TIBCO ActiveMatrix BusinessWorks™ Plug-in for Apache Cassandra User's Guide

Software Release 6.3
August 2017

Two-Second Advantage®

TIBCO®

Important Information

Contents

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:

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

The following documents provide additional information and can be found in the TIBCO Documentation Library:

- TIBCO ActiveMatrix BusinessWorks documentation
- TIBCO Enterprise Administrator documentation

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

Plug-in Overview

Apache Cassandra™ is a partitioned row store. It is an open-source, distributed-database system that is designed for storing and managing large amounts of data across commodity servers

Apache Cassandra is a database server that has good scalability and high availability without compromising its performance. For more information about its features and functionality, go to http://cassandra.apache.org/.

TIBCO ActiveMatrix BusinessWorks™ is easy to use integration product suite for enterprise, web, and mobile applications. It uses the Eclipse graphical user interface (GUI), TIBCO Business Studio™, for defining business processes and the process engine to execute them. TIBCO ActiveMatrix BusinessWorks Plug-in for Apache Cassandra plugs into TIBCO ActiveMatrix BusinessWorks. You can use this plug-in to configure a connection to Cassandra server, and use an activity to update and query your data.

The plug-in provides the following main features:

- **Cassandra Connection Shared Resource** - You can use the Cassandra connection shared resource to connect to Cassandra Database. The shared resource is used by the Cassandra plugin activities.
- **Update** - You can use this activity to create, insert, update, batch update, or remove data to and from a Cassandra table or other write operations.
- **Query** - You can use this activity to do query operation.
- **Get Cassandra Connection** - You can use this activity to get Cassandra connection instance.

Getting Started

A typical workflow for using the plug-in to achieve different goals includes creating a process, testing it in the debugger, and deploying the application.

TIBCO ActiveMatrix BusinessWorks uses the Eclipse graphical user interface (GUI) provided by TIBCO Business Studio to define business processes and generate Enterprise Archives (EAR files). The EAR file is deployed and run in the ActiveMatrix BusinessWorks runtime, and also is managed by using TIBCO Enterprise Administrator (TEA).

The typical workflow for using the plug-in is:

Creating a Project

Projects are BusinessWorks application modules that are created in TIBCO Business Studio. A project contains various resources.

Procedure

1. Start TIBCO Business Studio.
2. Click **File** > **New** > **BusinessWorks Resources**.
3. Click **BusinessWorks Application Module** in the BusinessWorks Resource dialog box. Click **Next**.

There are several ways to open the New BusinessWorks Application Module dialog box and create a new project in TIBCO Business Studio. See the TIBCO ActiveMatrix BusinessWorks documentation for more information.

4. Type a name for the project that you are creating in the **Project name** field.
5. Keep the **Use default location**, **Create empty process**, and **Create Application** check boxes selected. Click **Finish**.

Result

A project and an application are created and displayed in the Project Explorer view. The Process editor opens automatically.

Creating a Cassandra Connection Shared Resource

To use the plug-in activities, you must create a Cassandra connection using the Cassandra shared resource. The Cassandra shared resource describes the connection between the plug-in and the Cassandra server.

Procedure

1. Expand the created project in the Project Explorer view.
2. Right-click the **Resources** folder and select **New** > **Cassandra Connection Resource**.

3. Type a name in the **Resource Name** field in the Cassandra Connection Editor dialog. Click **Finish**.
4. Configure the Cassandra Connection shared resource in the displayed editor, as described in Cassandra Connection Shared Resource on page 9.

Configuring a Process

Processes define the business logic. After a project is created, you must configure the process by adding activities, conditions, and services.

Procedure

1. Select an activity from the Palette view and drop it in the Process editor.
   For example, select and drop the Timer activity from the General Activities palette and the Update activity from the Cassandra palette.
2. Click  to create links between the activities and configure the condition types.
3. Configure the added activities, as described in Cassandra Palette on page 11.

   A Cassandra shared resource is required when configuring the activities. See Creating a Cassandra Conneciton Shared Resource for more details on how to create the shared resource.
4. Click **File** > **Save** to save the project.

Debugging and Running a Process

Debug the application you have configured to ensure that the application configuration is correct.

Procedure

1. Open the process that you have configured in TIBCO Business Studio.
2. On the toolbar, click **Run** > **Debug Configurations**.
3. Click **BusinessWorks Application** > **BWApplication** in the left panel.
4. Ensure only the application you want to debug and run is selected in the **Applications** tab in the right panel.
5. Click the **Advanced** tab and click **Browse** to locate the `logback` file.
   By default, the log file resides in the *`TIBCO_HOME`*`/bw/<version>/config/design/logback` directory and error logs are captured. See Managing Logs on page 24 for more details.
6. Click **Debug**.
   TIBCO Business Studio changes to the Debug perspective. Logs are displayed in the Console view.

