

# **TIBCO ActiveMatrix BusinessWorks™ Plug-in for Apache Kafka User's Guide**

*Software Release 6.0  
May 2017*

## Important Information

SOME TIBCO SOFTWARE EMBEDS OR BUNDLES OTHER TIBCO SOFTWARE. USE OF SUCH EMBEDDED OR BUNDLED TIBCO SOFTWARE IS SOLELY TO ENABLE THE FUNCTIONALITY (OR PROVIDE LIMITED ADD-ON FUNCTIONALITY) OF THE LICENSED TIBCO SOFTWARE. THE EMBEDDED OR BUNDLED SOFTWARE IS NOT LICENSED TO BE USED OR ACCESSED BY ANY OTHER TIBCO SOFTWARE OR FOR ANY OTHER PURPOSE.

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

This document contains confidential information that 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 TIBCO Software Inc.

TIBCO, Two-Second Advantage, The Power of Now, TIB, Information Bus, Rendezvous, TIBCO Rendezvous, and Messaging Appliance are either registered trademarks or trademarks of TIBCO Software Inc. in the United States and/or other countries.

Enterprise Java Beans (EJB), Java Platform Enterprise Edition (Java EE), Java 2 Platform Enterprise Edition (J2EE), and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle Corporation in the U.S. and 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.

THIS 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 THIS SOFTWARE VERSION ON A SPECIFIC OPERATING SYSTEM PLATFORM.

THIS DOCUMENT IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT.

THIS DOCUMENT COULD INCLUDE TECHNICAL INACCURACIES OR TYPOGRAPHICAL ERRORS. CHANGES ARE PERIODICALLY ADDED TO THE INFORMATION HEREIN; THESE CHANGES WILL BE INCORPORATED IN NEW EDITIONS OF THIS DOCUMENT. TIBCO SOFTWARE INC. MAY MAKE IMPROVEMENTS AND/OR CHANGES IN THE PRODUCT(S) AND/OR THE PROGRAM(S) DESCRIBED IN THIS DOCUMENT AT ANY TIME.

THE CONTENTS OF THIS DOCUMENT MAY BE MODIFIED AND/OR QUALIFIED, DIRECTLY OR INDIRECTLY, BY OTHER DOCUMENTATION WHICH ACCOMPANIES THIS SOFTWARE, INCLUDING BUT NOT LIMITED TO ANY RELEASE NOTES AND "READ ME" FILES.

Copyright © 2017 TIBCO Software Inc. ALL RIGHTS RESERVED.

TIBCO Software Inc. Confidential Information

# Contents

---

<b>TIBCO Documentation and Support Services</b> .....	<b>4</b>
<b>Product Overview</b> .....	<b>5</b>
<b>Getting Started</b> .....	<b>7</b>
Creating a Project .....	7
Creating a Kafka Connection .....	8
Configuring a Process .....	9
Testing a Process .....	10
Deploying an Application .....	10
<b>Apache Kafka Palette</b> .....	<b>12</b>
Kafka Connection Shared Resource .....	12
Kafka SendMessage Activity .....	13
Fault Tolerance .....	17
Kafka ReceiveMessage Activity .....	18
<b>Sample Project Overview</b> .....	<b>22</b>
Working with Kafka_GeneralSample Project .....	22
Configuring Kafka Connection for Kafka_GeneralSample .....	23
Configuration of the Kafka_GeneralSample Processes .....	23
Running the Kafka_GeneralSample Project .....	24
Working with Kafka_CloudSample Project .....	25
Configuring Kafka Connection for Kafka_CloudSample .....	26
Configuration of the Kafka_CloudSample Processes .....	26
Running the Kafka_CloudSample Project .....	28
<b>Log Management</b> .....	<b>30</b>
Log Levels .....	30
Setting up Log Levels .....	30
Exporting Logs to a File .....	31
<b>Error Codes</b> .....	<b>32</b>

# TIBCO Documentation and Support Services

---

Documentation for this and other TIBCO products is available on the TIBCO Documentation site. This site is updated more frequently than any documentation that might be included with the product. To ensure that you are accessing the latest available help topics, visit:

<https://docs.tibco.com>

## Product-Specific Documentation

The following documents for this product can be found on the TIBCO Documentation site: <https://docs.tibco.com/products/tibco-activematrix-businessworks-plug-in-for-apache-kafka>.

- *TIBCO ActiveMatrix BusinessWorks Plug-in for Kafka Installation*
- *TIBCO ActiveMatrix BusinessWorks Plug-in for Kafka User's Guide*
- *TIBCO ActiveMatrix BusinessWorks Plug-in for Kafka Release Notes*

## How to Contact TIBCO Support

For comments or problems with this manual or the software it addresses, contact TIBCO Support:

- For an overview of TIBCO Support, and information about getting started with TIBCO Support, visit this site:

<http://www.tibco.com/services/support>

- If you already have a valid maintenance or support contract, visit this site:

<https://support.tibco.com>

Entry to this site requires a user name and password. If you do not have a user name, you can request one.

## How to Join TIBCO Community

TIBCO Community is an online destination for TIBCO customers, partners, and resident experts. It is a place to share and access the collective experience of the TIBCO community. TIBCO Community offers forums, blogs, and access to a variety of resources. To register, go to the following web address:

<https://community.tibco.com>

## Product Overview

Apache Kafka is a distributed messaging system, providing fast, highly scalable, and redundant messaging through a publisher-subscriber model.

It is available and resilient to node failures and supports automatic recovery. Apache Kafka has robust queues that can handle a high volumes of data and has an enabler to pass on the messages from one endpoint to another. Apache Kafka is suitable for both offline and online message consumption. Apache Kafka is built on top of the Apache ZooKeeper™ synchronization service. All Kafka messages are organized into topics.

TIBCO ActiveMatrix BusinessWorks™ Plug-in for Apache Kafka plugs into TIBCO ActiveMatrix BusinessWorks™. The Apache Kafka palette can be used to create producers, consumers, and perform send message and receive message operations.

The plug-in provides the following main features:

- Kafka Connection Shared Resource

Kafka connection shared resource is used to connect and fetch the list of topics from the Kafka server. The shared resource is used by the other activities for configuring the connection.

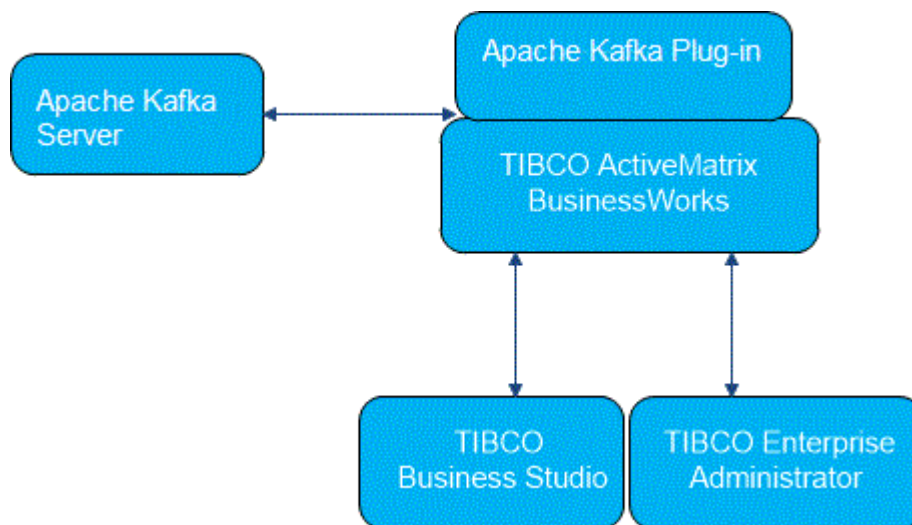
- Kafka SendMessage Activity

This activity is used to send messages to the Kafka consumer. It performs as a Kafka producer, which sends the message to a specified topic, and consumers can fetch the message from the specified topics.

