field is not available when **RabbitMQ-1-0** is selected from the **Broker Type** list of the used AMQP Connection shared resource.
Entity Name	Yes	Enter the name of the queue or topic entity from where the message is received. **Note:** This field is not available when **RabbitMQ-1-0** is selected from the **Broker Type** list of the used AMQP Connection shared resource.
Subscription Name	Yes	Enter the name of the subscription from where the message is received. **Note:** • This field is not available when **RabbitMQ-1-0** is selected from the **Broker Type** list of the used AMQP Connection shared resource. • This field is available only when **Topic** is selected as the **Entity type**.

Field	Module Property?	Description
Durable Subscription	No	Select this checkbox to create a durable subscription. **Note:** • This field is not available when **RabbitMQ-1-0** or **AzureSB-1-0** is selected from the **Broker Type** list of the used AMQP Connection shared resource. • This field is available only when **Topic** is selected as the **Entity type**.
Shared Subscription	No	Select this checkbox to create a shared subscription. **Note:** • A shared subscription is not supported with **ActiveMQ-1-0**, **RabbitMQ-1-0** or **AzureSB-1-0** from the **Broker Type** dropdown list of the used AMQP Connection resource. • This field is available only when **Topic** is selected as the **Entity type**.
Message Type	No	Select one of the following message types: • **TextMessage**: the message is in the string format. • **BytesMessage**: the message is in the byte stream format. • **Simple**: the message does not have a message body. • **Any**: Both text and byte type messages are allowed. **Note:** **Any** field is available only when **Qpid-1-0 broker** is selected from the Broker Type list of the used AMQP Connection shared resource.
MaxMessages	Yes	The maximum number of messages that can be fetched by

Field	Module Property?	Description
		the AMQP Receive Message activity for each request submitted to the Azure Service Bus server. The default value is 1. **Note:** • This field is available only when **AzureSB-1-0** is selected from the **Broker Type** list of the used AMQP Connection shared resource. • The **MaxMessages** field should be equal to 1 or equal to the FlowLimit property value.
Acknowledge Mode	No	Select one of the following acknowledge modes: • **Auto**: the message is automatically acknowledged when being received. • **Client**: the message is acknowledged after being received by the Confirm activity. **Note:** Ensure that the message can be confirmed when you use the **Client** mode. If the message is not confirmed before the process instance ends, the message is kept as `Unacked` status and is still in the queue. The default value is **Auto**.
Receiver Mode	No	Select one of the following modes: • **PeekLock**: the messages are locked by Azure Service Bus for processing by the AMQP Receive Message activity for a configurable period. • **ReceiveAndDelete**: the messages are deleted from Azure Service Bus immediately after being sent for processing by the AMQP Receive Message activity. The default value is **PeekLock**.

Field	Module Property?	Description
		Note: • The **Acknowledge Mode** option will not be available if you select **ReceiveAndDelete**. • The **Receiver Mode** option is available only when **AzureSB-1-0** is selected as the **Broker Type** in the AMQP Connection shared resource. For more information about Message receive modes, see the Microsoft Azure documentation.
Get CorrelationID	No	Select this checkbox to retrieve the CorrelationID of the message in the output. By default, this checkbox is not selected. **Note:** The **Get CorrelationID** option is available only when **AMQ-1-0** is selected as the **Broker Type** in the AMQP Connection shared resource.

Description

On the **Description** tab, you can add a short description for the AMQPReceiveMessage activity.

Advanced

The following table describes the configurations on the **Advanced** tab of the AMQPReceiveMessage activity.

Field	Module Property?	Description
Session Enabled	No	The **Session Enabled** checkbox is displayed only when AzureSB-1-0 is chosen as the **Broker Type** in the selected shared resource for this receiver. Select the **Session Enabled**

Field	Module Property?	Description
		checkbox to receive messages from a session enabled queue or topic.
Receive Type	No	The **Receive Type** field is available only when the **Session Enabled** checkbox is selected. The messages can be received from the queue or topic in one of the following methods: • First Available Session • Specific Session • All Available Sessions
Max Concurrent Sessions	Yes	The **Max Concurrent Sessions** field is available only when **All Available Sessions** is selected in the **Receive Type** field. The **Max Concurrent Sessions** field has the following properties: • The **Max Concurrent Sessions** field can be set to any number greater than 0. The **Max Concurrent Sessions** field value specifies the number of sessions from which the AMQP ReceiveMessage activity can receive messages. • The AMQP ReceiveMessage activity receives messages from the first available session. If there are no new messages in the session for a specific duration, the receiver lock on the session expires. The receiver acquires a lock on the next available session and receives messages from that session. The lock expire duration can be set in the shared resource under the **ConnectionTimeout(msec)** field. • It is not advisable to set the Flow limit when the **Max Concurrent Sessions** field is set to more than 1. It may lead to the session lock expired error.
Session Id	Yes	The **Session Id** field is available only when Single Session is

Field	Module Property?	Description
		selected in the **Receive Type** field. Provide the session Id of the session from where you want to receive messages.
Sequence Key	No	This field contains an Xpath expression that specifies the order in which the process runs. Process instances with sequencing keys that have the same value are run sequentially in the order in which the process instances are created.
		For more details, see the TIBCO ActiveMatrix BusinessWorks™ documentation.
Custom Job Id	No	This field contains an Xpath expression that specifies a custom job ID for every job in the process instance.
		For more details, see the TIBCO ActiveMatrix BusinessWorks documentation.