Checking Output of an Activity

After debugging the application, you can check the output of activities.

Procedure

1. In the Debug perspective, expand **BWApplication** and click the activity in the upper left panel.
2. In the upper right panel, click the **Job Data** view and click **Output**.

Result

The output of the activity is displayed.

You can also check the activity output in the plug-in logs. See Managing Logs on page 24 for more information.

Deploying Applications

After deploying applications, you can manage BusinessWorks applications by using TIBCO Enterprise Administrator.

Prerequisites

The following tasks are required before deploying applications:

- Creating a Project on page 6.
- Generating an EAR File on page 8.

A typical workflow of deployment includes:

1. Upload an EAR file.
2. Deploy an application.
3. Configure an application.
4. Start an application.

You can deploy an application EAR file by using the command-line mode with the **bwadmin** utility. See *TIBCO ActiveMatrix BusinessWorks Administration* for more details about how to deploy an application.

Generating an EAR File

Application archives are enterprise archive (EAR) files that are created in TIBCO Business Studio. An EAR file is required when deploying an application.

Prerequisites

An application project has already been created, as described in Creating a Project on page 6.

There are many ways to generate an EAR file, the following is one such method. For more information on this, see *TIBCO ActiveMatrix BusinessWorks Administration* for more information.

Procedure

1. Go to the File Explorer view and click the **Open Directory to Browse**  icon.
2. Select the folder where you want to generate the EAR file and click **OK**.
   The new folder is displayed in the File Explorer view.
3. Drag the application from the Project Explorer view to the new folder in the File Explorer view.

   The EAR file is generated with the name `<name>.<application>_<version>.ear`.

Cassandra Connection Shared Resource

The Cassandra Connection shared resource describes the Cassandra connection information.

General

The **General** panel contains the following fields:

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.

Cluster Configuration

The **Cluster Configuration** panel contains the following fields:

Field	Module Property?	Description
KeySpace	Yes	The name of the keyspace to be used for the session created.

Node	Yes	The addresses of the nodes to add as contact point.
		User can configure node details in following format:
		`<host>:<port>`
		Multiple node details can be configured separated by a comma as shown below:
		`<host1>:<port1>,<host2>:<port2>`

Advanced Configuration

The **Advanced Configuration** panel contains the following fields:

Field	Module Property?	Description
Connect Timeout(ms)	Yes	Specifies how long in milliseconds the connection remains active if no action is performed. Default value is 5000.
Read Timeout(ms)	Yes	Specifies the per-host read timeout in milliseconds. Default value is 12000.
FetchSize	Yes	Specifies the fetch size to use for SELECT queries. Default value is 5000.

Field	Module Property?	Description
Retry Policy	Yes	The retry policy determines the default behavior to adopt when a request either times out or a node is unavailable. Default value is `DefaultRetryPolicy`.
Logging Retry Policy	Yes	If this check box is selected, the policy is used with Retry Policy.
Load Balancing Policy	Yes	A load balancing policy will determine which node it is to run a query. Default value is `RoundRobinPolicy`.
Token Aware Policy	Yes	If this check box is selected, the policy is used with Load Balancing Policy.

Security Configuration

The **Security Configuration** panel contains the following fields:

Field	Module Property?	Description
Login Credential	No	If this check box is selected, enter the **Username** and **Password** to authenticate the login.
Username	Yes	The user name to access the Cassandra server.
Password	Yes	The password to access the Cassandra server.
Test Connection	No	Click this button to validate the connection.

Confidentiality

The **Confidentiality** panel contains the following fields:

Field	Module Property?	Description
SSL Client Configuration	No	This flag enables or disables transport layer security for the connection.
SSL Client	No	Select an SSL Client shared resource to establish a connection between the plug-in and the specified Cassandra server. This field is available when the **SSL Client Configuration** check box is selected.

Cassandra Palette

Each activity in the TIBCO ActiveMatrix BusinessWorks Plug-in for Apache Cassandra plug-in achieves a different functionality. The following activities are supported:

- Update activity
- Query activity
- GetCassandraConnection activity

You can use one or more activities from the palette in your project. All activities can use the same Cassandra Shared Connection because they work independently.