- Kafka ReceiveMessage Activity

This activity is a process starter activity that starts the process execution on receiving a Kafka message.

The following figure describes the relationship between the Apache Kafka Server, TIBCO ActiveMatrix BusinessWorks™ Plug-in for Apache Kafka, and TIBCO ActiveMatrix BusinessWorks™.



The following list describes the relationship between different products in the previous figure.

- TIBCO ActiveMatrix BusinessWorks™ Plug-in for Apache Kafka communicates with the Kafka server instance.
- TIBCO ActiveMatrix BusinessWorks™ Plug-in for Apache Kafka plugs into TIBCO ActiveMatrix BusinessWorks™ and connects to a Kafka server instance.

- TIBCO ActiveMatrix BusinessWorks™ is an easy-to-use integration product suite for enterprise applications.
- TIBCO Business Studio™ is the graphical user interface (GUI) used by TIBCO ActiveMatrix BusinessWorks™ and the plug-in to design business processes, and it is also the process engine used to execute them.
- 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 to get users started with TIBCO ActiveMatrix BusinessWorks™ Plug-in for Apache Kafka in TIBCO Business Studio.

All the operations are performed in TIBCO Business Studio. Refer to *Tibco ActiveMatrix BusinessWorks™ Concepts* Guide to get familiar with TIBCO Business Studio.

The basic steps to create and deploy an application using TIBCO ActiveMatrix BusinessWorks™ Plug-in for Apache Kafka include:

1. [Creating a Project](#)
2. [Creating a Kafka Connection](#)
3. [Configuring a Process](#)
4. [Testing a Process](#)
5. [Deploying an Application](#)

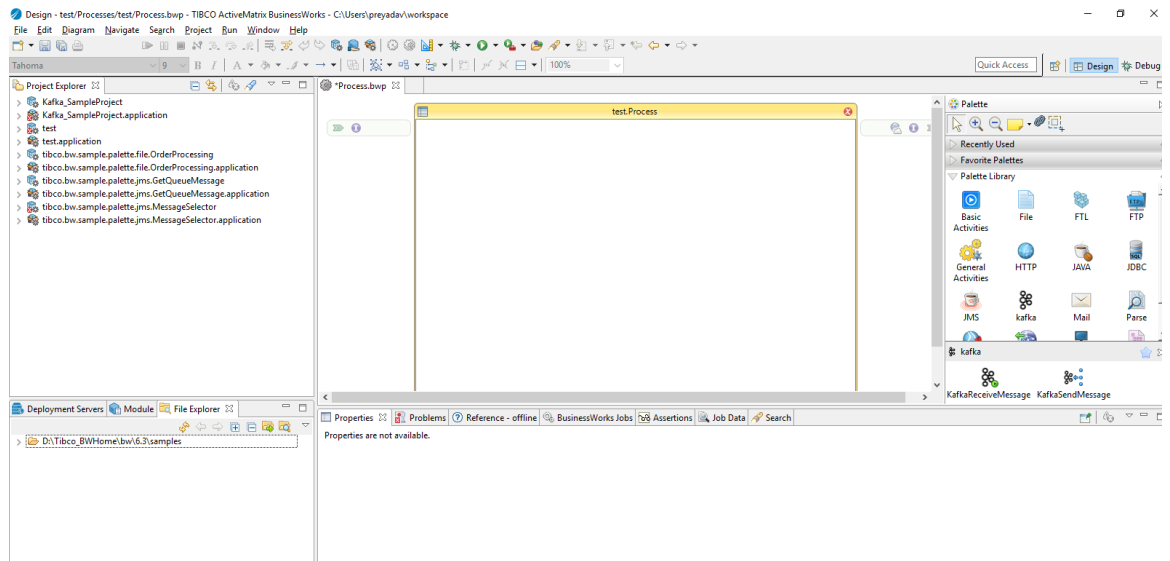
## Creating a Project

The first task for using a plug-in is creating a project. You can add resources and processes after creating a project.

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

### Procedure

1. Start TIBCO Business Studio by using one of the following ways:
  - Microsoft Windows: click **Start > All Programs > TIBCO > TIBCO\_HOME > TIBCO Business Studio version\_number > Studio for Designers**.
  - Linux: run the TIBCO Business Studio 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** check box 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** check boxes selected to automatically create an empty process and an application when creating the project.
  - e) Select the **Use Java configuration** check box 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 a Kafka Connection

After creating a project, you have to create a Kafka Connection shared resource to connect to a Kafka server instance.

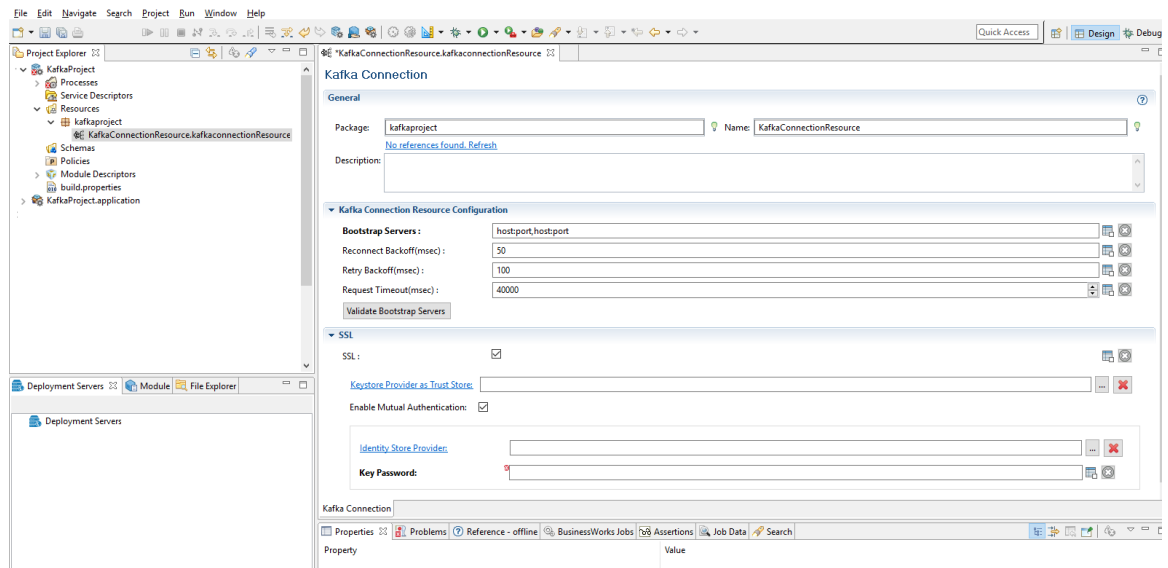
### Prerequisites

Kafka Connection shared resource is available at the **Resources** level. Create the 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 > Kafka Connection** to open the Kafka Connection wizard. The resource folder, package name, and resource name of the Kafka connection are provided by default, which can be customized.
3. Click **Finish** to open the Kafka Connection editor.
4. Configure the Kafka Connection shared resource in the Kafka Connection editor.
5. Click **Validate Bootstrap Servers** to validate the connection.





## Configuring a Process

An empty process is created by default when creating a project. You can add activities to the empty process.