Conversations

On the **Conversations** tab, you can click the **Add New Conversation** 🔳 icon to initiate a conversation. A conversation represents two or more related message exchanges in the same process that are correlated by the BusinessWorks engine.

Output

The following table describes the output elements on the **Output** tab of the AMQPReceiveMessage activity.

Output Item	Data Type	Description
UserProperties	Complex	The message header consists of the following user properties: • name (a String value): the name of the user property.

Output Item	Data Type	Description
		• `type`: the property data type. The valid data types include String, Number, Boolean.
		• `value` (a String value): the value of the user property.
		Note: If the property data type is Boolean, any other value except for `true` or `True` or `TRUE` is considered false.
`MessageProperties`	Complex	The message properties are listed as follows:
		• `deliveryMode` (a Boolean value): the delivery mode of the messages. If the value is set to `true`, the messages are stored before forwarding. If the value is set to `false`, the messages are not stored before forwarding. In this case, if the transmission failed, the messages might be lost.
		• `messageID` (a String value): the message ID of the received message.
		• `timestamp` (an Integer value): the time when the message is sent to the queue.
		• `expiration` (an Integer value): the time of the message remains active (in milliseconds).
		Note: For `Qpid-1-0`, `AMQ-1-0`, `AzureSB-1-0`, `ActiveMQ-1-0`, and `ActiveMQ-Artemis-1-0` displays the point in time of the expiration. For `RabbitMQ`, displays the active time length.
		• `priority` (an Integer value): the delivery priority of the message.
		• `type` (a String value): the event type or command type the message represents.
		• `contentType` (a String value): value indicating the message content type.

Output Item	Data Type	Description
		• `correlationID` (a String value): value enabling an application to specify the context for the message for correlation. **Note:** Displayed only when **AzureSB-1-0** or **AMQ-1-0** is chosen as the **Broker Type** in the selected shared resource for this receiver. • `sessionID`: session ID of the Queue or Topic entity depending on whether the receiver is listening for messages from an Azure Service Bus Queue or Topic. **Note:** Displayed only when **AzureSB-1-0** is chosen as the **Broker Type** in the selected shared resource for this receiver. • `entityPath`: the relative path of the Queue or Topic entity depending on whether the receiver is listening for messages from an Azure Service Bus Queue or Topic. **Note:** Displayed only when **AzureSB-1-0** is chosen as the **Broker Type** in the selected shared resource for this receiver. If Queue is chosen as the **Entity Type**, the value would be the Queue name. If Topic is chosen as the **Entity Type**, the value would be of the form `<topicname>/subscriptions/<subscriptionname>`
body	String/Byte	The message body of the received message: • If **TextMessage** is selected from the **Message Type** list of the **General** tab, the message is in the String format. • If **BytesMessage** is selected from the **Message**

Output Item	Data Type	Description
		Type list of the **General** tab, the message is in the Binary format. • If **Any** is selected from the **Message Type** list of the **General** tab, the message is in the String or Binary format. **Note:** If **Simple** is selected from the **Message Type** list of the **General** tab, the **body** field is unavailable.

> **Important:** If you set the Flow limit for the AMQP Receive Message activity and are using the **RabbitMQ 0-9** broker type, refer to the section RabbitMQ Receiver Stops Receiving Messages.

System Property for Azure Service Bus

You can set the following system property for the AMQPReceiveMessage activity when **AzureSB-1-0** is selected from the **Broker Type** list of the used AMQP Connection shared resource:

Property	Description
com.tibco.plugin.amqp.maxMessageLimit	The maximum number of messages that can be fetched by the AMQPReceiveMessage activity for each request submitted to the Azure Service Bus server. The default value is 1.

AMQPSendMessage

You can use the AMQPSendMessage activity to send messages to a specified destination.

General

The following table describes the configurations on the **General** tab of the AMQPSendMessage activity.

Field	Module Property?	Description
Name	No	The name of the activity displayed in the process. The default value is `AMQPSendMessage`.
Connection	Yes	Click 🔍 to select an AMQP Connection shared resource. The AMQP Connection shared resource creates a connection between the plug-in and an AMQP server. If no matching AMQP Connection shared resources are found, click **Create Shared Resource** to create one. For more details, see Creating an AMQP Connection.
Protocol Version	No	AMQP protocol 1.0 (AMQP-1-0) is supported with Qpid-1-0, ActiveMQ-1-0, AMQ-1-0, ActiveMQ-Artemis-1-0, or AzureSB-1-0 broker. AMQP protocol 0.9 (AMQP-0-9) is supported with the RabbitMQ-0-9 broker.
Entity Type	No	Select an entity type from the following: • Queue • Topic **Note:** This field is not available when **RabbitMQ-1-0** is selected from the **Broker Type** list of the used AMQP Connection shared resource.
Entity Name	Yes	Enter the name of the queue or topic where the message is published. **Note:** This field is not available when **RabbitMQ-1-0** is selected from the **Broker Type** list of the used AMQP Connection shared resource.