Update Activity

Update is an asynchronous activity that helps to create, insert, update, batch update, or remove data to and from a Cassandra table or other write operations.

General

The **General** tab contains 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.
Cassandra Connection	Literal Value Module Property	Shared configuration resource containing the Cassandra connection information.
Statement	None	Input the Cassandra Query Language (CQL) statement to perform on the keyspace.
Fetch UserTypes	None	This button fetches the User Defined Data Types (UDT) for the **Prepared Statements**. The UDT fetch is successful if the following requirements are met: • Cassandra Shared Resource connection is ready and is provided in the **Cassandra Connection** field in the activity. • The UDTs are already defined in the keyspace. For every new activity, you must fetch the UDT again. If you make changes to the UDT in the back end, ensure to fetch it again to see the latest types.

Field	Literal Value/ Process Property/ Module Property?	Description
Use Nested Collections	None	Select this check box if you are using nested collections in the **Prepared Statements**. If the check box is selected, enter the table name used in the **Statement** above. **Fetch Columns** fetches the nested columns in the `Data Type/ Nested Collection` drop-down of the **Prepared Statements**.  See the following examples for using nested collections: • `List<list<datatype>>` - You can use nested collection column of type List of a data type, nested within List. • `Set<Set<datatype>>` - You can use nested collection column of type Set of a data type, nested within Set. • `Map<text, List<datatype>>` - You can use nested collection column of type Map with key as a data type and value as List of a data type. Similarly, you can have other combinations using List, Set and Map.

Field	Literal Value/ Process Property/ Module Property?	Description
Prepared Statements	None	The **Prepared Statements** contains the `Parameter Name`, `Parameter Cardinality` and `Data Type/Nested Collection` fields. All the parameters defined in the Prepared Statements table are available in the **Input** tab of the activity for the user to provide values. Each parameter corresponds to the question mark in the same position in the CQL statement. That is, the first parameter in the list corresponds to the first question mark, the second parameter in the list corresponds to the second question mark, and so on. The warning is only for the purpose of information. Ensure that the parameters in this field correspond correctly to the statement. You can optionally specify names for each parameter. By default, the parameters are named Parameter, Parameter1, and so on. You must supply the data type of each parameter to the CQL statement, and this data type is used in the input schema for the statement With the `Parameter Cardinality` drop-down you can map a data type of type Collection. The Collection data type is one of the following: • Map • Set • List  When you select `Map` as the cardinality of a parameter: • You must provide the **Key Type** and **Value Type** for its `Data Type/Nested Collection`. • In case of nested collections, select the `Data Type/ Nested Collection` of the parameter before you select its cardinality.

Description

Provide a short description for the activity.

Advanced

This tab has the following fields:

Field	Literal Value/ Process Property/ Module Property?	Description
Write Consistency	All	Refers to how up-to-date and synchronized a row of Cassandra data is on all of its replicas.
Tracing	None	The check box for turning the tracing for activity on or off. Checking this box turns the tracing on, which helps you understand Cassandra's internal operations and troubleshoot performance related issues.

Input

This tab has the following fields:

Input Item	Data Type	Mandatory?	Description
Statement	String	No	Input the CQL statement to perform in the KeySpace.  This field takes precedence over the statement field in **General** tab.
Server Time Zone	String	No	The **ServerTimeZone** is used for the timestamp data type. If you use **Prepared Statements** to insert timestamp data, map it to ServerTimeZone. It is the database server machine time zone, for example: America/New_York

Output

The following is the output for the activity:

Output Item	Data Type	Description
ExecutionInfo		The information for Update activity executed.
QueriedHost	String	Return the Cassandra host that coordinated this execution.
Applied	Boolean	If the query that produced this ResultSet was a conditional update, return whether it was successfully applied.

Output Item	Data Type	Description
TraceID	String	Return the operation message data ID that is saved in the database. You can use this ID to query the database. For example: SELECT * FROM system_traces.sessions WHERE session_id =?  This field is displayed only when the **Tracing** check box on the **Advanced** tab is selected.

Fault

The **Fault** tab lists exceptions that are thrown by this activity:

Error Schema Element	Data Type	Description
CassandraPluginException		Any exception thrown by the activity after execution. It has a **msg** and **msgCode** field for the exception.
DriverException		Any exception thrown by Cassandra driver. It has a **msg** and **msgCode** field for the exception.
msg	string	The error message returned by the plug-in.
msgCode	string	The error code returned by the plug-in.

Query

Query is an asynchronous activity that helps query data from a Cassandra table.

