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

Software Release 6.0
September 2015

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:

https://docs.tibco.com

Documentation on the TIBCO Documentation 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, please visit https://docs.tibco.com.

Product-Specific Documentation

Documentation for TIBCO products is not bundled with the software. Instead, it is available on the TIBCO Documentation site. To directly access documentation for this product, double-click the following file:

TIBCO_HOME/release_notes/TIB_bwpluginmongodb_*version*_docinfo.html

where *TIBCO_HOME* is the top-level directory in which TIBCO products are installed. On Windows, the default *TIBCO_HOME* is `C:\Program Files\tibco`. On UNIX systems, the default *TIBCO_HOME* is `/opt/tibco`.

The following documents for this product can be found in the TIBCO Documentation Site:

- *TIBCO ActiveMatrix BusinessWorks Plug-in for MongoDB Installation*
- *TIBCO ActiveMatrix BusinessWorks Plug-in for MongoDB User's Guide*
- *TIBCO ActiveMatrix BusinessWorks Plug-in for MongoDB 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 TIBCOmmunity

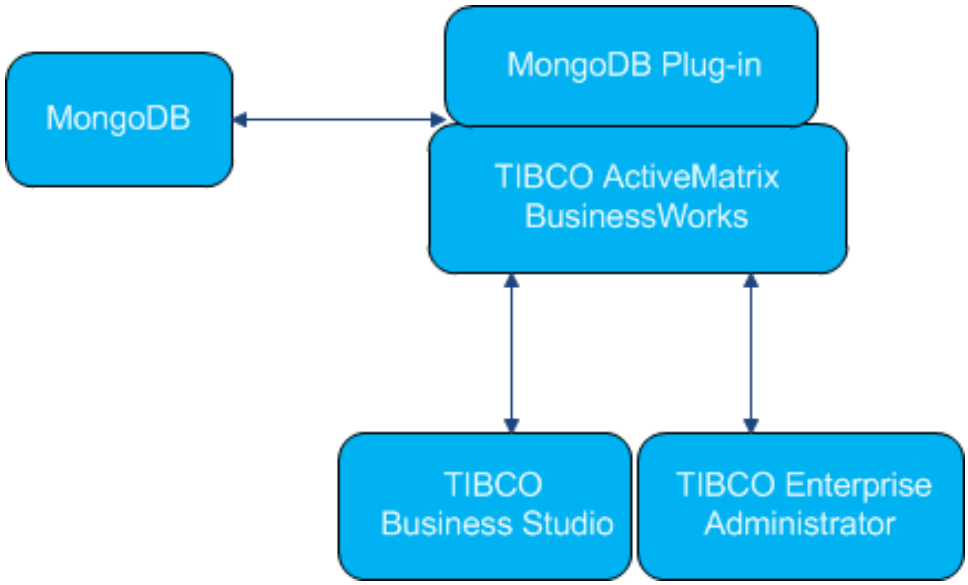
TIBCOmmunity is an online destination for TIBCO customers, partners, and resident experts. It is a place to share and access the collective experience of the TIBCO community. TIBCOmmunity offers forums, blogs, and access to a variety of resources. To register, go to the following web address:

https://www.tibcommunity.com

Product Overview

TIBCO ActiveMatrix BusinessWorks™ Plug-in for MongoDB provides you with a user-friendly environment to manage documents and files in MongoDB.

In addition to inserting, querying, updating, and removing documents or files in MongoDB, you can use the plug-in to wait for and listen to write operations in a collection in MongoDB. You can also use the plug-in to get MongoDB Connection instances, run commands on MongoDB, and perform map-reduce operations in any collection in MongoDB.



The following list describes each item and the relationship between them in this figure.

- MongoDB is an open-source, document-oriented database that provides high performance, high availability, and automatic scaling. For more information about MongoDB features and functionalities, visit MongoDB website at http://docs.mongodb.org/manual/.
- TIBCO ActiveMatrix BusinessWorks Plug-in for MongoDB is plugged into TIBCO ActiveMatrix BusinessWorks™ and is connected to MongoDB.
- 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 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 for the beginners who want to use TIBCO ActiveMatrix BusinessWorks Plug-in for MongoDB in TIBCO Business Studio.

All the operations are performed in TIBCO Business Studio. See TIBCO Business Studio Overview to get familiar with TIBCO Business Studio.

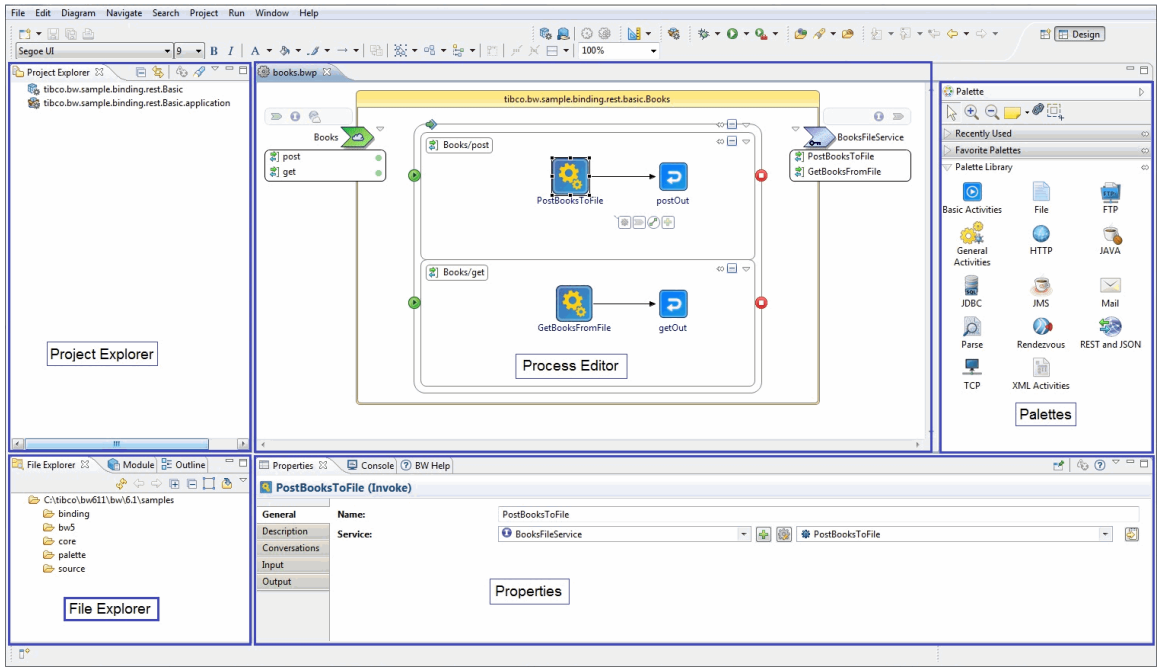
A basic procedure of using TIBCO ActiveMatrix BusinessWorks Plug-in for MongoDB includes:

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

TIBCO Business Studio Overview

TIBCO Business Studio is an Eclipse-based integration development environment that is used to design, develop, and test ActiveMatrix BusinessWorks applications.

TIBCO Business Studio provides a workbench in which you can create, manage, and navigate resources in your workspace. A *workspace* is the central location on your machine where all data files are stored.



The workbench consists of:

- **Menu**: contains menu items such as File, Edit, Diagram, Navigate, Search, Project, Run, Window, and Help.
- **Toolbar**: contains buttons for frequently used commands such as New, Save, Enable/Disable Business Studio Capabilities, Create a new BusinessWorks Application Module, Create a new BusinessWorks Shared Module, Debug, Run, and so on.

- **Perspective**: contains an initial set and layout of views that are required to perform a certain task. TIBCO Business Studio launches the Modeling perspective by default. You can change the perspective from the menu **Window** > **Open Perspective** > *Perspective_Name*.
- **View**: displays resources. 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. You can open a view from the menu **Window** > **Show View** > *View_Name*.
- **Editor**: 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 an ActiveMatrix BusinessWorks 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 provide quick access to activities when you configure a process.

Creating a Project

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

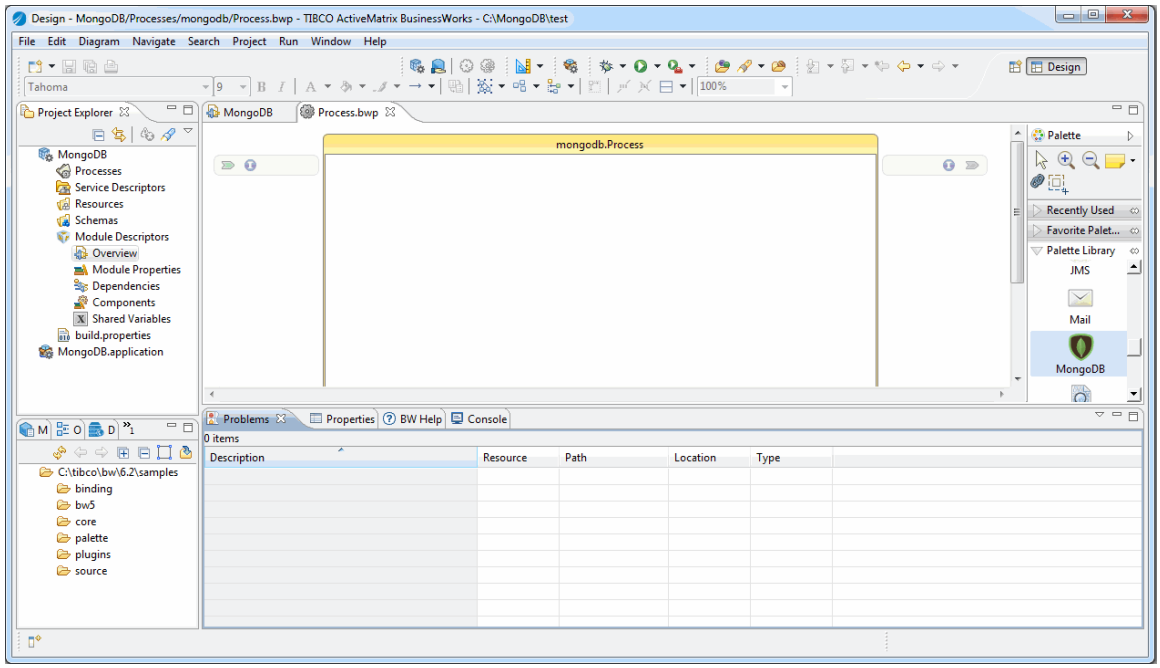
An Eclipse project is an application module configured for TIBCO ActiveMatrix BusinessWorks. An application module is the smallest unit of resources that is named, versioned, and packaged as part of an application.

Procedure

1. Start TIBCO Business Studio using one of the following ways:
   - Microsoft Windows: click **Start** > **All Programs** > **TIBCO** > *TIBCO_HOME* > **TIBCO Business Studio** *version_number* > **Studio for Designers**.
   - Mac OS and 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 MongoDB Connection

After creating a project, you have to create a MongoDB Connection shared resource to connect to the MongoDB server.

Prerequisites

MongoDB 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** > **MongoDB Connection** to open the MongoDB Connection wizard.
3. The resource folder, package name, and resource name of the MongoDB Connection are provided by default. If you do not want to use the default configurations, change them accordingly. Click **Finish** to open the MongoDB Connection Editor.
4. Configure the MongoDB Connection shared resource in the MongoDB Connection editor.

   See MongoDB Connection regarding the configuration fields.
5. Click **Test Connection** to validate the connection.