Field	Module Property?	Description
Exchange Type	No	The type of the exchange. The valid types are listed as follows: • **direct** • **topic** • **fanout** • **headers** **Note:** This field is available only when **RabbitMQ-0-9** is selected from the **BrokerType** list of the used AMQP Connection shared resource.
Exchange Name	Yes	The name of the exchange. **Note:** This field is available only when **RabbitMQ-0-9** is selected from the **BrokerType** list of the used AMQP Connection shared resource. Default exchange is now supported and hence you can leave this field empty to achieve the same.
Routing Key	Yes	The routing key of the exchange. This field is only required when **direct** or **topic** is selected as the exchange type. **Note:** This field is available only when **RabbitMQ-0-9** is selected from the **BrokerType** list of the used AMQP Connection shared resource.
Message Type	No	Select one of the following message types: • **TextMessage**: the message is in the string format. • **BytesMessage**: the message is in the byte stream format. • **Simple**: the message does not have a message body.

Field	Module Property?	Description
Get MessageID	No	Select the **Get MessageID** checkbox to populate the message ID on the **Output** tab.
User Properties	No	Right-click inside the **User Properties** text box and click the **Add Parameter** ⊕ icon to add a user property. To delete a user property, click the **Delete** ✖ icon. Specify the following fields in this text box: • `name` (a String value): the name of the user property. • `type`: the property data type. The valid data types include String, Number, Boolean. • `value` (a String value): the value of the user property. **Note:** If the property data type is Boolean, any other value except for true or True or TRUE is considered false.

Description

On the **Description** tab, you can add a short description for the AMQPSendMessage activity.

Advanced

The following table describes the configurations on the **Advanced** tab of the AMQPSendMessage activity.

Field	Module Property?	Description
Delivery Mode	No	Select one of the following delivery modes: • **Persistent**: the messages are stored before forwarding. • **Non-Persistent**: the messages are not stored before

Field	Module Property?	Description
		forwarding. If the transmission failed, the messages might be lost.
Expiration (msec)	No	Corresponds to the Expiration property that specifies how long the message remains active (in milliseconds). The default value is `0`, which means the message remains active permanently.
Priority	No	Specifies the priority of the message. The value ranges from 0 to 9. The default value is `4`.
Type	Yes	Specifies the event type or command type that this message represents.

Input

The **Input** tab displays the input schema of the activity as a tree structure. The input values vary depending on the value selected for **Entity Type** on the **General** tab. The value specified for the **queueName** or **topicName** field on the **Input** tab takes precedence over the value specified for the **Entity Name** field on the **General** tab. The following table describes the input elements on the **Input** tab of the AMQPSendMessage activity.

Input Item	Data Type	Description
name	String	Enter the name of the user property. **Note:** The value entered in this field overwrites the value entered in the **User Properties** field of the **General** tab.
type	String	Enter one of the following types for the user property: • Boolean • Number

Input Item	Data Type	Description
		• String The value entered in this field overwrites the value entered in the **User Properties** field of the **General** tab.
value	String	Enter the value of the user property. The value entered in this field overwrites the value entered in the **User Properties** field of the **General** tab.
deliveryMode	String	Right-click this element and click **Enumerations** to select one of the following delivery modes for the message, or enter the value in this field: • Persistent The message is stored and forwarded. • Non-Persistent The message is not stored and might be lost due to transmission failure. The value entered in this field overwrites the item selected from the **Delivery Mode** list of the **Advanced** tab.
messageID	String	Enter the message ID for the message, such as `"ID:1234567891A"`. If you leave this field blank, a random message ID is assigned.
expiration	Integer	Enter the expiration value of the message. The value entered in this field overwrites the value entered in the **Expiration(msec)** field of the **Advanced** tab.
priority	Integer	Enter the priority level for sending the message.

Input Item	Data Type	Description
		The value entered in this field overwrites the value entered in the **Priority** field of the **Advanced** tab.
type	String	Enter the type for the message.
		The value entered in this field overwrites the value entered in the **Type** field of the **Advanced** tab.
contentType	String	The format of the message payload.
sessionID	String	The value that can enable an application to specify the session ID. This field is available only when **AzureSB-1-0** is selected from the **Broker Type** list of the used AMQP Connection shared resource.
correlationID	String	The value that can enable an application to specify the context for the message for correlation. This field is available only when **AzureSB-1-0** or **AMQ-1-0** is selected from the **Broker Type** list of the used AMQP Connection shared resource.
exchangeName	String	The name of the exchange. This field is available only when **RabbitMQ-0-9** is selected from the **Broker Type** list of the used AMQP Connection shared resource.
routingKey	String	The routing key of the exchange. This field is available only when **RabbitMQ-0-9** is selected from the **Broker Type** list of the used AMQP Connection shared resource.
queueName or topicName	String	Enter the name of the queue or the topic where you want to publish the message. **Note:** The value entered in this field overwrites the value entered in the **Entity Name** field of the **General** tab. This field is not available for the **RabbitMQ-0-9** broker.