General

The **General** tab contains 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.
Cassandra Connection	Literal Value Module Property	Shared configuration resource containing the Cassandra connection information.
Statement	None	Input the CQL statement to perform on the KeySpace.

Field	Literal Value/ Process Property/ Module Property?	Description
Fetch Output Schema	None	When you first configure the Query activity, click **Fetch Output Schema** to populate the schema/columns of the table in the **Query Output**. After you have configured your activity, this button is useful when you make a change in the table. **Fetch Output Schema** synchronizes with the table and changes the output schema, if necessary. Ensure that the table used in the **Statement** is defined in the keyspace.
Fetch UserTypes	None	This button fetches the User Defined Data Types (UDT) for the **Prepared Statements** and **Query Output**.  The UDT fetch is successful if the following requirements are met: • Cassandra Shared Resource connection is ready and is provided in the **Cassandra Connection** field in the activity. • The UDTs are already defined in the keyspace. For every new activity, you must fetch the UDT again. If you make changes to the UDT in the back end, ensure to fetch it again to see the latest types.
Use Nested Collections	None	Select this check box if you are using nested collections in the **Prepared Statements** and **Query Output**. If the check box is selected, enter the table name used in the **Statement** above. **Fetch Columns** fetches the nested columns in the `Data Type/ Nested Collection` drop-down of the **Prepared Statements**.  See the following examples for using nested collections: • `List<list<datatype>>` - You can use nested collection column of type List of a data type, nested within List. • `Set<Set<datatype>>` - You can use nested collection column of type Set of a data type, nested within Set. • `Map<text, List<datatype>>` - You can use nested collection column of type Map with key as a data type and value as List of a data type. Similarly, you can have other combinations using List, Set and Map.

Field	Literal Value/ Process Property/ Module Property?	Description
Prepared Statements	None	The **Prepared Statements** contains the `Parameter Name`, `Parameter Cardinality` and `Data Type/Nested Collection` fields. All the parameters defined in **Prepared Statements** are available in the **Input** tab of the activity for the user to provide values. Each parameter corresponds to the question mark in the same position in the CQL statement. That is, the first parameter in the list corresponds to the first question mark, the second parameter in the list corresponds to the second question mark, and so on. The warning is only for the purpose of information. Ensure that the parameters in this field correctly correspond to the statement. You can optionally specify names for each parameter. By default, the parameters are named Parameter, Parameter1, and so on. You must supply the data type of each parameter to the CQL statement, and this data type is used in the input schema for the statement
Query Output	None	The **Query Output** contains the `Parameter Name`, `Parameter Cardinality` and `Data Type/Nested Collection` fields. All the parameters defined in the Query Output table are shown in the **Output** tab of the activity for display the query result. Each Query Output corresponds to field name in the table. The warning is only for information. Ensure that the parameters in this field correctly correspond to the table fields. For the query output, use the parameter name that corresponds to the table field. You must supply the data type of each parameter that corresponds to the table field. This data type is used in the output schema for the query result. With the `Parameter Cardinality` drop-down you can map a data type of type Collection. The Collection data type is one of the following: • Map • Set • List  When you select `Map` as the cardinality of a parameter: • You must provide the **Key Type** and **Value Type** for its `Data Type/Nested Collection`. • In case of nested collections, select the `Data Type/ Nested Collection` of the parameter before you select its cardinality.

Description

Provide a short description for the activity.

Advanced

This tab has the following fields:

Field	Literal Value/ Process Property/ Module Property?	Description
Read Consistency	All	Refers to how up-to-date and synchronized a row of Cassandra data is on all of its replicas.
Tracing	None	The check box for turning the tracing for activity on or off. Checking this box turns the tracing on, which helps you understand Cassandra's internal operations and troubleshoot performance related issues.

Input

This tab the following fields:

Input Item	Data Type	Mandatory?	Description
Statement	String	No	Input the CQL statement to perform in the KeySpace.  This field takes precedence over the statement field in **General** tab.
Fetch Size	Integer	No	The size of data that gets queried after executing the activity once.  This field has a higher priority than it has in the Cassandra shared resource connection.
ReadSize	Integer	No	The size of data that you want to retrieve from a table by executing this activity. For example, if ReadSize = 100 and FetchSize = 20, it creates a group for the activity which executes 5 times to get the data. The default value of **ReadSize** is `Integer.MaxValue`.