Configuring a Process

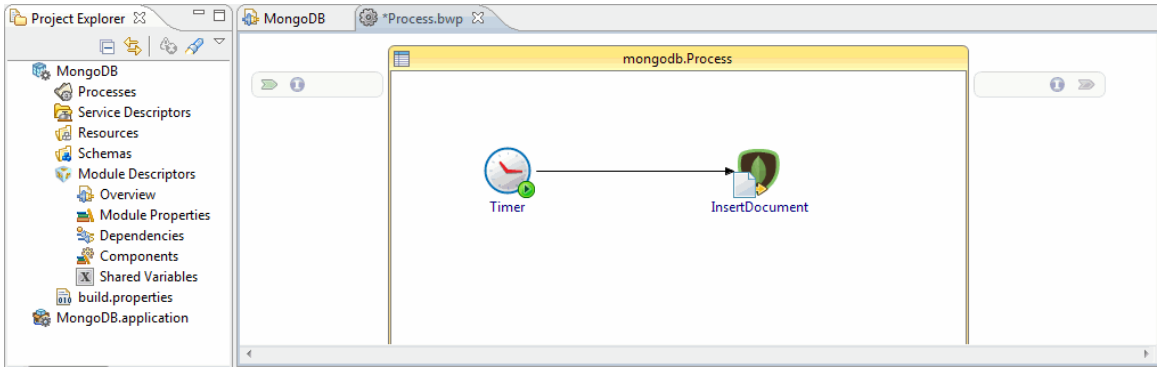
After creating a project, an empty process is created. You can add activities to the empty process to complete a task, such as inserting documents.

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 InsertDocument activity from the MongoDB palette.



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

Testing a Process

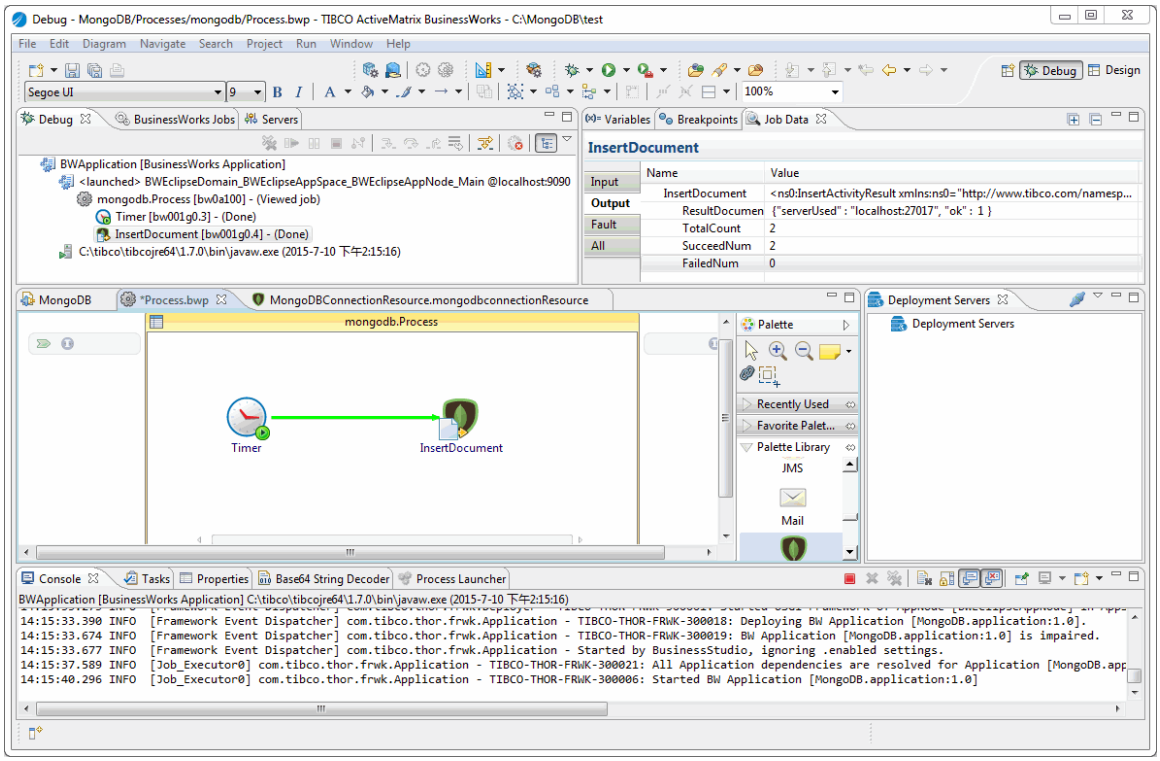
After configuring a process, you can test the process to check if the process completes your task.

Prerequisites

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

Procedure

1. Open the process you have configured in 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 InsertDocument activity.
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, 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.

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

Deploying an application involves the following tasks:

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.

MongoDB Connection

The MongoDB Connection shared resource contains all the necessary parameters that have to be configured when you use the plug-in to connect to the Mongo DB server. The MongoDB Connection is used by all the activities in the MongoDB palette.

General

The following table describes the fields in the General panel:

Field	Module Property?	Description
Package	No	The name of the package where the shared resource is located.
Name	No	The name of the shared resource.
Description	No	A short description for the shared resource.

Configuration

The following table describes the fields in the Configuration panel:

Field	Module Property?	Description
PlainURL	Yes	Select this check box if you want to manually enter the URL for the server.
ConnectionURL	Yes	Enter the URL for the server which the plug-in connects to. Available only when the **PlainURL** check box is selected. The default value is `mongodb://localhost:27017/?connectTimeoutMS=10000&socketTimeoutMS=0&waitQueueTimeoutMS=120000&maxPoolSize=100`
Hostport	Yes	Specify the net address and connection port of the MongoDB server. The default value is `localhost:27017`.
DatabaseName	Yes	Enter the name for the database which you want to connect to. The default value is `databaseName`.
ConnectTimeout(milliseconds)	Yes	Specify the maximum amount of time that a MongoDB driver waits for a connection to be established with a MongoDB server. The default value is `10000`.
SocketTimeout(milliseconds)	Yes	Specify the maximum amount of time that a socket waits for a message sent by the server or the plug-in. The default value is `0`.
MaxWaitTime(milliseconds)	Yes	Specify the maximum amount of time that a thread waits for a socket from the connection pool. The default value is `120000`.

Field	Module Property?	Description
MaxPoolSize	Yes	Specify the maximum number of connections in the connection pool. The default value is `100`.

Security

The following table describes the fields in the Security panel.