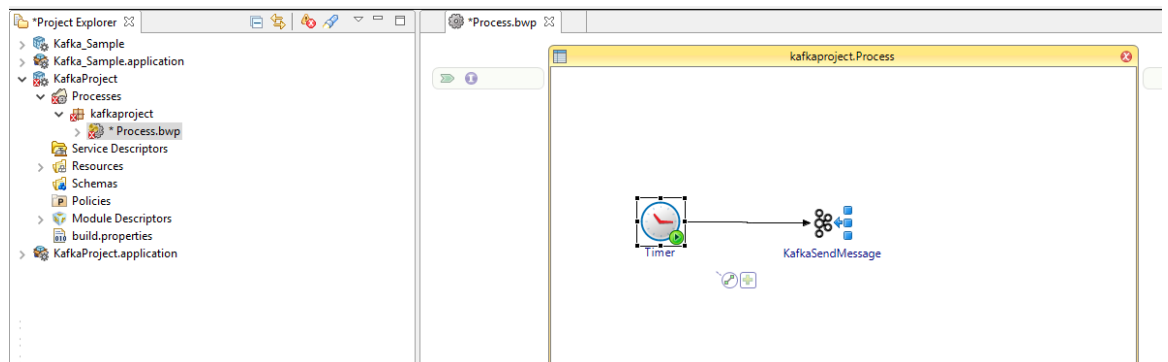
### Prerequisites


Ensure that you have created an empty process when creating a project. See [Creating a Project](#) for more information.

### 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 KafkaSendMessage activity from the Kafka palette library.



3. Click an activity in the Process editor and drag the  icon to create a transition between the added activities.
4. Configure the added Kafka activities, as described in [Kafka Palette](#).
5. Click **File > Save** to save the process.


## Testing a Process

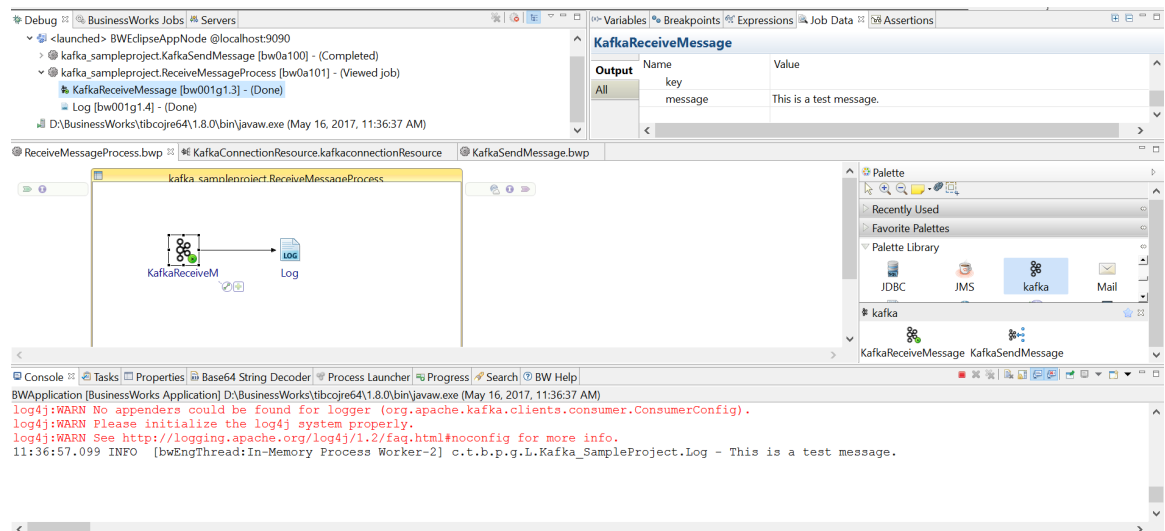
After configuring, you might debug or test a process.

### Prerequisites

Ensure that you have configured a process, as described in [Configuring a Process](#).

### Procedure

1. Open the process you have configured in the TIBCO Business Studio.
2. On the toolbar, click  **Debug > Debug Configurations**.
3. Click **BusinessWorks Application > BWApplication** in the left panel.  
By default, all the applications in the current workspace are selected in the **Applications** tab. Ensure that only the application you want to debug is selected in the **Applications** tab in the right panel.
4. Click **Debug** to test the process in the selected application.  
TIBCO Business Studio changes to the Debug perspective. The debug information is displayed in the Console view.
5. In the **Debug** tab, expand the debugged process and click an activity, such as the **KafkaReceiveMessage** process.
6. In the upper-right panel, click **Job Data** view, and then click the **Output** tab to check the activity output.



## Deploying an Application

After testing, 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.

### Prerequisites

Before deploying an application, generate an application archive, which is an enterprise archive (EAR) file created in TIBCO Business Studio.

Deploying an application involves:

**Procedure**

1. Uploading an application archive.
2. Deploying an application archive.
3. Starting an application.

See [TIBCO ActiveMatrix BusinessWorks Administration](#) for more details on how to deploy an application.

# Apache Kafka Palette

This plug-in contains one shared connection and two activities.

- [Kafka Connection Shared Resource](#)
- [Kafka SendMessage activity](#)
- [Kafka ReceiveMessage activity](#)

All Apache Kafka Plug-in activities can use the same Kafka Shared Connection resource and can work independently.

## Kafka Connection Shared Resource

Kafka Connection resource is used to specify the configuration details of the Kafka server hosted across various nodes. Typically, a producer would publish the messages to a specific topic hosted on a server node of a Kafka cluster and consumer can subscribe to any specific topic to fetch the data.

### General

In the **General** tab, you can specify the required parameters before you use this activity. The **General** tab contains the following fields:

Field	Description
Package	Package path to be added
Name	The name to be displayed as the label for the resource
Description	A short description of the shared resource

### Kafka Connection Resource Configuration

The Kafka Connection Configuration section has the following fields:

Field	Literal Value/Module Property?	Description
Bootstrap Servers	Yes	A list of host/port pairs to use for establishing the initial connection to the Kafka cluster.
Reconnect Backoff(msec)	Yes	This is the amount of time to wait before attempting to reconnect to a given host. This avoids repeatedly connecting to a host in a tight loop.  The default value is 50.
Retry Backoff(msec)	Yes	The amount of time to wait before attempting to retry a failed fetch request to a given topic partition. This avoids repeated fetching-and-failing in a tight loop.  The default value is 100.

Field	Literal Value/Module Property?	Description
Request Timeout(msec)	Yes	The amount of time in milliseconds after which metadata is refreshed forcefully even though there are no changes in the partition leadership, to proactively discover any new brokers or partitions.  The default value is 40000.

## SSL

Add required SSL properties in the `server.properties` file to enable SSL.

The SSL section has the following fields:


Field	Literal Value/Module Property	Description
SSL	Yes	Select this check box to use the SSL authentication to verify the user and the server.
Keystore Provider as Trust Store	None	This field is used to create KeystoreProviderResource and then provide trust store URL and password. Available only when <b>SSL</b> check box is selected.
Enable Mutual Authentication	None	Select this check box to provide two-way SSL authentication. Available only when <b>SSL</b> checkbox is selected.
Identity Store Provider	None	This field is used to create KeystoreProviderResource and then provide key store URL and password. Available only when <b>Enable Mutual Authentication</b> check box is selected.
Key Password	Yes	Specify the key password. Available only when <b>Enable Mutual Authentication</b> check box is selected.

## Kafka SendMessage Activity

The Kafka SendMessage activity is used to send or publish messages to consumers through Kafka brokers.