Input Item	Data Type	Description
body	String/Byte	The message body of the received message: • If **TextMessage** is selected from the **Message Type** list of the **General** tab, the message is in the String format. • If **BytesMessage** is selected from the **Message Type** list of the **General** tab, the message is in the String format. **Note:** If **Simple** is selected from the **Message Type** list of the **General** tab, the **body** field is unavailable.

Output

The following table describes the output elements on the **Output** tab of the AMQPSendMessage activity.

Output Item	Data Type	Description
sendResult	Boolean	The result after sending the message. If the message is successfully sent, the value is true.
MessageId	String	A unique identifier of the message that is sent. **Note:** The **MessageId** field appears only if you select the **Get MessageID** checkbox on the **General** tab.

Fault

The following table describes the error schema elements on the **Fault** tab of the AMQPSendMessage activity.

Error Schema Element	Description
AMQPPluginException	This error message is displayed when the plug-in captures any

Error Schema Element	Description
	exception related to sending a message to an AMQP broker.
AMQPConnectionException	This error message is displayed when any exception occurs about the connection problem that is in sending a message to an AMQP broker.

Setting Prefetch Count for RabbitMQ Broker

The default value of the prefetch count is 20. You can change the prefetch count for the RabbitMQ broker through the system property.

Setting Prefetch Count for RabbitMQ Broker in TIBCO Business Studio™ at Run Time

1. From the TIBCO Business Studio™ menu, click **Run** > **Run Configurations**.

2. In the Run Configurations dialog, perform the following tasks:

   a. In the left pane, click **BusinessWorks Application** > **BWApplication**.

   b. In the right pane, click the **Arguments** tab.

   c. In the **VM arguments** field, add the following parameter:

   ```
   Dcom.tibco.plugin.amqp.rabbitmq.prefetch=<prefetch count>
   ```

Setting Prefetch Count for RabbitMQ Broker in TIBCO® Enterprise Administrator at Run Time

1. Navigate to the `TIBCO_HOME/bw/version_number/bin` directory and open the `bwcommon.tra` file.

2. Add the following parameter in the `java.extended.properties` file:

   ```
   com.tibco.plugin.amqp.rabbitmq.prefetech=<prefetch count>
   ```

Working with the Sample Project

The plug-in packages five sample projects with the installer. The sample projects show how the TIBCO ActiveMatrix BusinessWorks™ Plug-in for AMQP works.

After installing the plug-in, you can locate the sample file at `TIBCO_HOME`/bw/palettes/amqp/`version_number`/samples.

This sample file contains six projects:

- ActiveMQ_1-0

  This project shows how to use the ActiveMQ_1-0 broker to get, receive, and send messages.

- AMQ_1-0

  This project shows how to use the AMQ_1-0 broker to get, receive, and send messages.

- AzureServiceBus_1-0

  This project shows how to use the AzureServiceBus_1-0 broker to get, receive, and send messages.

- QPID_1-0

  This project shows how to use the QPID_1-0 broker to get, receive, and send messages.

- RabbitMQ_0-9

  This project shows how to use the RabbitMQ_0-9 broker to get, receive, and send messages.

- ActiveMQ-Artemis_1-0

  This project shows how to use the ActiveMQ-Artemis_1-0 broker to get, receive, and send messages.

Importing the Sample Project

Before running the project, you must import the sample project to TIBCO Business Studio™.

Procedure

1. Start TIBCO Business Studio™ in one of the following ways:

    *   Microsoft Windows: click **Start > All Programs > TIBCO >** *TIBCO_HOME* **> TIBCO Business Studio** *version_number* **> Studio for Designers**.

    *   macOS and Linux: run the TIBCO Business Studio™ executable file in the `TIBCO_ HOME`/studio/`version_number`/eclipse directory.

2. In TIBCO Business Studio™, on the menu bar, click **File > Import**.

3. In the Import wizard, expand the **General** folder. Select the **Existing Studio Projects into Workspace** item and click **Next**.

4. On the Import Projects wizard, click **Browse** next to the **Select archive file** field to select the example.zip file and click **Finish**. The example.zip file is by default at `TIBCO_HOME/bw/palettes/amqp/version_number/samples`.