Field	Module Property?	Description
Credential	Yes	Select this check box if you want to use the Credential authentication to verify the user name and password when connecting to a MongoDB database.
UserName	Yes	Specify the user name you created in a MongoDB database. Available only when the **Credential** check box is selected.
Password	Yes	Specify the password that is used to access the database. Available only when the **Credential** check box is selected.
Kerberos	Yes	Select this check box if you want to use the Kerberos authentication to verify the Kerberos user name against a MongoDB cluster.
Kerberos UserName	Yes	Specify the Kerberos user name. Available only when the **Kerberos** check box is selected.
SSL	Yes	Select this check box if you want to use the SSL authentication to verify the user and the server.
Key File	Yes	Specify the key file that your deployment uses to verify the user and the server. Available only when the **SSL** check box is selected.
Key Password	Yes	Specify the key password. Available only when the **SSL** check box is selected.
Trust File	Yes	Specify the trust file with CA. Available only when the **SSL** check box is selected.
Trust Password	Yes	Specify the trust password. Available only when the **SSL** check box is selected.
X509 Credential	Yes	Select this check box if you want to use X509 credential to connect to the MongoDB server. Available only when the **SSL** check box is selected.

MongoDB Palette

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

The MongoDB palette contains the following activities:

- InsertDocument
- QueryDocument
- UpdateDocument
- RemoveDocument
- GetMongoDBConnection
- DatabaseCommand
- MapReduce
- EventListener
- WaitForEvent

InsertDocument

You can use the InsertDocument activity to insert one or more documents into a collection. You can also use the InsertDocument activity to insert one or more files into a bucket.

General

The following table describes the fields in the **General** tab of the InsertDocument activity.

Field	Module Property?	Description
Name	No	The name of the activity displayed in the process. The default value is `InsertDocument`.
MongoDB Connection	Yes	Select a shared resource by completing the following steps: 1. Click the **Choose/Create Default Resource**  icon to open the **Select MongodbconnectionResource Resource Template** window, where all the available resources are listed. 2. In the Matching items panel, select one shared resource by clicking it. Click **OK**.

Field	Module Property?	Description
Collection Name	Yes	Specify the name for the collection where documents are inserted.  Adhere to the following rules when specifying the collection name: • Do not start the collection name with `system.` • Do not contain empty strings, and characters such as `$`.

Field	Module Property?	Description
Write Concern	Yes	Select an assurance that MongoDB provides when it reports on the success of a write operation.  The strength of a write concern determines the level of an assurance. When the plug-in performs a write operation with a weak write concern, the write operation is returned quickly. While the plug-in performs a write operation with a strong write concern, the client has to wait for the MongoDB to confirm the write operation before reporting the result of the operation. Select a write concern from the following list, ranked from the weakest to the strongest: • **UNACKNOWLEDGED** With an unacknowledged write concern, MongoDB does not acknowledge the receipt of a write operation and the write operation is returned as soon as the message is written to the socket. Only network errors are captured and handled. • **ACKNOWLEDGED** With an acknowledged write concern, a write operation waits for an acknowledgment from the primary server before returning a result. Network, duplicate key, and other errors are captured and handled. • **JOURNALED** With a journaled write concern, MongoDB acknowledges a write operation after committing the data to the journal. Both network issues and server errors are captured and handled. This type of write concern ensures that MongoDB can recover the data if there is a shutdown or power interruption emergency. Before using this type of write concern, you must have journaling enabled. • **FSYNCED** With a fsynced write concern, a write operation waits for the server to flush data to the disk before returning a result. Network, duplicate key, and other errors are captured and handled. • **REPLICA_ACKNOWLEDGED** With a replica acknowledged write concern, a write operation waits for acknowledgments from at least 2 servers before returning a result. Network, duplicate key, and other errors are captured and handled. • **MAJORITY**

Field	Module Property?	Description
		With a majority write concern, a write operation waits for acknowledgments from a majority of servers before returning a result. Network, duplicate key, and other errors are captured and handled.
Continue On Error	Yes	Select this check box if you want the remaining insert operations to continue when an error occurs.
Is GridFS	No	Select this check box if you want to insert one or more files into a bucket.  GridFS is a specification for storing and retrieving files that exceed the BSON-document size limit of 16 MB.

Description

In the **Description** tab, add a short description for the InsertDocument activity.

Input

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

Input Item	Data Type	Description
`CollectionName`	String	Specify the name for the collection, where documents are inserted, with single or double quotation marks.  Adhere to the following rules when specifying the collection name: • Do not start the collection name with `system.` • Do not contain empty strings, and characters such as `$`. The value specified in this field overwrites the value specified in the **Collection Name** field in the **General** tab.
`Document`	String	Enter a document into this field with double quotation marks. This document is inserted into the specified collection.  If you want to insert multiple documents in the specified collection, you can duplicate the `Document` node by hovering your cursor over the `Document` node and right-clicking to select **Duplicate** from the pop-up menu.

Input Item	Data Type	Description
ContinueOnError	Boolean	Enter `true()` into this field if you want the remaining insert operations to continue when an error occurs. Otherwise, leave this field blank.  If you select the **Continue On Error** check box in the **General** tab, leave this field blank.
IndexDocument	String	Enter a document with double quotation marks to index documents in the specified collection. The index document can be further used in querying documents. For more information on creating an index, refer to http://docs.mongodb.org/manual/administration/indexes-creation/.
IndexOptionsDocument	String	Enter an index type for the index document with double quotation marks, such as `"{unique:1}"`. For more information on index options, refer to http://docs.mongodb.org/manual/core/index-properties/.
The following fields are displayed when the **Is GridFS** check box is selected in the **General** tab.		
BucketName	String	Specify the name for the bucket, where files are inserted, with single or double quotation marks.  Adhere to the following rules when specifying the bucket name: • Do not start the bucket name with `system.` • Do not contain empty strings, and characters such as `$`.
GridFSFile	N/A	This node supports duplication.  If you want to insert multiple files into a bucket, you can duplicate the `GridFSFile` node by hovering your cursor over the `GridFSFile` node and right-clicking to select **Duplicate** from the pop-up menu.
FullFileName	String	Enter the absolute path for the file with single or double quotation marks. This file is inserted into the specified bucket.  At least 8GB memory is required for the machine, if the inserted file is 1GB.
NewFileName	String	Enter a new name for the file with single or double quotation marks when you insert it into a bucket. The new file name can be used when you query the file.

Output

The following table describes the items in the **Output** tab of the InsertDocument activity.