Input Item	Data Type	Mandatory?	Description
Server Time Zone	String	No	The **ServerTimeZone** is used for the timestamp data type. If you use **Prepared Statements** to insert timestamp data, map it to **ServerTimeZone**. It is the database server machine time zone, for example: America/New_York .  For the timestamp data type, when you insert a timestamp data to the database, Cassandra converts the data with database server machine time zone. When you query the data, Cassandra converts the data according to the plug-in machine time zone. So, if the server machine and plug-in machine time zone configuration is not same, the timestamp data display differs from the database plug-in output. When you use **Prepared Statements** to insert timestamp data, be sure to input it in **ServerTimeZone** in the **Input** tab.  It is best that you set the same time zone for database machine and plug-in machine.

Output

This tab has the following fields:

Output Item	Data Type	Description
ResultSetInfo		The information for Query activity executed.
QueriedHost	String	Return the Cassandra host that coordinated this execution.
NoMore	Boolean	Whether there is more data in the table after the query completes. The default has no more data that matches the query condition.
Rows		The rows value that is returned from the query.
Row	String	Each row value data.  If you are using nested collections, for the nested column schema to reflect in the **Row**, ensure that you enter the table name and fetch the columns in the **General** tab.
TraceID	String	Return the operation message data ID that is saved in the database. You can use this ID to query the database. For example: SELECT * FROM system_traces.sessions WHERE session_id =?  This field is displayed only when the **Tracing** check box on the **Advanced** tab is selected.

Fault

The **Fault** tab lists exceptions that are thrown by this activity:

Error Schema Element	Data Type	Description
CassandraPluginException		Any exception thrown by the activity after execution. It has a **msg** and **msgCode** field for the exception.
DriverException		Any exception thrown by the Cassandra driver. It has a **msg** and **msgCode** field for the exception.
msg	string	The error message returned by the plug-in.
msgCode	string	The error code returned by the plug-in.

GetCassandraConnection Activity

The GetCassandraConnection activity is an asynchronous activity that gets a Cassandra connection instance.

General

The **General** tab contains 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.
Cassandra Connection	Literal Value Module Property	Shared configuration resource containing the Cassandra connection information.

Description

Provide a short description for the activity.

Output

This tab has the following fields:

Output Item	Data Type	Description
CassandraConnectionAccessor	Object	Cassandra connection instance. You can get the connection and perform some operation by writing code in Java invoke activity.

Fault

The **Fault** tab lists exceptions that are thrown by this activity.

Error Schema Element	Datatype	Description
CassandraPluginException		Any exception thrown by the activity after execution. It has a **msg** and **msgCode** field for the exception.
DriverExeption		Any exception thrown by Cassandra driver. It has a **msg** and **msgCode** field for the exception.
msg	string	The error message returned by the plug-in.
msgCode	string	The error code returned by the plug-in.

Sample

TIBCO ActiveMatrix BusinessWorks Plug-in for Apache Cassandra packages a sample project within the installer. The sample project is located in the *TIBCO_HOME*\bw\palettes\cassandra\6.3\samples directory.

Working with Sample Project

This Sample project gives you a quick overview of how to use TIBCO ActiveMatrix BusinessWorks Plug-in for Apache Cassandra.

Prerequisites

Navigate to *TIBCO_HOME*\bw\palettes\cassandra\6.3\samples directory and extract the SampleProject.zip file that contains schema.cql. Run the schema.cql script on the Cassandra cqlsh to create a schema for the project activities.