Result

The sample projects are imported to TIBCO Business Studio™.
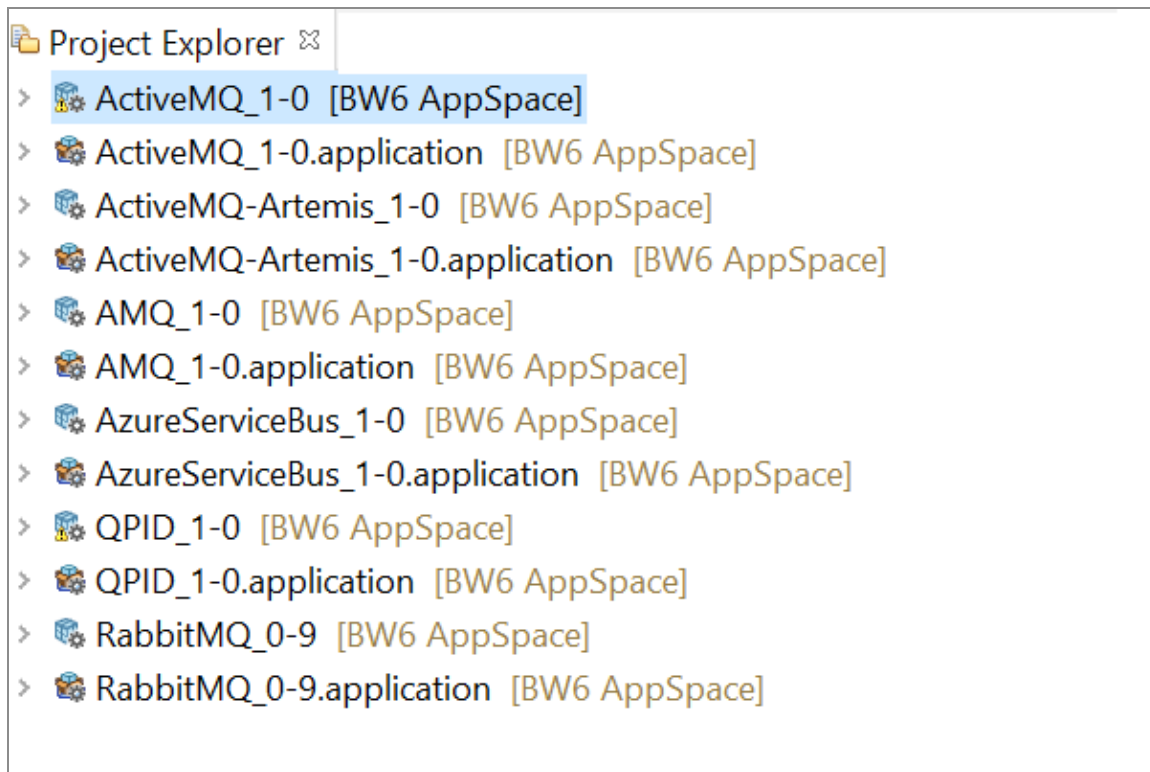
Configuring the AMQP Connection

Configuring the AMQP Connection shared resource is essential for the plug-in to establish a connection to the AMQP server.

Before you begin

Ensure that you have imported the sample projects, as described in Importing the Sample Projects.

Procedure

1. In the Project Explorer view, expand *Project_name* > **Resources** > *bundle_name*.

2. Double-click **AMQPConnectionResource.amqpconnectionResource**.

3. In the AMQP Connection editor, configure each field accordingly.

    For more details, see AMQP Connection.

4. Click **Test Connection** to validate the connection.

5. On the toolbar, click the **Save** 🖫 icon to save your changes.

Running the Sample Project

The sample projects show how to use the plug-in to get, receive, and send messages by using ActiveMQ-1-0, Qpid-1-0, AzureSB-1-0, AMQ-1-0, RabbitMQ-0-9, and ActiveMQ-Artemis_ 1-0 brokers.

Before you begin

Ensure that you have configured the AMQP Connection shared resource, as described in Configuring the AMQP Connection.

Procedure

1. In Project Explorer, click *Project_name* **> Module Descriptors** resource and then double-click **Components**.

2. By default, all the processes are listed in the Components Editor. Select the process that you do not want to run and click the **remove selected component** ❌ icon.

3. On the toolbar, click the **Save** 🖫 icon.

4. From the menu, click **Run > Run Configurations** to run the selected process.

5. In the Run Configurations dialog, expand **BusinessWorks Application**, and then click **BWApplication**.

6. In the right pane, click the **Applications** tab, and select the checkbox next to *Example_name*.

7. Click **Run** to run the selected process.

> ℹ️ **Note:** Click the **Terminate** 🟥 icon to stop the process.

Managing Logs

When an error occurs, check the logs to trace and troubleshoot the plug-in exceptions.

By default, error logs are displayed in the Console view when you run a process in debug mode. You can change the log level of the plug-in to trace different messages and export logs to a file. Different log levels correspond to different messages, as described in Log Levels.

Log Levels

Different log levels include different information.

The plug-in supports the following log levels.

Log Level	Description
Debug	Indicates a developer-defined tracing message.
Info	Indicates normal plug-in operations. No action is required. A tracing message tagged with Info indicates that a significant processing step is reached, and logged for tracking or auditing purposes. Only info messages preceding a tracking identifier are considered as significant steps.
Error	Indicates that an unrecoverable error occurred. Depending on the severity of the error, the plug-in might continue with the next operation or might stop.

Setting Up Log Levels

In the `logback.xml` file, you can configure a different log level for the plug-in and plug-in activities to trace different messages.

If you do not configure any log levels, the plug-in uses the default log level of TIBCO ActiveMatrix BusinessWorks. The default log level is `Error`.

Procedure

1. Navigate to the `TIBCO_HOME`/bw/`version_number`/config/design/logback directory and open the `logback.xml` file.

2. Add the following node in the **BusinessWorks Palette and Activity loggers** area to specify a log level for the plug-in:

   ```
   <logger name="com.tibco.bw.palette.amqp.runtime">
       <level value="DEBUG"/>
   </logger>
   ```

   The value of the `level` element can be `Error`, `Info`, or `Debug`.

   > ℹ **Note:** If you set the log level to Debug, the input and output for the plug-in activities are also displayed in the Console view. For more details about each log level, see Log Levels.