Output Item	Data Type	Description
`ResultDocument`	String	The document that records the returned value after using the InsertDocument activity.
`TotalCount`	Integer	The total documents that are inserted into the specified collection.
`SucceedNum`	Integer	The total documents that are successfully inserted into the specified collection.
`FailedNum`	Integer	The total documents that are not successfully inserted into the specified collection.

Fault

The following table describes the items in the **Fault** tab of the InsertDocument activity.

Error Schema Element	Data Type	Description
`MongoDBPluginException`		
`msg`	String	This error message is displayed when the MongoDB plug-in captures an exception for the InsertDocument activity.
`msgCode`	String	This error code is displayed when the MongoDB plug-in captures an exception for the InsertDocument activity.
`MongoException`		
`msg`	String	This error message is displayed when MongoDB encounters internal errors.
`msgCode`	String	This error code is displayed when MongoDB encounters internal errors.
`NetWorkException`		
`msg`	String	This error message is displayed when a network-related exception occurs.
`msgCode`	String	This error code is displayed when a network-related exception occurs.
`JSONParseException`		
`msg`	String	This error message is displayed when an invalid JSON object is passed to the JSON parser.

Error Schema Element	Data Type	Description
`msgCode`	String	This error code is displayed when an invalid JSON object is passed to the JSON parser.
`DuplicateKeyException`		
`msg`	String	This error message is displayed when an error of duplicate key occurs.
`msgCode`	String	This error code is displayed when an error of duplicate key occurs.

QueryDocument

You can use the QueryDocument activity to query one or more documents from a collection. You can also use the QueryDocument activity to query one or more files from a bucket.

General

The following table describes the fields in the **General** tab of the QueryDocument activity.

Field	Module Property ?	Description
Name	No	The name of the activity displayed in the process. The default value is `QueryDocument`.
MongoDB Connection	Yes	Select a shared resource by completing the following steps: 1. Click the **Choose/Create Default Resource** icon to open the **Select MongodbconnectionResource Resource Template** window, where all the available resources are listed. 2. In the Matching items panel, select one shared resource by clicking it. Click **OK**.
Collection Name	Yes	Specify the name for the collection where documents are queried. Adhere to the following rules when specifying the collection name: • Do not start the collection name with `system.` • Do not contain empty strings, and characters such as `$`.

Field	Module Property ?	Description
Query Type	Yes	Specify the query type that you want to use. Select one from the following list: • **FIND_ONE** Finds the first document that matches the query condition. • **FIND_MANY** Finds all documents that match the query condition. • **COUNT** Counts the total number of documents that match the query condition. • **DISTINCT** Finds an array of distinct values for a specified field in documents. • **AGGREGATE** Processes documents in a collection by using versatile stage-based data processing pipeline or map-reduce operations.
Read Preference	Yes	Select the read preference when the plug-in queries documents from a collection. Select one from the following list: • **PRIMARY** Reads from the primary member of a replica set. • **PRIMARY_PREFERRED** Reads from the primary member preferentially, but if the primary member is unavailable, reads from secondary members of a replica set. • **SECONDARY** Reads from secondary members of a replica set. • **SECONDARY_PREFERRED** Reads from secondary members preferentially, but if no secondary members are available, reads from the primary member of a replica set. • **NEAREST** Reads from a member of a replica set with the least network latency, irrespective of the member type of the replica set.  The query preference functionality does not support the aggregate query type. When you select **AGGREGATE** from the **Query Type** list, the **Read Preference** list is unavailable.

Field	Module Property ?	Description
Is GridFS	Yes	Select this check box if you want to query one or more files from a bucket.  GridFS is a specification for storing and retrieving files that exceed the BSON-document size limit of 16 MB.
File Query Type	Yes	Select a file query type from the following list: • **FIND_ONE** Finds the first file that matches the query condition. • **FIND_MANY** Finds all files that match the query condition. Available only when the **Is GridFS** check box is selected.
File Content Output Type	Yes	Select an output type for a file from the following list: • **JAVA_OBJECT** Produces an object which contains the returned file content as the output. • **BINARY** Produces a byte array as the output. • **WRITE_TO_FILE** Creates a new file that contains the returned file content. Available only when the **Is GridFS** check box is selected.

Description

In the **Description** tab, add a short description for the QueryDocument activity.

Input

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

Input Item	Data Type	Description
The following fields are displayed when **FIND_ONE** in the **Query Type** list is selected in the **General** tab.		
`CollectionName`	String	Specify the name for the collection, where query operations are performed, with single or double quotation marks.  The value specified in this field overwrites the value specified in the **Collection Name** field in the **General** tab.

Input Item	Data Type	Description
QueryDocument	String	Enter a document with double quotation marks. This document is used to query one or more documents from a specified collection.
ReturnFieldsDocument	String	Enter a document with double quotation marks. This document is used to decide the return field values in the queried documents.
SortDocument	String	Enter a document with double quotation marks. This document is used to define the display order of the documents in the query result.
The following fields are displayed when **FIND_MANY** in the **Query Type** list is selected in the **General** tab.		
CollectionName	String	Specify the name for the collection, where query operations are performed, with single or double quotation marks.  The value specified in this field overwrites the value specified in the **Collection Name** field in the **General** tab.
QueryDocument	String	Enter a document with double quotation marks. This document is used to query one or more documents from a specified collection.
ReturnFieldsDocument	String	Enter a document with double quotation marks. This document is used to decide the return field values in the queried documents.
Skip	Integer	Enter a number to decide how many documents to skip before the query result is displayed.
Limit	Integer	Enter a number to decide how many documents to display as the query result.
HintDocument	String	Enter a document with double quotation marks. This document contains an index name or index specification and is used to overwrite the default index in a collection.
SortDocument	Integer	Enter a document with double quotation marks. This document is used to define the display order of the documents in the query result.
The following fields are displayed when **COUNT** in the **Query Type** list is selected in the **General** tab.		