### General

In the **General** tab, you can specify the required parameters before you use this activity. The **General** tab contains the following fields:

Field	Literal Value/ Module Property?	Description
Name	None	The name to be displayed as the label for the activity in the process.
Kafka Connection	Yes	The Kafka connection resource provides the connection details to communicate with a Kafka server Instance.
Topic Name	Yes	Provide the topic name where the Kafka cluster stores streams of records.
Assign Custom Partition	None	This is a check box to select if <b>Partition ID</b> needs to be entered. You can select the check box to override Kafka's default partition assignment behavior.
Partition ID	Yes	<p>Sequence ID of the custom partition to which the message is sent. The default value is 0.</p> <div>  <p>This field is enabled only if <b>Assign Custom Partition</b> is checked.</p> </div>
Key Serializer	Yes	This is a serializer class for key that implements the serializer interface.
Value Serializer	Yes	Value for the serializer interface.
Acks	Yes	<p>When writing to Kafka, producers can choose whether they wait for the message to be acknowledged by 0, 1 or all replicas. This controls the durability of records that are sent. This configuration controls the criteria under which producer requests are considered complete.</p> <ul style="list-style-type: none"> <li>acks=0. The producer does not wait for any acknowledgment from the Kafka server. The sent record is added to buffer and considered sent. In this scenario, no guarantee can be made that server has received the record, and retries do not take any effect.</li> <li>acks=1. The record is added to the log. In this case if the leader has written the record to its local log, it responds without waiting for follower's acknowledgment.</li> <li>acks=all. An acknowledgment of the record is sent if the data is committed by all the in-sync replicas. This is the strongest available guarantee.</li> </ul>
Buffer Memory	Yes	<p>This is total amount of memory available to the producer to buffer records waiting to be sent to the server.</p> <p>The default value is 33554432.</p>

Field	Literal Value/ Module Property?	Description
Compression Type	Yes	If enabled, data will be compressed by the producer, written in compressed format on the server, and decompressed by the consumer. The default is none.
Retries	Yes	Specifies the number of retries that are made by a client to resend any record in event of a transient error.

### Description

In the **Description** tab, you can enter a short description for the activity.

Field	Literal Value/Process Property/Module Property?	Description
Description	None	A description of the activity.

### Advanced

Specify the Batch Size, Client ID, Linger, Max Request Size, and Properties.


Field	Literal Value/ Module Property?	Description
Batch Size	None	Records are batched together by producer whenever multiple records are sent to the same partition. No attempt is made to batch the records larger than this size.  The default value is 16384.
Client ID	Yes	A String client ID is passed to the server while making the requests to track the source of the requests.
Linger	Yes	You can set linger.ms to something greater than 0 to instruct the producer to wait up to that number of milliseconds before sending a request in hope that more records arrive to fill up the same batch. The default behavior is to send messages immediately even if there is additional space in the buffer.
Max Request Size	Yes	Limits the number of record batches the producer sends in a single request to avoid sending huge requests.  The default value is 1048576.
Properties	None	Provide the properties name and value for the producer.



Kafka plug-in supports [Fault Tolerance](#). In case of any errors, you can use **Properties** to add the correct name and value.

## Input

The following table describes the fields in the **Input** tab of the SendMessage activity.

Field	Data Type	Description
ProducerConfig (All fields in this section are optional.)		
Topic	String	Topic name where Kafka cluster stores streams of records.
Partition ID	Number	Sequence ID of the partition to which the message is sent. Sequence ID of the partition can be entered here or in the <b>Partition ID</b> field under <b>General</b> tab.  The <b>Assign Custom Partition</b> check box under <b>General</b> tab must be selected to enter a value here.
Batch Size	Number	Batch size detail to batch the records sent to same partition.
Client ID	String	Client ID passed to the server while making the request.
Linger.ms	Number	Time in millisecond to add artificial delay while sending the message from producer.
Max Request Size	Number	Maximum size of a request in bytes. Use it to limit the number of record batches the producer sends in a single request to avoid sending huge requests.
Additional properties	String	<ul style="list-style-type: none"> <li>• Key</li> <li>• Value</li> </ul> <ul style="list-style-type: none"> <li>• Provide key for additional properties.</li> <li>• Provide value for specified key in additional properties.</li> </ul>
Messages		
<ul style="list-style-type: none"> <li>• Key</li> <li>• Message</li> </ul>	String	<ul style="list-style-type: none"> <li>• Optional. Key for the message to be sent.</li> <li>• Required. The actual message as value for the specified key.</li> </ul>

## Output

The following table describes the fields in the Output tab of the SendMessage activity.

Output Item	Data Type	Description
KafkaSendMessageOutput	complex	The complete output of the SendMessage activity.
result	complex	Information about the content of the sent and failed messages.
status	string	Status of the message sent by a producer.



Output Item	Data Type	Description
SendSuccess	complex	Information about the content of the sent message.
topic	string	The topic name for publishing the message.
offset	number	The sequence ID number assigned to each record within a partition.
Partition	number	The sequence ID of the partition to which a record is sent within a topic.
SendFailed	complex	Information about the content of the failed message.
errorCode	string	Displays the error code.
errorMessage	string	Displays the error message.

### Fault

The **Fault** tab has the following exceptions:

- KafkaCreatedProducerException
- KafkaPluginException

Each exception has the following fields:

Field	Type	Description
msg	string	The error message description returned by the plug-in.
msgCode	string	The error code returned by the plug-in.

### Fault Tolerance

Kafka plug-in supports fault tolerance.

This is an example to understand the fault tolerance feature.

- Two brokers (A & B)
- Partitions (P0 & P1)
- P0 - Leader is A, Slave is B
- P1 - Leader is B, Slave is A

Two brokers (A & B) - A topic having 2 partitions (P0 & P1) and 2 replication factors where each broker is acting as a leader for each of the partitions.

When you send messages on a particular partition (P0), leader (A) of that partition (P0) is terminated abruptly. As a result, for partition (P0), new leader (B) gets elected internally and messages are continuously sent and received on Kafka plug-in without any error.

As A is terminated for P0, Slave B gets elected as a leader and messages are seamlessly sent and received on Kafka plug-in.

The plug-in supports multiple brokers, partitions, and replication factors.



In case of any errors, try increasing the ZooKeeper timeout using any one of the following:


- In `server.properties` file, increase the connection time of property `zookeeper.connection.timeout.ms`.
- Add property `zookeeper.connection.timeout.ms` in **Advanced** tab and increase the value (connection time) to support Kafka fault tolerance.

## Kafka ReceiveMessage Activity

Kafka ReceiveMessage activity is an event source activity which can be configured as a process starter in any TIBCO BusinessWorks process. It starts the process execution on receiving a Kafka message event.

### General

In the **General** tab, specify the required parameters before using this activity. The **General** tab has the following fields:

Field	Literal Value/ Process Property/ Module Property?	Description
Name	None	The name to be displayed as the label for the activity in the process.
Kafka Connection	Yes	The Kafka connection resource for communicating with a Kafka server instance.
Group Id	Yes	The group ID for the consumer group.
Topic Names	Yes	The topic name where Kafka cluster stores streams of record.
Assign Custom Partition	None	This is a check box to select if <b>Partition ID</b> needs to be entered. You can select the check box to override Kafka's default partition assignment behavior.
Partition ID	Yes	Sequence ID of the custom partition to which the message is sent. Default is 0.  This field is enabled only if <b>Assign Custom Partition</b> is checked.
Key Deserializer	Yes	Class for the key that implements the serializer interface.
Value Deserializer	Yes	Value for the serializer interface.
Fetch Timeout	Yes	Specifies the maximum time in milliseconds to get the metadata about the topic before a timeout occurs. The default value is 1000.

Field	Literal Value/ Process Property/ Module Property?	Description
Fetch Min Bytes	Yes	The minimum amount of data that the server would send on receiving a fetch request. The default setting of 1 byte means that fetch requests are answered as soon as a single byte of data is available or the fetch request times out waiting for data to arrive.
Fetch Max Wait	Yes	The maximum amount of time that the server would block before answering a fetch request if there is not sufficient data to immediately satisfy the requirement given by <code>fetch.min.bytes</code> .  The default value is 500.
Heartbeat Interval	Yes	Time in milliseconds between heartbeats to the consumer. Heartbeats are used to make sure that the consumer's session stays active and to facilitate rebalancing and information when consumers join or leave a group.  The default value is 3000.
Session Timeout	Yes	The consumer sends periodic heartbeats to server indicating about its liveness to the broker. If no heartbeats are received by a broker before the expiration of this session timeout, the broker removes this consumer from the group and initiates a rebalance.  The default value is 30000.