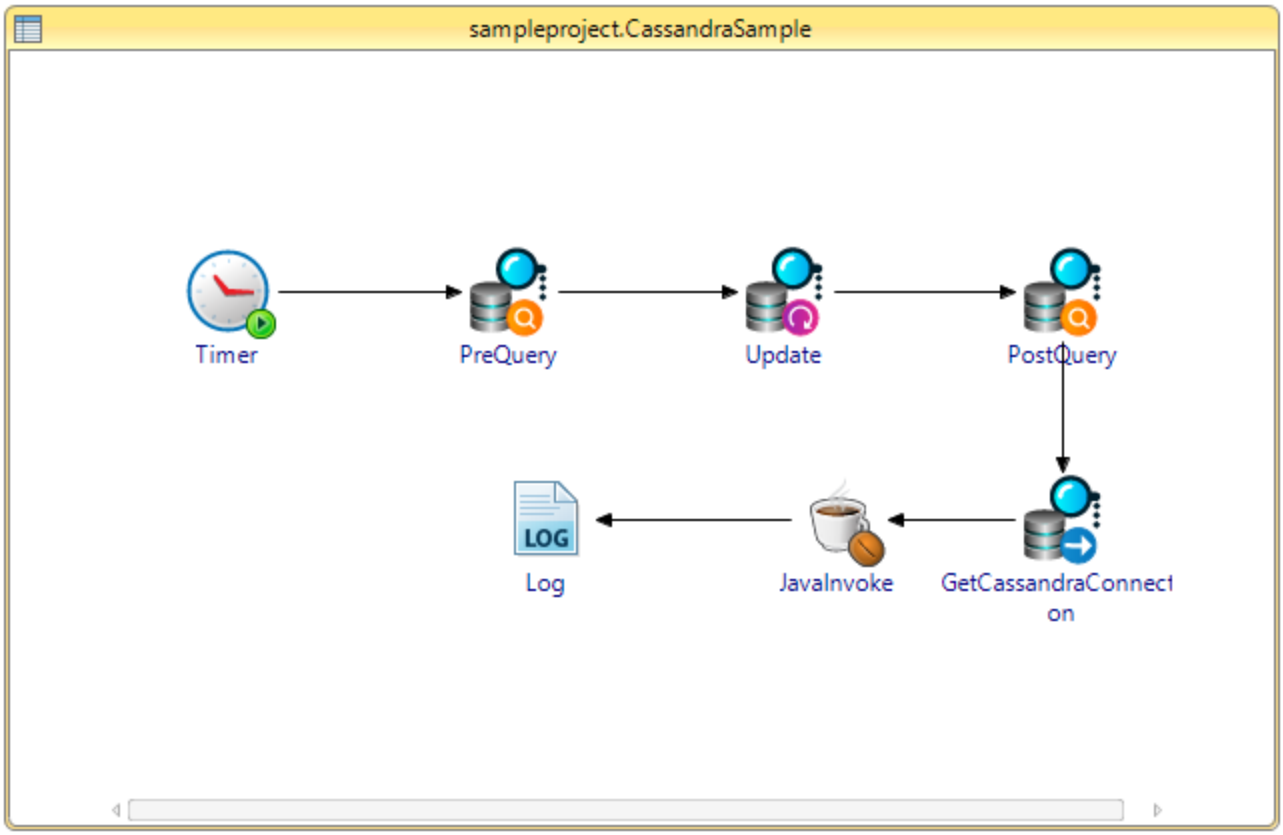
Procedure

1. Start TIBCO Business Studio.
2. 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 locate the samples. Click **Finish**.
   The sample project is located in the *TIBCO_HOME*\bw\palettes\cassandra\6.3\samples directory.
   In this project, the CassandraSample Process is created. This process is used to show the general configurations of Cassandra activities, such as Update, Query, and GetCasandraConnection.
5. In the Module Properties of the SampleProject, enter the keyspace and cassandraNodes values. The cassandraNodes value is the *host name* and *port number* of your Cassandra database server.
6. (Optional) In the Confidentiality panel of the Cassandra Connection Resource, configure the SSL Client.
7. Test the connection and run the **CassandraSample** process in the project.

CassandraSample Process

The CassandraSample process is used to show the general configurations of Cassandra activities.

The process is designed with the following activities:



The following table lists and explains the activities in the process:

Activity	Description
PreQuery	Query activity. Retrieves the details of *users* table by firing the **SELECT** command. The columns of *users* table are populated in the **Query Output**.
Update	Update activity. Inserts a record in the table using the **Prepared Statement**. The values are passed through the **Input** tab.
PostQuery	Query activity. Retrieves the details to see the record inserted in Update activity.
GetCassandraConnection	GetCassandraConnection activity. Establishes the connection with the database and gets the database active object. This object is used in the JavaInvoke activity.
JavaInvoke	JavaInvoke activity. Fetches the name of the *keyspace* using the database active object from GetCassandraConnection activity.

Managing Logs

Logs are used to trace and troubleshoot the plug-in exceptions.

When an error occurs, you can check the 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

A `logback.xml` file is located in the *`TIBCO_HOME`*`\bw\`*`<version>`*`\config\design\logback` directory. Update this file to set up a log file and export logs to a file.

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.
Warn	Indicates that an abnormal condition occurred. Processing continues, but special attention from an administrator is recommended.
Trace	Includes all information regarding the running process.

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`.

If neither the plug-in log nor the BusinessWorks log is configured in the `logback.xml` file, the error logs of the plug-in are displayed in the Console view by default. If the plug-in log is not configured, but the BusinessWorks log is configured in the `logback.xml` file, the configuration for the BusinessWorks log is implemented by the plug-in.