Input Item	Data Type	Description
CollectionName	String	Specify the name for the collection, where query operations are performed, with single or double quotation marks.  The value specified in this field overwrites the value specified in the **Collection Name** field in the **General** tab.
QueryDocument	String	Enter a document with double quotation marks. This document is used to query one or more documents from a specified collection.
Skip	Integer	Enter a number to decide how many documents to skip before the query result is displayed.
Limit	Integer	Enter a number to decide how many documents to display as the query result.
The following fields are displayed when **DISTINCT** in the **Query Type** list is selected in the **General** tab.		
CollectionName	String	Specify the name for the collection, where query operations are performed, with single or double quotation marks.  The value specified in this field overwrites the value specified in the **Collection Name** field in the **General** tab.
QueryDocument	String	Enter a document with double quotation marks. This document is used to query one or more documents from a specified collection.
DistinctField	String	Enter a field name with single or double quotation marks. This document decides the distinct field values returned from the queried documents.
The following fields are displayed when **Aggregate** in the **Query Type** list is selected in the **General** tab.		
CollectionName	String	Specify the name for the collection, where query operations are performed, with single or double quotation marks.  The value specified in this field overwrites the value specified in the **Collection Name** field in the **General** tab.

Input Item	Data Type	Description
Document	String	Enter a document with double quotation marks. This document contains an aggregation pipeline or a map-reduce operation and is used to perform an aggregation in a collection. For more information on defining an aggregation pipeline or a map-reduce operation, refer to http://docs.mongodb.org/manual/core/aggregation-introduction.
The following fields are displayed when the **Is GridFS** check box, and **FIND_ONE** in the **File Query Type** list are selected in the **General** tab.		
BucketName	String	Enter the name for the bucket, where you can query one matching file, with single or double quotation marks.
FileName	String	Enter the name for the file with single or double quotation marks.
GenerateNewFile	String	Enter the absolute path for the file, where you want to save the query output result, with double quotation marks. Available only when **WRITE_TO_FILE** in the **File Content Output Type** list is selected in the **General** tab.
The following fields are displayed when the **Is GridFS** check box, and **FIND_MANY** in the **File Query Type** list are selected in the **General** tab.		
BucketName	String	Enter the name for the bucket, where you can query all matching files, with single or double quotation marks.
QueryDocument	String	Enter a document with double quotation marks. This document is used to query files from a bucket.
OutputFileDirectory	String	Enter the directory for the output file with single or double quotation marks. Available only when **WRITE_TO_FILE** in the **File Content Output Type** list is selected in the **General** tab.

Output

The following table describes the items in the **Output** tab of the QueryDocument activity.

Output Item	Data Type	Description
The following item is displayed only when **FIND_ONE** in the **QUERY TYPE** list is selected in the **General** tab.		
Document	String	The first document that matches the query condition.
The following items are displayed only when **FIND_MANY** in the **QUERY TYPE** list is selected in the **General** tab.		

Output Item	Data Type	Description
Document	String	All documents that match the query condition, or an empty set if no documents match the query condition.
TotalCount	Integer	The total number of documents that match the query condition in the specified collection.
RemainingCount	Integer	The total number of documents that match the query condition but that are not displayed in the output result.
The following item is displayed only when **COUNT** in the **QUERY TYPE** list is selected in the **General** tab.		
Count	Integer	The total number of documents that match the query condition in the specified collection.
The following item is displayed only when **DISTINCT** in the **QUERY TYPE** list is selected in the **General** tab.		
Object	String	The document that contains an array of distinct values of the specified field.
The following item is displayed only when **AGGREGATE** in the **QUERY TYPE** list is selected in the **General** tab.		
Document	String	The document that contains the aggregation result of the map-reduce operation or aggregation pipeline.
The following items are displayed only when the **IS GridFS** check box, and **JAVA_OBJECT** in the **FILE CONTENT OUTPUT TYPE** list are selected in the **General** tab.		
Document	String	The document that contains data from the queried file.
QueryFileContentStream	Java Object	The instance of a file stream returned by the QueryDocument activity.
The following items are displayed only when the **IS GridFS** check box, and **BINARY** in the **FILE CONTENT OUTPUT TYPE** list are selected in the **General** tab.		
Document	String	The document that contains data from the queried file.
BinaryContent	Binary	The file content returned as Binary.
The following items are displayed only when the **IS GridFS** check box, and **WRITE_TO_FILE** in the **FILE CONTENT OUTPUT TYPE** list are selected in the **General** tab.		
Document	String	The document that contains data from the queried file.
FileName	String	The file name returned after using the QueryDocument activity.

Fault

The following table describes the items in the **Fault** tab of the QueryDocument activity.

Error Schema Element	Data Type	Description
MongoDBPluginException		
msg	String	This error message is displayed when the MongoDB plug-in captures an exception for the QueryDocument activity.
msgCode	String	This error code is displayed when the MongoDB plug-in captures an exception for the QueryDocument activity.
MongoException		
msg	String	This error message is displayed when MongoDB encounters internal errors.
msgCode	String	This error code is displayed when MongoDB encounters internal errors.
NetWorkException		
msg	String	This error message is displayed when a network-related exception occurs.
msgCode	String	This error code is displayed when a network-related exception occurs.
JSONParseException		
msg	String	This error message is displayed when an invalid JSON object is passed to the JSON parser.
msgCode	String	This error code is displayed when an invalid JSON object is passed to the JSON parser.
CursorNotFoundException		
msg	String	This error message is displayed when a cursor is not found.
msgCode	String	This error code is displayed when a cursor is not found.

UpdateDocument

You can use the UpdateDocument activity to update one or more documents in a collection.

General

The following table describes the fields in the **General** tab of the UpdateDocument activity.

Field	Module Property ?	Description
Name	No	The name of the activity displayed in the process. The default value is `UpdateDocument`.
MongoDB Connection	Yes	Select a shared resource by completing the following steps: 1. Click the **Choose/Create Default Resource** icon to open the **Select MongodbconnectionResource Resource Template** window, where all the available resources are listed. 2. In the Matching items panel, select one shared resource by clicking it. Click **OK**.
Collection Name	Yes	Specify the name for the collection where documents are updated. Adhere to the following rules when specifying the collection name: • Do not start the collection name with `system.` • Do not contain empty strings, and characters such as `$`.