### Description


In the **Description** tab, you can enter a short description for the activity.

Field	Literal Value/ Process Property/ Module Property?	Description
Description	None	A description of the activity.

### Advanced

The following table describes the **Advanced** tab of the ReceiveMessage activity.

Field	Literal Value/ Process Property/ Module Property?	Description
Sequence Key	None	XPath expression that specifies which processes must run in sequence. Process instances with sequencing keys that evaluate to the same value are executed sequentially in the sequence the process instance was created.

Field	Literal Value/ Process Property/ Module Property?	Description
Custom Job Id	None	This field can contain an XPath expression that specifies a custom ID for the process instance.
Enable Auto Commit	None	<ul style="list-style-type: none"> <li>Select this check box to auto commit the message record.</li> <li>Deselect the check box to enable Manual Commit for the message record.</li> </ul>  Use the Confirm activity from the <b>General Activities</b> palette for manual commit.
AutoCommit Interval	Yes	Interval in milliseconds in which the consumer offsets are auto-committed to Kafka when the auto-commit mode is enabled.  The default value is 5000.
AutoOffset Reset	Yes	This is required and selected when there is no initial offset or the offset is out of range. <ul style="list-style-type: none"> <li>Earliest: resets the offset to the earliest offset.</li> <li>Latest: resets the offset to the latest offset.</li> <li>None: throws exception to the consumer.</li> </ul>
Properties	None	The properties name and value.



Kafka plug-in supports [Fault Tolerance](#). In case of any errors, you can use **Properties** to add the correct name and value.

## Conversation

You can initiate the conversation here. Click the **Add New Conversation**  button to initiate multiple conversations.

## Output

The following table describes the fields in the **Output** tab.

Field	Type	Description
KafkaReceiverMessageOutput	complex	The complete output for the ReceiveMessage activity.
topic	string	The topic name.
Partition	number	The sequence ID of the partition.

Field	Type	Description
offset	number	The sequence ID assigned to each record within the partition.
Key	String	Specified key of the incoming record.
Message	String	Message received through Kafka.

# Sample Project Overview

The sample projects help to understand how TIBCO ActiveMatrix BusinessWorks™ Plug-in for Apache Kafka operates.

TIBCO ActiveMatrix BusinessWorks™ Plug-in for Apache Kafka is packaged with sample projects. After installing the plug-in, the following sample projects can be found in the `TIBCO_HOME\bw\palettes\kafka\version_number\samples` directory.

- [Kafka\\_GeneralSample](#)
- [Kafka\\_CloudSample](#)

## Working with Kafka\_GeneralSample Project

### Prerequisites

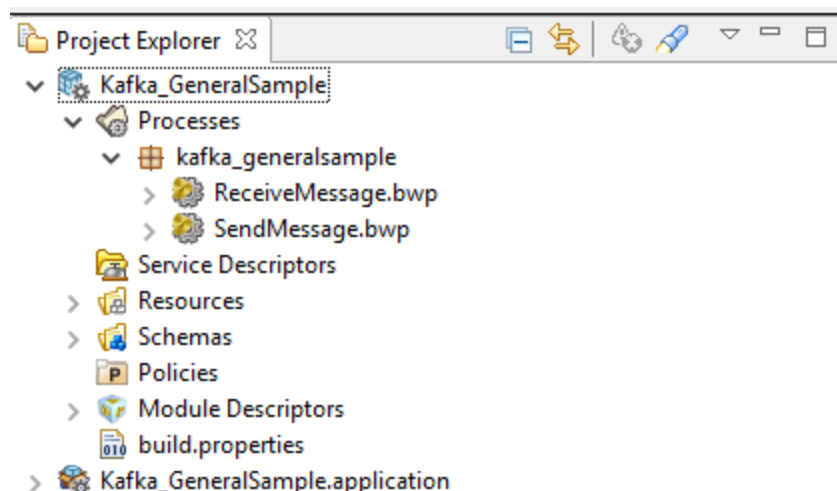
Before running the project, you must import the sample project to TIBCO Business Studio.

### Procedure

1. Start TIBCO Business Studio using one of the following ways:
  - For Microsoft Windows, click **Start > All Programs > TIBCO > TIBCO\_HOME > TIBCO Business Studio version\_number > Studio for Designers**.
  - For Linux, run the TIBCO Business Studio executable file located in the `TIBCO_HOME\studio\version_number\eclipse` directory.
2. From the menu, click **File > Import**.
3. In the Import window, expand the **General** folder and select the **Existing Studio Projects into Workspace** item. Click **Next**.
4. Click **Browse** next to the **Select archive file** field to select the `Kafka_GeneralSample.zip` file. Click **Finish**. The `Kafka_GeneralSample.zip` file is in the `TIBCO_HOME\bw\palettes\kafka\version\samples` directory.

### Result


The sample project is imported to TIBCO Business Studio.



## Configuring Kafka Connection for Kafka\_GeneralSample

Configuring Kafka shared connection resource is required to configure a connection with the Kafka server.

### Procedure

1. In the Project Explorer view, expand **Kafka\_GeneralSample**.
2. In the Resources folder, double click **KafkaConnectionResource.kafkaconnectionResource**.
3. In the KafkaConnectionResource Editor, configure each field accordingly.
4. On the toolbar, click the **Save**  icon to save your changes.

For more information, refer [Creating a Kafka Connection](#).

## Configuration of the Kafka\_GeneralSample Processes

The sample project contains two processes. After [Importing the Sample Project](#), expand the Processes resource to display the processes.

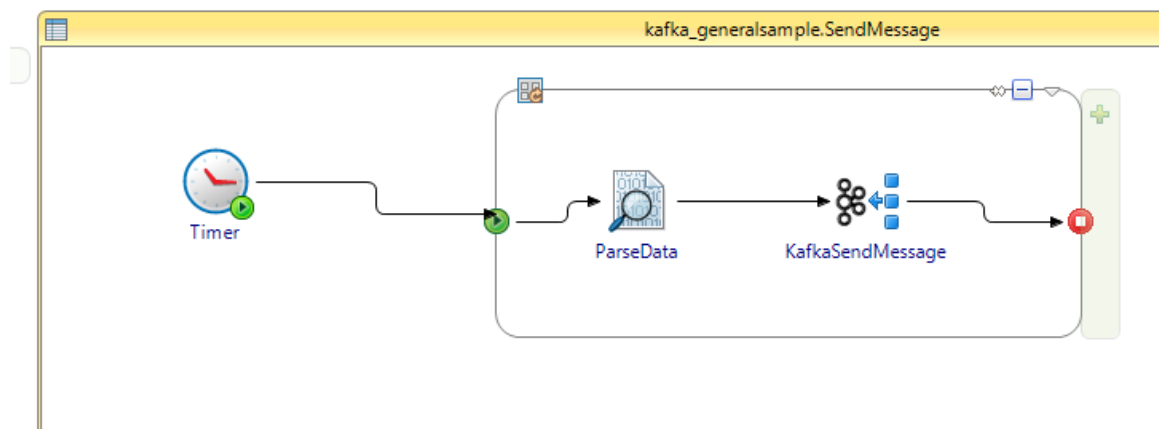
### Kafka\_GeneralSample

This sample project contains the following two processes:

- SendMessage.bwp
- ReceiveMessage.bwp

### SendMessage.bwp

This process demonstrates how to use the plug-ins to send messages through producer using Kafka server. ParseData activity is used to read messages one at a time from the text file so that each line is sent as a separate message.