3. Add different nodes in the **BusinessWorks Palette and Activity loggers** area to specify different log levels for the activities and shared resources.

   For the AMQP Connection shared resource, add the following node:

   ```
   <logger name="com.tibco.bw.sharedresource.amqp.runtime">
       <level value="DEBUG"/>
   </logger>
   ```

   For the AMQPReceiveMessage activity, add the following node:

   ```
   <logger
   name="com.tibco.bw.palette.amqp.runtime.receive.AmqpReceiveEventSou
   rce">
       <level value="DEBUG"/>
   </logger>
   ```

   For the AMQPSendMessage activity, add the following node:

   ```
   <logger
   ```

```
name="com.tibco.bw.palette.amqp.runtime.send.AmqpSendActivity">
    <level value="DEBUG"/>
</logger>
```

For the AMQPGetMessage activity, add the following node:

```
<logger
name="com.tibco.bw.palette.amqp.runtime.getamqp.AmqpGetMessageActiv
ity">
    <level value="DEBUG"/>
</logger>
```

> **ℹ Note:** The activities that are not configured with specific log levels use the log level configured for the plug-in.

4. Save the file.

Exporting Logs to a File

You can update the `logback.xml` file to export plug-in logs to a file.

Procedure

1. Navigate to the *TIBCO_HOME*/bw/*version_number*/config/design/logback directory and open the `logback.xml` file.

   > **ℹ Note:** After deploying an application in TIBCO® Enterprise Administrator, navigate to the *TIBCO_HOME*/bw/*version_number*/domains/*domain_name*/appnodes/*space_name*/*node_name* directory to find the `logback.xml` file.

2. Add the following node to specify the file where the log is exported:

```
<appender name="FILE" class="ch.qos.logback.core.FileAppender">
    <file>c:/bw6-amqp.log</file>
        <encoder>
          <pattern>%d{HH:mm:ss.SSS} [%thread] %-5level %logger
```

```
{36}-%msg%n</pattern>
        </encoder>
</appender>
```

The value of the `file` element is the absolute path of the file that stores the exported log.

3. Add the following node to the `root` node at the bottom of the `logback.xml` file:

```
<root level="DEBUG">
    <appender-ref ref="STDOUT"/>
    <appender-ref ref="FILE"/>
</root>
```

4. Save the file.

Error Codes

The following table lists error codes, a detailed explanation of each error, and where applicable, ways to resolve different errors.

Error Code and Error Message	Role	Category	Description	Solution
ERROR_OCCURED_RETRIEVE_ RESULT.errorCode=500002 `IOException occurred while retrieving XML Output for activity [{0}].`	errorRole	BW-Plug-in	Occurs when an activity retrieves an XML output.	None
ERROR_OCCURED_INVOKE_ EXECUTE_ METHOD.errorCode=500003 `Exception occurred while invoking execute method for activity [{0}].`	errorRole	BW-Plug-in	Occurs when an activity invokes an execute method.	None
AMQP_SEND_ ERROR.errorCode=500007 `Error occurred while attempting to send a message {0}.`	errorRole	BW-Plug-in	Occurs when an activity sends a message.	None
ERROR_OCCURED_GET_XML_ FROM_SERIALIZED_ OBJECT.errorCode=500009	errorRole	BW-Plug-in	Occurs when an activity gets an XML output from a serialized	None

Error Code and Error Message	Role	Category	Description	Solution
`Error occurred while attempting to get XML Output from serialized object.`			object.	
ERROR_OCCURED_GET_MESSAGE_ CONTENT.errorCode=500010 `Error occurred while attempting to get message content: {0}.`	errorRol e	BW-Plug-in	Occurs when an activity gets a message content.	None
ERROR_OCCURED_START_ CONNECTION.errorCode=500011 `Error occurred while attempting to start javax.jms.Connection: {0}.`	errorRol e	BW-Plug-in	Occurs when a javax.jms.connecti on is started.	None
ERROR_OCCURED_STOP_ CONNECTION.errorCode=500012 `Error occurred while attempting to stop javax.jms.Connection: {0}.`	errorRol e	BW-Plug-in	Occurs when a javax.jms.connecti on is stopped.	None
ERROR_OCCURED_CLOSE_ CONNECTION.errorCode=500013 `Error occurred while attempting to close javax.jms.Connection:`	errorRol e	BW-Plug-in	Occurs when a javax.jms.connecti on is closed.	None

Error Code and Error Message	Role	Category	Description	Solution
`{0}.`				
ERROR_OCCURED_GENERATE_ CONNECTION_ FILE.errorCode=500014 `Error occurred while attempting to generate connection file: {0}.`	errorRole	BW-Plug-in	Occurs when the plug-in generates a connection file.	None
ERROR_OCCURED_CREATE_ CONNECTION.errorCode=500015 `Error occurred while attempting to create connection: {0}.`	errorRole	BW-Plug-in	Occurs when the plug-in creates a connection.	None
ERROR_OCCURED_INITIALIZE_ CONNECTIONFACTORY.errorCode= 500016 `Error occurred while attempting to initialize a connection factory: {0}.`	errorRole	BW-Plug-in	Occurs when the plug-in initializes a connection factory.	None
ERROR_OCCURED_CREATE_TEXT_ MESSAGE.errorCode=500017 `Error occurred while attempting to create text message from Session.`	errorRole	BW-Plug-in	Occurs when the plug-in creates a text message from a session.	None