Field	Module Property ?	Description
Write Concern	Yes	Select an assurance that MongoDB provides when it reports on the success of a write operation.  The strength of a write concern determines the level of an assurance. When the plug-in performs a write operation with a weak write concern, the write operation is returned quickly. While the plug-in performs a write operation with a strong write concern, the client has to wait for the MongoDB to confirm the write operation before reporting the result of the operation. Select a write concern from the following list, ranked from the weakest to the strongest: • **UNACKNOWLEDGED** With an unacknowledged write concern, MongoDB does not acknowledge the receipt of a write operation and the write operation is returned as soon as the message is written to the socket. Only network errors are captured and handled. • **ACKNOWLEDGED** With a receipt acknowledged write concern, a write operation waits for an acknowledgment from the primary server before returning a result. Network, duplicate key, and other errors are captured and handled. • **JOURNALED** With a journaled write concern, MongoDB acknowledges a write operation after committing the data to the journal. Both network issues and server errors are captured and handled. This type of write concern ensures that MongoDB can recover the data if there is a shutdown or power interruption emergency. Before using this type of write concern, you must have journaling enabled. • **FSYNCED** With a fsynced write concern, a write operation waits for the server to flush data to the disk before returning a result. Network, duplicate key, and other errors are captured and handled. • **REPLICA_ACKNOWLEDGED** With a replica acknowledged write concern, a write operation waits for acknowledgments from at least 2 servers before returning a result. Network, duplicate key, and other errors are captured and handled. • **MAJORITY**

Field	Module Property ?	Description
		With a majority write concern, a write operation waits for acknowledgments from a majority of servers before returning a result. Network, duplicate key, and other errors are captured and handled.
Continue On Error	Yes	Select this check box if you want the remaining update operations to continue when an error occurs.

Description

In the **Description** tab, add a short description for the UpdateDocument activity.

Input

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

Input Item	Data Type	Description
`CollectionName`	String	Specify the name for the collection, where documents are updated, with single or double quotation marks.  Adhere to the following rules when specifying the collection name: • Do not start the collection name with `system.` • Do not contain empty strings, and characters such as `$`. The value specified in this field overwrites the value specified in the **Collection Name** field in the **General** tab
`Query Document`	String	Enter a document with double quotation marks. This document is used to query one or more documents from the specified collection.
`Update Document`	String	Enter a document with double quotation marks. This document is used to update one or more documents from the specified collection.
`Upsert`	Boolean	Enter `true()` into this field if you want to insert the document specified in the **UpdateDocument** field when no matching document is found.
`Multi`	Boolean	Enter `true()` into this field if you want to update multiple documents which match the query condition.
`ContinueOnError`	Boolean	Enter `true()` into this field if you want the remaining update operations to continue when an error occurs.  If you select the **Continue On Error** check box in the **General** tab, leave this field blank.

Output

The following table describes the items in the **Output** tab of the UpdateDocument activity.

Output Item	Data Type	Description
`ResultDocument`	String	The document that records the status of the UpdateDocument activity.
`SucceedNum`	Integer	The total number of documents that are updated successfully in the specified collection.
`FailedNum`	Integer	The total number of documents that are not successfully updated in the specified collection.
`Upserts`	Integer	The new document or documents that are inserted into the specified collection.

Fault

The following table describes the items in the **Fault** tab of the UpdateDocument activity.

Error Schema Element	Data Type	Description
`MongoDBPluginException`		
`msg`	String	This error message is displayed when the MongoDB plug-in captures an exception for the UpdateDocument activity.
`msgCode`	String	This error code is displayed when the MongoDB plug-in captures an exception for the UpdateDocument activity.
`MongoException`		
`msg`	String	This error message is displayed when MongoDB encounters internal errors.
`msgCode`	String	This error code is displayed when MongoDB encounters internal errors.
`NetWorkException`		
`msg`	String	This error message is displayed when a network-related exception occurs.
`msgCode`	String	This error code is displayed when a network-related exception occurs.
`JSONParseException`		
`msg`	String	This error message is displayed when an invalid JSON object is passed to the JSON parser.

Error Schema Element	Data Type	Description
`msgCode`	String	This error code is displayed when an invalid JSON object is passed to the JSON parser.

RemoveDocument

You can use the RemoveDocument activity to remove one or more documents from a collection. You can also use the plug-in to remove one or more files from a bucket.

General

The following table describes the fields in the **General** tab of the RemoveDocument activity.

Field	Module Property?	Description
Name	No	The name of the activity displayed in the process. The default value is `RemoveDocument`.
MongoDB Connection	Yes	Select a shared resource by completing the following steps: 1. Click the **Choose/Create Default Resource** icon to open the **Select MongodbconnectionResource Resource Template** window, where all the available resources are listed. 2. In the Matching items panel, select one shared resource by clicking it. Click **OK**.
Collection Name	Yes	Specify the name for the collection where documents are removed. Adhere to the following rules when specifying the collection name: • Do not start the collection name with `system.` • Do not contain empty strings, and characters such as `$`.

Field	Module Property?	Description
Write Concern	Yes	Select an assurance that MongoDB provides when it reports on the success of a write operation.  The strength of a write concern determines the level of an assurance. When the plug-in performs a write operation with a weak write concern, the write operation is returned quickly. While the plug-in performs a write operation with a strong write concern, the client has to wait for the MongoDB to confirm the write operation before reporting the result of the operation. Select a write concern from the following list, ranked from the weakest to the strongest: • **UNACKNOWLEDGED** With an unacknowledged write concern, MongoDB does not acknowledge the receipt of a write operation and the write operation is returned as soon as the message is written to the socket. Only network errors are captured and handled. • **ACKNOWLEDGED** With an acknowledged write concern, a write operation waits for an acknowledgment from the primary server before returning a result. Network, duplicate key, and other errors are captured and handled. • **JOURNALED** With a journaled write concern, MongoDB acknowledges a write operation after committing the data to the journal. Both network issues and server errors are captured and handled. This type of write concern ensures that MongoDB can recover the data if there is a shutdown or power interruption emergency. Before using this type of write concern, you must have journaling enabled. • **FSYNCED** With a fsynced write concern, a write operation waits for the server to flush data to the disk before returning a result. Network, duplicate key, and other errors are captured and handled. • **REPLICA_ACKNOWLEDGED** With a replica acknowledged write concern, a write operation waits for acknowledgments from at least 2 servers before returning a result. Network, duplicate key, and other errors are captured and handled. • **MAJORITY**

Field	Module Property?	Description
		With a majority write concern, a write operation waits for acknowledgments from a majority of servers before returning a result. Network, duplicate key, and other errors are captured and handled.
Continue On Error	Yes	Select this check box if you want the remaining remove operations to continue when an error occurs.
Is GridFS	No	Select this check box if you want to remove one or more files from a bucket.