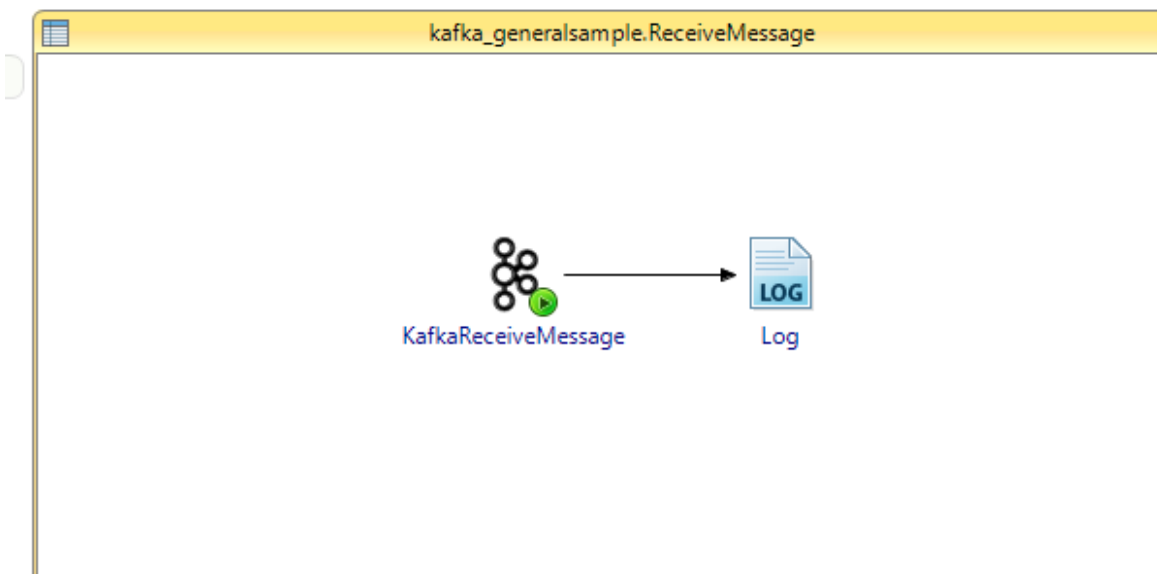


**KafkaSendMessage (KafkaSendMessage)**

<b>General</b>	<b>Name:</b>	KafkaSendMessage
<b>Description</b>	<b>Kafka Connection :</b>	KafkaProperty kafka_generalsample.KafkaConnectionResource
<b>Advanced</b>	<b>Topic Name :</b>	test
<b>Input</b>	<b>Assign Custom Partition :</b>	<input type="checkbox"/>
<b>Output</b>	<b>Key Serializer :</b>	String
<b>Fault</b>	<b>Value Serializer :</b>	String
	<b>Acks :</b>	1
	<b>Buffer Memory :</b>	33554432
	<b>Compression Type :</b>	none
	<b>Retries :</b>	0

## ReceiveMessage.bwp

This process contains a ReceiveMessage process starter. The ReceiveMessage activity listens for an incoming message event and starts the job execution on receiving the incoming records and writes them to a log.



Activity configuration for the ReceiveMessage.bwp:

**KafkaReceiveMessage (KafkaReceiveMessage)**

<b>General</b>	<b>Name:</b>	KafkaReceiveMessage
<b>Description</b>	<b>Kafka Connection :</b>	KafkaProperty kafka_generalsample.KafkaConnectionResource
<b>Advanced</b>	<b>Group Id :</b>	test-consumer-group
<b>Conversations</b>	<b>Topic Names :</b>	test
<b>Output</b>	<b>Assign Custom Partition :</b>	<input type="checkbox"/>
	<b>Key Deserializer :</b>	String
	<b>Value Deserializer :</b>	String
	<b>Fetch Timeout :</b>	1000
	<b>Fetch Min Bytes :</b>	1
	<b>Fetch Max Wait(msec) :</b>	500
	<b>Heartbeat Interval(msec) :</b>	3000
	<b>Session Timeout(msec) :</b>	30000

## Running the Kafka\_GeneralSample Project



The sample project shows how to use the plug-in to send and receive messages using Apache Kafka.



## Prerequisites

Import the sample project to TIBCO Business Studio, as described in [Working with Kafka\\_GeneralSample Project](#), and configure Kafka shared resource connection, as described in [Creating a Kafka Connection](#).

## Procedure

1. In the Project Explorer view, expand the **Module Descriptors** resource and double click **Components**.
2. By default, all the processes are listed in the Components editor.
3. On the toolbar, click the **Save**  icon to save your changes.
4. From the menu, click **Run > Run Configurations** to run the selected process.
5. In the Run Configuration dialog, expand **BusinessWorks Application**, and then click **BWApplication**.
6. In the right panel, click the **Applications** tab, and select the check box next to **Kafka\_GeneralSample.application**.
7. Click **Run** to run the selected process.
8. Click the **Terminate**  icon to stop the process.

## Working with Kafka\_CloudSample Project

### Prerequisites

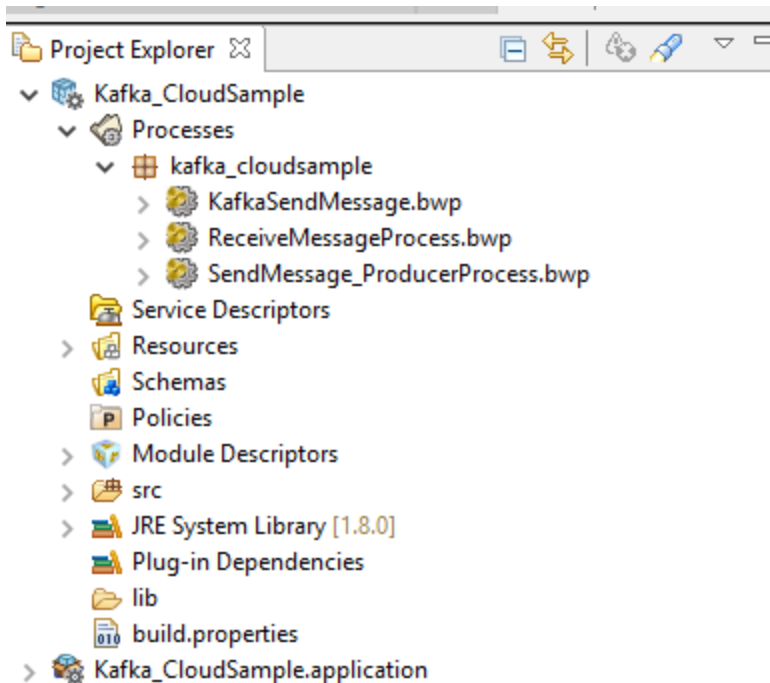
Before running the project, you must import the sample project to TIBCO Business Studio.

### Procedure

1. Start TIBCO Business Studio using one of the following ways:
  - For Microsoft Windows, click **Start > All Programs > TIBCO > TIBCO\_HOME > TIBCO Business Studio version\_number > Studio for Designers**.
  - For Linux, run the TIBCO Business Studio executable file located in the `TIBCO_HOME\studio\version_number\eclipse` directory.
2. From the menu, click **File > Import**.
3. In the Import window, expand the **General** folder and select the **Existing Studio Projects into Workspace** item. Click **Next**.
4. Click **Browse** next to the **Select archive file** field to select the `Kafka_CloudSample.zip` file. Click **Finish**. The `Kafka_CloudSample.zip` file is in the `TIBCO_HOME\bw\palettes\kafka\version\samples` directory.

### Result


The sample project is imported to TIBCO Business Studio.



## Configuring Kafka Connection for Kafka\_CloudSample

Configuring Kafka shared connection resource is required to configure a connection with the Kafka server.

### Procedure

1. In the Project Explorer view, expand **Kafka\_CloudSample**.
2. In the Resources folder, double click **KafkaConnectionResource.kafkaconnectionResource**.
3. In the KafkaConnectionResource Editor, configure each field accordingly.
4. On the toolbar, click the **Save**  icon to save your changes.