Error Code and Error Message	Role	Category	Description	Solution
ERROR_OCCURED_RECEIVE_ MESSAGE.errorCode=500018 `Error occurred while receive message: {0}.`	errorRol e	BW-Plug- in	Occurs when an activity receives a message.	None
ERROR_OCCURED_INITIALIZE_ CHANNEL.errorCode=500019 `Error occurred while attempting to initialize channel : {0}.`	errorRol e	BW-Plug- in	Occurs when the plug-in initializes a channel.	None
ERROR_OCCURED_CLOSE_ CHANNEL.errorCode=500020 `Error occurred while attempting to close channel.`	errorRol e	BW-Plug- in	Occurs when the plug-in closes a channel.	None
ERROR_OCCURED_FILE_NOT_ FIND.errorCode=500021 `Error occurred while attempting to find file {0}.`	errorRol e	BW-Plug- in	Occurs when the plug-in finds a file.	None
ERROR_OCCURED_INITIALIZE_ BROKER_TYPE.errorCode=500022 `Error occurred while attempting to initialize Broker type.`	errorRol e	BW-Plug- in	Occurs when the plug-in initializes a broker type.	None

Error Code and Error Message	Role	Category	Description	Solution
ERROR_OCCURED_CLOSE_ CHANNEL_ TIMEOUT.errorCode=500023 `Timeout error occurred while attempting to close channel.`	errorRol e	BW-Plug- in	A timeout error occurs when closing a channel.	None
ERROR_OCCURED_CREATE_ CONNECTION_ TIMEOUT.errorCode=500024 `Timeout error occurred while attempting to create new connection.`	errorRol e	BW-Plug- in	A timeout error occurs when creating a connection.	None
ERROR_OCCURED_USER_ PROPERTY_ INVALID.errorCode=500025 `Data type and value mismatch error occurred for user property name: {0}.`	errorRol e	BW-Plug- in	The data type and value do not match the user property name.	None
ERROR_OCCURED_SEND_ MESSAGE_ UNKNOWN.errorCode=500026 `Error occurred while attempting to send message for unknown issues: {0}.`	errorRol e	BW-Plug- in	Occurs when an activity sends a message for unknown issues.	None

Error Code and Error Message	Role	Category	Description	Solution
ERROR_OCCURED_SEND_ MESSAGE_INVALID_ DESTINATION.errorCode=500027 `Invalid Queue name: {0} detected. Please provide valid Queue name.`	errorRole	BW-Plug-in	The provided queue name is invalid.	Provide a valid queue name.
ERROR_OCCURED_SEND_ MESSAGE_CREATE_QUEUE_ NAME=500028 `Error occurred while attempting to create queue name: {0}.`	errorRole	BW-Plug-in	Occurs when the plug-in creates a queue name.	None
ERROR_OCCURED_SEND_ MESSAGE_GET_QUEUE_ NAME=500029 `Error occurred while attempting to get queue name.`	errorRole	BW-Plug-in	Occurs when the plug-in gets a queue name.	None
ERROR_OCCURED_CLOSE_ SESSION.errorCode=500031 `Error occurred while attempting to close session.`	errorRole	BW-Plug-in	Occurs when the plug-in closes a session.	None
ERROR_OCCURED_DUE_TO_ SECURITY_INFO.errorCode=500032	errorRole	BW-Plug-in	Occurs when incorrect connection details	Provide correct connection

Error Code and Error Message	Role	Category	Description	Solution
Error occurred due to an unauthorized access. Check connection Properties.			are configured in the Shared Resource being used by the activity.	details in the Shared Resource and make sure that the configuration is correct by clicking 'Test Connection'.
ERROR_OCCURED_DUE_TO_ ILLEGAL_INFO.errorCode=500033 Error occurred due to illegal state for the requested operation {0}.	errorRole	BW-Plug-in	Occurs when incorrect information is provided for the requested operation for queue and topic subscribers.	Provide correct information to the plug-in based on the error message details.

Troubleshooting

Issue: AMQP Receive Message activity for RabbitMQ broker stops receiving messages when Flow Limit is applied.

Summary: The RabbitMQ receiver stops receiving new messages after processing the initial messages in the queue when the Flow limit is set. This occurs because the receiver closes after consuming the pending messages and does not receive further messages without restarting the application.

Resolution:

1. Set the custom property `com.tibco.bw.amqp.rabbitmq.poll.delay.limit` to define the maximum delay. The default value is 5 seconds.

2. Set `sessioncount`=*1* when applying the `FlowLimit` for RabbitMQ connections.

3. Apply `BW FlowLimit` at either the application or component level, regardless of the `Prefetch` value.

   a. To apply the Flow limit at the component level for one process: Enable the Flow limit at the component level and set the property *<process_name>*`.job.flowlimit.enabled` to *true*.

   b. To apply the Flow limit at the component level for multiple processes: Enable the Flow limit at the component level and set the property *<process1-name>.<process2-name>.<process3-name>*`.job.flowlimit.enabled` to *true*.