Procedure

1. Navigate to the *`TIBCO_HOME`*`\bw\`*`<version>`*`\config\design\logback` directory and open the `logback.xml` file.
2. Add the following node in the Console Appender area to specify the log level for the plug-in:

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

   The `level` tag defines the log level and the value of the level element can be `Trace`, `Info`, `Error` or `Debug`.

When the `level` is set to `Debug`, the input and output for the plug-in activities are also displayed in the Console view.

3. Optional: Add the following node in Console Appender area to specify the log level for an activity:

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

See the following examples:

- If you want to set the log level of the Update activity to `Debug`, you must add the following node:

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

- If you want to set the log level of the Query activity to `Debug`, you must add the following node:

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

- If you want to set the log level of the GetCassandraConnection activity to `Debug`, you must add the following node:

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

For the activities that do not configure with specific log levels, they still inherit log level configured for the plug-in or BusinessWorks.

4. Save the file.

Exporting Logs to a File

Modify the `logback.xml` file to export plug-in logs to a file.

Procedure

1. Navigate to the *TIBCO_HOME*\bw*<version>*\config\design\logback directory and open the `logback.xml` file.

When deploying an application in TIBCO Enterprise Administrator, you must navigate to the *TIBCO_HOME*\bw\domains\defaultdomain\appnodes\defaultappspace\defaultappnode directory to find the `logback.xml` file.

2. Add the following node to specify the file location:

```
<appender name="FILE" class="ch.qos.logback.core.FileAppender">
   <file>c:/bw6-cassandra.log</file>
      <encoder>
        <pattern>%d{HH:mm:ss.SSS} [%thread] %-5level %logger{36}-%msg%n</pattern>
      </encoder>
</appender>
```

The `file` tag defines the location to which the log is exported and the value is the absolute path of the file that is detailed to the file name.

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

```
    <appender-ref ref="FILE" />
```

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

4. Save the file.

Error Codes

The error codes in the **Fault** tab for each activity are listed with corresponding descriptions and resolutions.

Error codes for Cassandra palette:

Error Code and Error Message	Role	Category	Description	Resolution
TIBCO-BW-PALETTE-CASSANDRA-500001	errorRole	BW-Plug-in	Activity initialization error.	Ensure that the activity is configured correctly.
TIBCO-BW-PALETTE-CASSANDRA-500002	errorRole	BW-Plug-in	Exception occurred during activity execution.	Ensure that the activity is configured correctly.
TIBCO-BW-PALETTE-CASSANDRA-500003	errorRole	BW-Plug-in	Exception occurred during the authentication phase while connecting to a node.	Ensure that the connection shared resource is configured correctly.
TIBCO-BW-PALETTE-CASSANDRA-500004	errorRole	BW-Plug-in	Exception occurred due to an unexpected error internally in driver.	Ensure that at least one host is responding successfully.
TIBCO-BW-PALETTE-CASSANDRA-500005	errorRole	BW-Plug-in	Exception occurred because the object or the byte buffer content being processed does not comply with the expected CQL type.	Ensure that content provided complies with the type configured in the schema.
TIBCO-BW-PALETTE-CASSANDRA-500006	errorRole	BW-Plug-in	Exception thrown when a query cannot be performed because no host is available.	Ensure that at least one host is available.
TIBCO-BW-PALETTE-CASSANDRA-500007	errorRole	BW-Plug-in	Exception thrown because a paging state does not match the statement being executed.	Ensure that schema has not changed.
TIBCO-BW-PALETTE-CASSANDRA-500008	errorRole	BW-Plug-in	Exception thrown due to the timeout during a read query.	Ensure proper network resources and connection.

Error Code and Error Message	Role	Category	Description	Resolution
TIBCO-BW-PALETTE-CASSANDRA-500009	errorRole	BW-Plug-in	Exception thrown due to the timeout during a write query.	Ensure proper network resources and connection.
TIBCO-BW-PALETTE-CASSANDRA-500010	errorRole	BW-Plug-in	Exception thrown due to the timeout during a truncation operation.	Ensure proper network resources and connection.
TIBCO-BW-PALETTE-CASSANDRA-500011	errorRole	BW-Plug-in	Exception thrown when the coordinator knows there are not enough replicas alive to perform a query with the requested consistency level.	Ensure that the server has enough replicas or adjust consistency level.
TIBCO-BW-PALETTE-CASSANDRA-500012	errorRole	BW-Plug-in	Exception thrown when a query attempts to create a keyspace or table that already exists.	Rename keyspace or delete existing keyspace with same name.
TIBCO-BW-PALETTE-CASSANDRA-500013	errorRole	BW-Plug-in	Exception thrown which indicates a syntactically correct but invalid query.	Check the validity of the query being executed.
TIBCO-BW-PALETTE-CASSANDRA-500014	errorRole	BW-Plug-in	Exception thrown which indicates a syntax error in a query.	Check the syntax of query being executed.
TIBCO-BW-PALETTE-CASSANDRA-500015	errorRole	BW-Plug-in	Exception thrown which indicates that a query cannot be performed due to the authorization restrictions of the logged user.	Check the authorization restrictions of logged user.
TIBCO-BW-PALETTE-CASSANDRA-500016	errorRole	BW-Plug-in	Exception thrown if a query trace cannot be retrieved.	Check the trace enabled on Cassandra server.
TIBCO-BW-PALETTE-CASSANDRA-500017	errorRole	BW-Plug-in	Exception thrown when a feature is not supported by the native protocol currently in use.	Ensure that the feature being used is supported.