For more information, refer [Creating a Kafka Connection](#).

## Configuration of the Kafka\_CloudSample Processes

The sample project contains three processes. Each process has a different function. After [Importing the Sample Project](#), expand the Processes resource to display the processes.

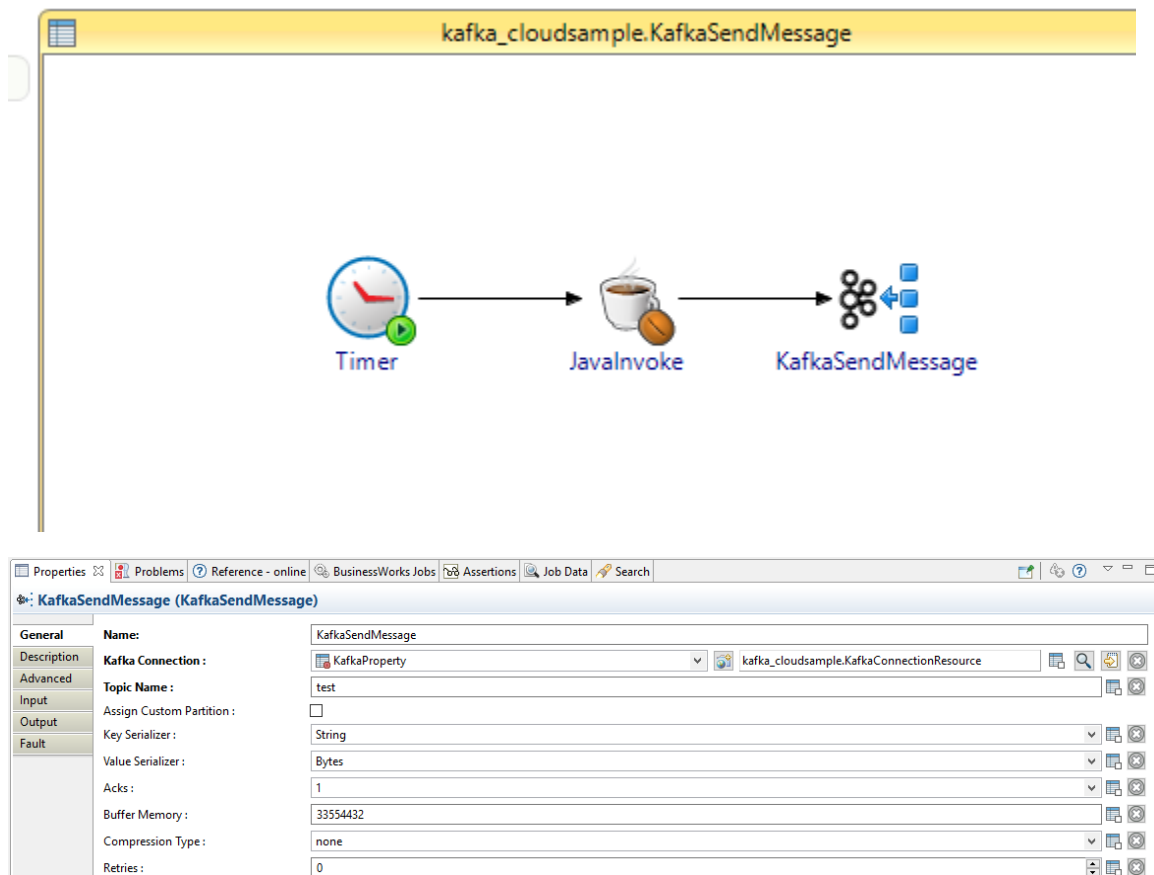
### Kafka\_CloudSample

This sample project contains the following three processes:

- KafkaSendMessage.bwp
- SendMessage\_ProducerProcess.bwp
- ReceiveMessageProcess.bwp

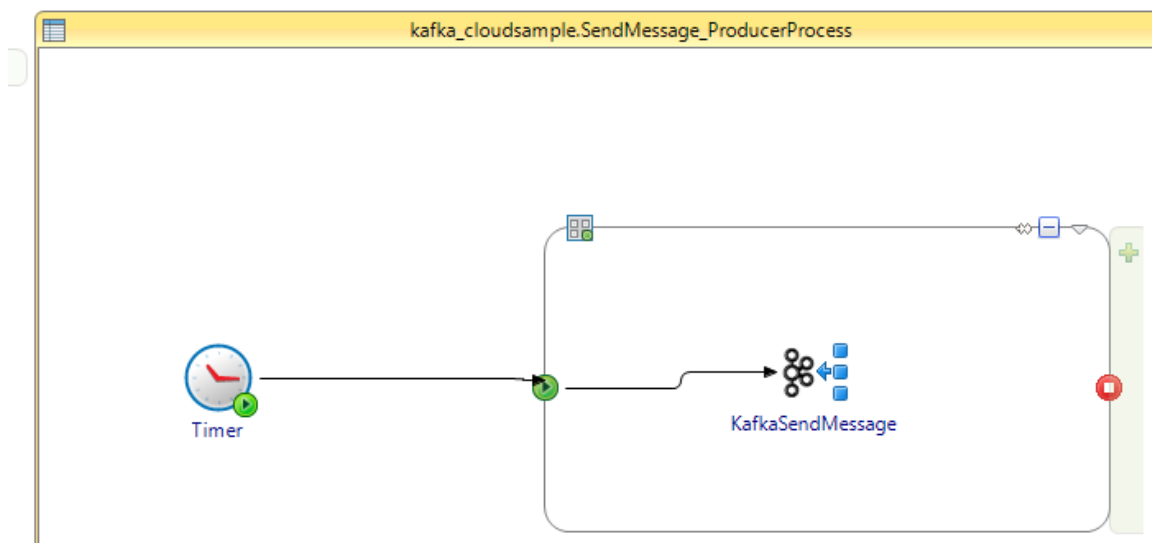
### KafkaSendMessage.bwp

This process demonstrates how to use the plugins to send messages through producer using Kafka server. JavaInvoke activity is used to invoke Java class method. It returns message, which will be read by SendMessage activity.



### SendMessage\_ProducerProcess.bwp

This process demonstrates how to use the plugins to send messages through producer using Kafka server. This process uses "While" loop to send message multiple times.

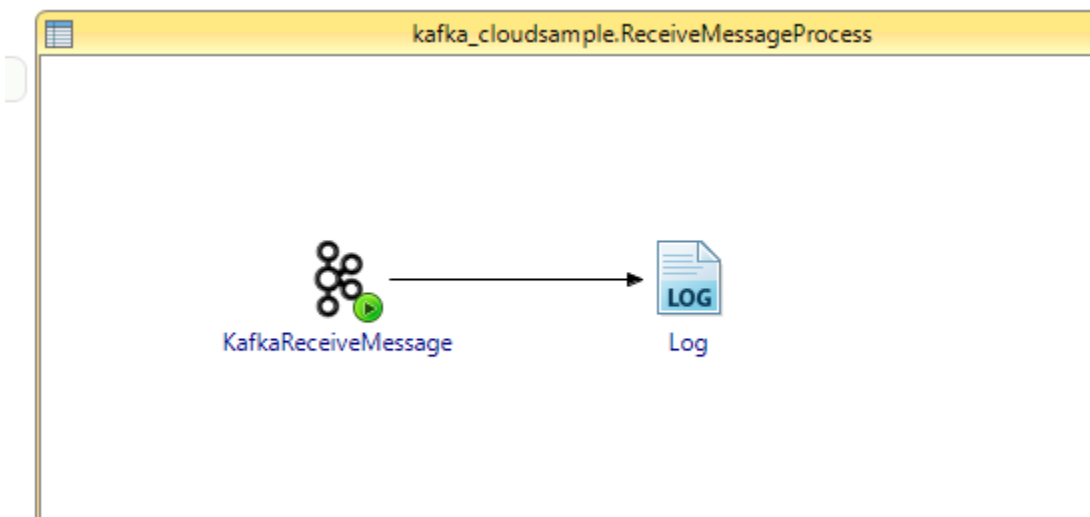


**KafkaSendMessage (KafkaSendMessage)**

<b>General</b>	<b>Name:</b>	KafkaSendMessage
<b>Description</b>	<b>Kafka Connection :</b>	KafkaProperty kafka_cloudsample.KafkaConnectionResource
<b>Advanced</b>	<b>Topic Name :</b>	test
<b>Input</b>	<b>Assign Custom Partition :</b>	<input type="checkbox"/>
<b>Output</b>	<b>Key Serializer :</b>	String
<b>Fault</b>	<b>Value Serializer :</b>	String
	<b>Acks :</b>	1
	<b>Buffer Memory :</b>	33554432
	<b>Compression Type :</b>	none
	<b>Retries :</b>	0