   > ❗ **Important:** Ensure to apply the property `–D`*<process_name>*`.Receiver.job.flowlimit.enabled`=*true* along with the component-level Flow limit property.

   c. To apply the Flow limit at the application level, refer to the example properties below.

   Example properties:

   ```
   –Dbw.application.job.flowlimit.testRabbitMQ.application=1
   –Dbw.application.job.flowlimit.testRabbitMQ.application.1.0=1
   ```

```
-
Dbw.application.job.flowlimit.testRabbitMQ.application.1.0.Com
ponentReceiver=1 (where "ComponentReceiver" is the component
name as it appears in the components view inside "Module
Descriptors")

-Dtestrabbitmq.Receiver.job.flowlimit.enabled=true
```

TIBCO Documentation and Support Services

For information about this product, you can read the documentation, contact TIBCO Support, and join TIBCO Community.

How to Access TIBCO Documentation

Documentation for TIBCO products is available on the Product Documentation website, mainly in HTML and PDF formats.

The Product Documentation website is updated frequently and is more current than any other documentation included with the product.

Product-Specific Documentation

The documentation for this product is available on the TIBCO ActiveMatrix BusinessWorks™ Plug-in for AMQP Product Documentation page.

How to Contact Support for TIBCO Products

You can contact the Support team in the following ways:

- To access the Support Knowledge Base and getting personalized content about products you are interested in, visit our product Support website.

- To create a Support case, you must have a valid maintenance or support contract with a Cloud Software Group entity. You also need a username and password to log in to the product Support website. If you do not have a username, you can request one by clicking **Register** on the website.

How to Join TIBCO Community

TIBCO Community is the official channel for TIBCO customers, partners, and employee subject matter experts to share and access their collective experience. TIBCO Community offers access to Q&A forums, product wikis, and best practices. It also offers access to extensions, adapters, solution accelerators, and tools that extend and enable customers to gain full value from TIBCO products. In addition, users can submit and vote on feature

requests from within the TIBCO Ideas Portal. For a free registration, go to TIBCO Community.

Legal and Third-Party Notices

SOME CLOUD SOFTWARE GROUP, INC. ("CLOUD SG") SOFTWARE AND CLOUD SERVICES EMBED, BUNDLE, OR OTHERWISE INCLUDE OTHER SOFTWARE, INCLUDING OTHER CLOUD SG SOFTWARE (COLLECTIVELY, "INCLUDED SOFTWARE"). USE OF INCLUDED SOFTWARE IS SOLELY TO ENABLE THE FUNCTIONALITY (OR PROVIDE LIMITED ADD-ON FUNCTIONALITY) OF THE LICENSED CLOUD SG SOFTWARE AND/OR CLOUD SERVICES. THE INCLUDED SOFTWARE IS NOT LICENSED TO BE USED OR ACCESSED BY ANY OTHER CLOUD SG SOFTWARE AND/OR CLOUD SERVICES OR FOR ANY OTHER PURPOSE.

USE OF CLOUD SG SOFTWARE AND CLOUD SERVICES IS SUBJECT TO THE TERMS AND CONDITIONS OF AN AGREEMENT FOUND IN EITHER A SEPARATELY EXECUTED AGREEMENT, OR, IF THERE IS NO SUCH SEPARATE AGREEMENT, THE CLICKWRAP END USER AGREEMENT WHICH IS DISPLAYED WHEN ACCESSING, DOWNLOADING, OR INSTALLING THE SOFTWARE OR CLOUD SERVICES (AND WHICH IS DUPLICATED IN THE LICENSE FILE) OR IF THERE IS NO SUCH LICENSE AGREEMENT OR CLICKWRAP END USER AGREEMENT, THE LICENSE(S) LOCATED IN THE "LICENSE" FILE(S) OF THE SOFTWARE. USE OF THIS DOCUMENT IS SUBJECT TO THOSE SAME TERMS AND CONDITIONS, AND YOUR USE HEREOF SHALL CONSTITUTE ACCEPTANCE OF AND AN AGREEMENT TO BE BOUND BY THE SAME.

This document is subject to U.S. and international copyright laws and treaties. No part of this document may be reproduced in any form without the written authorization of Cloud Software Group, Inc.

TIBCO, the TIBCO logo, the TIBCO O logo, ActiveMatrix BusinessWorks, Business Studio, and TIBCO Business Studio are either registered trademarks or trademarks of Cloud Software Group, Inc. in the United States and/or other countries.

All other product and company names and marks mentioned in this document are the property of their respective owners and are mentioned for identification purposes only. You acknowledge that all rights to these third party marks are the exclusive property of their respective owners. Please refer to Cloud SG's Third Party Trademark Notices (https://www.cloud.com/legal) for more information.

This document includes fonts that are licensed under the SIL Open Font License, Version 1.1, which is available at: https://scripts.sil.org/OFL

Copyright (c) Paul D. Hunt, with Reserved Font Name Source Sans Pro and Source Code Pro.

Cloud SG software may be available on multiple operating systems. However, not all operating system platforms for a specific software version are released at the same time. See the "readme" file for the availability of a specific version of Cloud SG software on a specific operating system platform.