Error Code and Error Message	Role	Category	Description	Resolution
TIBCO-BW-PALETTE-CASSANDRA-500018	errorRole	BW-Plug-in	Exception thrown indicates that a connection has run out of stream IDs.	Check the Cassandra connection settings on server.
TIBCO-BW-PALETTE-CASSANDRA-500019	errorRole	BW-Plug-in	Exception thrown when a data type provided in Prepared Statements mismatches to that provided in Cassandra schema.	Check the data types provided in Prepared Statements.
TIBCO-BW-PALETTE-CASSANDRA-500020	errorRole	BW-Plug-in	Exception thrown indicates that the contacted host reported itself being overloaded.	Ensure that the hosts configured are running and available to take requests.
TIBCO-BW-PALETTE-CASSANDRA-500021	errorRole	BW-Plug-in	Exception thrown when some of the replicas that were contacted by the coordinator replied with an error.	Check the query and execute after some time.
TIBCO-BW-PALETTE-CASSANDRA-500022	errorRole	BW-Plug-in	Exception thrown when some of the replicas that were contacted by the coordinator replied with an error.	Check the query and execute after some time.
TIBCO-BW-PALETTE-CASSANDRA-500023	errorRole	BW-Plug-in	Exception thrown when the contacted host was bootstrapping when it received a read query.	Ensure that all hosts are running and available to execute reads.
TIBCO-BW-PALETTE-CASSANDRA-500024	errorRole	BW-Plug-in	Exception thrown can be raised when the driver is rebuilding its schema metadata, and a user-defined type cannot be completely constructed due to some missing information.	Ensure that the user defined is created and is available for use.

Error Code and Error Message	Role	Category	Description	Resolution
TIBCO-BW-PALETTE-CASSANDRA-500025	errorRole	BW-Plug-in	Exception thrown when the client did not hear back from the server.	Ensure that the server is running and responsive.
TIBCO-BW-PALETTE-CASSANDRA-500026	errorRole	BW-Plug-in	Exception thrown when the node is down.	Ensure that the node is running and responsive.
TIBCO-BW-PALETTE-CASSANDRA-500027	errorRole	BW-Plug-in	Exception thrown when executing a function.	Ensure that the function being executed exists on server and is correct.
TIBCO-BW-PALETTE-CASSANDRA-500028	errorRole	BW-Plug-in	Exception thrown indicates that the query is invalid because of some configuration problem.	Check the configuration options of the query being executed.
TIBCO-BW-PALETTE-CASSANDRA-500029	errorRole	BW-Plug-in	Exception thrown by the Cassandra driver.	Check the Cassandra server.
TIBCO-BW-PALETTE-CASSANDRA-500030	errorRole	BW-Plug-in	Exception thrown because the existing data type does not match the type of item selected in Prepared Statements.	Ensure that all Prepared Statements types are according to the type defined on Cassandra server.
TIBCO-BW-PALETTE-CASSANDRA-500031	errorRole	BW-Plug-in	Exception thrown because statement is not provided or is empty.	Ensure that the statement is provided.
TIBCO-BW-PALETTE-CASSANDRA-500032	errorRole	BW-Plug-in	Exception thrown because statement is invalid.	Ensure that the statement is valid.
TIBCO-BW-PALETTE-CASSANDRA-500033	errorRole	BW-Plug-in	Exception thrown because Prepared Statements are not equal to number of '?'.	Ensure that the Prepared Statements are equal to number of '?'.
TIBCO-BW-PALETTE-CASSANDRA-500034	errorRole	BW-Plug-in	Exception thrown because read size is not integer or less than zero.	Ensure that the read size is integer greater than zero.