## ReceiveMessageProcess.bwp

This process contains a ReceiveMessage process starter. ReceiveMessage activity listens for incoming message event and starts the job execution on receiving the incoming records and write them to a log.



Activity configuration for the ReceiveMessage.bwp:

**KafkaReceiveMessage (KafkaReceiveMessage)**

<b>General</b>	<b>Name:</b>	KafkaReceiveMessage
<b>Description</b>	<b>Kafka Connection :</b>	KafkaProperty kafka_cloudsample.KafkaConnectionResource
<b>Advanced</b>	<b>Group Id :</b>	test-consumer-group
<b>Conversations</b>	<b>Topic Names :</b>	test
<b>Output</b>	<b>Assign Custom Partition :</b>	<input type="checkbox"/>
	<b>Key Deserializer :</b>	String
	<b>Value Deserializer :</b>	String
	<b>Fetch Timeout :</b>	1000
	<b>Fetch Min Bytes :</b>	1
	<b>Fetch Max Wait(msec) :</b>	500
	<b>Heartbeat Interval(msec) :</b>	3000
	<b>Session Timeout(msec) :</b>	30000



## Running the Kafka\_CloudSample Project

The sample project shows how to use the plug-in to send and receive messages using Apache Kafka.

### Prerequisites

Import the sample project to TIBCO Business Studio, as described in [Working with Kafka\\_CloudSample Project](#), and configure Kafka shared resource connection, as described in [Creating a Kafka Connection](#).

## Procedure

1. In the Project Explorer view, expand the **Module Descriptors** resource and double click **Components**.
2. By default, all the processes are listed in the Components editor.
3. On the toolbar, click the **Save**  icon to save your changes.
4. From the menu, click **Run > Run Configurations** to run the selected process.
5. In the Run Configuration dialog, expand **BusinessWorks Application**, and then click **BWApplication**.
6. In the right panel, click the **Applications** tab, and select the check box next to **Kafka\_CloudSample.application**.
7. Click **Run** to run the selected process.
8. Click the Terminate  icon to stop the process.

# Log Management

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

By default, error logs are displayed in the Console view when you run a process in the 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.

Trace	Includes all information regarding the running process.
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

You can configure different log levels for the plug-in and plug-in activities to trace different messages. By default, the plug-in uses the log level configured for TIBCO ActiveMatrix BusinessWorks. The default log level of TIBCO ActiveMatrix BusinessWorks 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 logger area to specify a log level for the plug-in:

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

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



If you set the log level to Debug, the input and output for the plug-in activities are also displayed in the Console view. See [Log Levels](#) for more details regarding each log level.

3. Optional. Add one of the following nodes in the BusinessWorks Palette and Activity loggers' area to control a log level for the activity.
  - For example, to control the debug log level for the Kafka Sendmessage activity, set the following parameters:

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

- For example, to control the debug log level for the Kafka ReceiveMessage activity, set the following parameters:

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

- For example, to control the debug log level for the Kafka Shared resource, set the following parameters:

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



The activities that are not configured with specific log levels use the log level configured for the plug-in.

- Save the file.

## Exporting Logs to a File

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

### Procedure

- In Windows system, navigate to `TIBCO_HOME\bw\version_number\config\design\logback` directory and open the `logback.xml` file.



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.

- 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-kafka.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.

- Add the following node to the root node at the end of the `logback.xml` file:

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

- Save the file.

## Error Codes

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

Error Code and Error Message	Role	Category	Description	Solution
BW-KAFKA-100001 PARAMETER_NOT_SPECIFIED.	errorRole	BW-Plugin	Occurs when a parameter is not specified while running activity. Message will also contain parameter name.	Provide required parameter.
BW-KAFKA-100002 SPECIAL_CHARACTER_NOT_ALLOWED.	errorRole	BW-Plugin	Occurs when special character is used for group ID.	Remove special character from group ID.
ERROR_OCCURED_RETRIEVE_RESULT.errorCode=500002  IOException occurred while retrieving XML Output for activity [{0}].	errorRole	BW-Plugin	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-Plugin	Occurs when an activity invokes an execute method.	None.
SEND_PRODUCER_CREATED_EXCEPTION.errorCode=500004  Exception occurred while create producer.{0}	errorRole	BW-Plugin	Occurs when producer creation fails.	Check if Kafka configuration is correct and it is running.
SEND_MESSAGE_EXCEPTION.errorCode=500005  Exception occurred while send message to broker. {0}.	errorRole	BW-Plugin	Occurs when sending message to broker fails.	Check if broker is running.



Error Code and Error Message	Role	Category	Description	Solution
RECEIVED_CONSUMER_CREATED_EXCEPTION.errorCode=500007  Exception occurred while create consumer.{0}	errorRole	BW-Plugin	Occurs when Consumer creation fails.	Check if Kafka configuration is correct and it is running.
ERROR_OCCURED_RECEIVE_MESSAGE.errorCode=500008  Exception occurred while receive message from broker.{0}.	errorRole	BW-Plugin	Occurs when receiving message from broker fails.	Check if broker is running.
ERROR_KAFKA_CONNECTION_ESTABLISH.errorCode=500009  Exception occurred while establishing the Kafka connection.Please verify the Kafka connection configuration [{0}].	errorRole	BW-Plugin	Occurs when kafka topic metadata is incorrect.	Check Kafka connection configuration for metadata.
SEND_MESSAGE_BUFFER_MEMORY_EXCEPTION.errorCode=500010  Exception occurred while sending message to broker because message is larger than the total memory buffer you have configured with the buffer.memory configuration.	errorRole	BW-Plugin	Occurs when memory buffer configuration is incorrect.	Change buffer memory settings.
SEND_TOPICNAME_EXCEPTION.errorCode=500011  Exception occurred because topic name is illegal, contains a character other than ASCII alphanumerics, '.', '_', '-' and '-'.	errorRole	BW-Plugin	Occurs when topic name contains special character other than ., -, _.	Update topic name in BW.

Error Code and Error Message	Role	Category	Description	Solution
ERROR_SEND_CUSTOM_PARTITION.errorCode=500012  Exception occurred because provided customer partition does not exist on Kafka broker. {0}.	errorRole	BW-Plugin	Occurs when specified partition does not exist.	Check partition for the broker.
ERROR_OCCURED_RECEIVE_MESSAGE_CUSTOM_PARTITION.errorCode=500013  Exception occurred because given partition does not exist on Kafka broker. {0}.	errorRole	BW-Plugin	Occurs when specified partition does not exist.	Check partition for the broker.
ERROR_SEND_MAX_REQUEST_SIZE.errorCode=500014  Exception occurred because message is too large. {0}.	errorRole	BW-Plugin	Occurs when message is larger than allowed size.	Archive message or increase allowed size in server settings.