Description

In the **Description** tab, add a short description for the RemoveDocument activity.

Input

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

Input Item	Data Type	Description
`CollectionName`	String	Enter the name for the collection, where documents are removed, with single or double quotation marks.  The value specified in this field overwrites the value specified in the **Collection Name** field in the **General** tab.
`QueryDocument`	String	Enter a document with double quotation marks. This document is used to query one or more documents from a specified collection.
`ContinueOnError`	Boolean	Enter `true()` into this field if you want the remaining remove operations to continue when an error occurs.  If you select the **Continue On Error** check box in the General tab, leave this field blank.
`Multi`	Boolean	Enter `true()` into this field if you want to remove all documents that match the query condition.
The following fields are displayed when the **Is GridFS** check box is selected in the **General** tab.		
`BucketName`	String	Enter the name for the bucket, where one or more files are removed, with single or double quotation marks.
`QueryDocument`	String	Enter a document with double quotation marks. This document is used to query one or more files from a bucket.

Output

The following table describes the items in the **Output** tab of the RemoveDocument activity.

Output Item	Data Type	Description
`ResultDocument`	String	The document that records the returned value after using the RemoveDocument activity.
`SucceedNum`	Integer	The total number of documents that are successfully removed from the specified collection.
`FailedNum`	Integer	The total number of documents that are not successfully removed from the specified collection.

Fault

The following table describes the items in the **Fault** tab of the RemoveDocument activity.

Error Schema Element	Data Type	Description
`MongoDBPluginException`		
`msg`	String	This error message is displayed when the MongoDB plug-in captures an exception for the RemoveDocument activity.
`msgCode`	String	This error code is displayed when the MongoDB plug-in captures an exception for the RemoveDocument activity.
`MongoException`		
`msg`	String	This error message is displayed when MongoDB encounters internal errors.
`msgCode`	String	This error code is displayed when MongoDB encounters internal errors.
`NetWorkException`		
`msg`	String	This error message is displayed when a network-related exception occurs.
`msgCode`	String	This error code is displayed when a network-related exception occurs.
`JSONParseException`		
`msg`	String	This error message is displayed when an invalid JSON object is passed to the JSON parser.
`msgCode`	String	This error code is displayed when an invalid JSON object is passed to the JSON parser.

GetMongoDBConnection

You can use the GetMongoDBConnection activity to get a MongoDB Connection instance, which can be used in Java Code activities to perform some operations in MongoDB.

General

The following table describes the fields in the **General** tab of the GetMongoDBConnection activity.

Field	Module Property ?	Description
Name	No	The name of the activity displayed in the process. The default value is `GetMongoDBConnection`.
MongoDB Connection	Yes	Select a shared resource by completing the following steps: 1. Click the **Choose/Create Default Resource**  icon to open the **Select MongodbconnectionResource Resource Template** window, where all the available resources are listed. 2. In the Matching items panel, select one shared resource by clicking it. Click **OK**.

Description

In the **Description** tab, add a short description for the GetMongoDBConnection activity.

Output

The following table describes the item in the **Output** tab of the GetMongoDBConnection activity.

Output Item	Data Type	Description
`MongoDBConnection Accessor`	Java Object	The instance of the MongoDB connection obtained by the GetMongoDBConnection activity. The following is the definition of the instance: `com.tibco.bw.palette.mongodb.runtime.resources.MongoDBConnectionAccessorImpldb8d5e37-6926-47ea-93bf-a7c0129ead29`

Fault

The following table describes the items in the **Fault** tab of the GetMongoDBConnection activity.

Error Schema Element	Data Type	Description
`MongoDBPluginException`		

Error Schema Element	Data Type	Description
`msg`	String	This error message is displayed when the MongoDB plug-in captures an exception for the GetMongoDBConnection activity.
`msgCode`	String	This error code is displayed when the MongoDB plug-in captures an exception for the GetMongoDBConnection activity.
`MongoException`		
`msg`	String	This error message is displayed when MongoDB encounters internal errors.
`msgCode`	String	This error code is displayed when MongoDB encounters internal errors.
`NetWorkException`		
`msg`	String	This error message is displayed when a network-related exception occurs.
`msgCode`	String	This error code is displayed when a network-related exception occurs.

DatabaseCommand

You can use the DatabaseCommand activity to perform commands in a MongoDB database.

For more information about the commands supported by MongoDB, refer to http://docs.mongodb.org/manual/reference/command/.

General

The following table describes the fields in the **General** tab of the DatabaseCommand activity.

Field	Module Property?	Description
Name	No	The name of the activity displayed in the process. The default value is `DataBaseCommand`.
MongoDB Connection	Yes	Select a shared resource by completing the following steps: 1. Click the **Choose/Create Default Resource** icon to open the **Select MongodbconnectionResource Resource Template** window, where all the available resources are listed. 2. In the Matching items panel, select one shared resource by clicking it. Click **OK**.

Field	Module Property?	Description
Read Reference	Yes	Select the read preference when the plug-in reads documents from a collection. Select one from the following list: • **PRIMARY** Reads from the primary member of a replica set. • **PRIMARY_PREFERRED** Reads from the primary member preferentially, but if the primary member is unavailable, reads from secondary members of a replica set. • **SECONDARY** Reads from secondary members of a replica set. • **SECONDARY_PREFERRED** Reads from secondary members preferentially, but if no secondary members are available, reads from the primary member of a replica set. • **NEAREST** Reads from a member of a replica set with the least network latency, irrespective of the member type of the replica set.

Description

In the **Description** tab, add a short description for the DatabaseCommand activity.

Input

The following table describes the field in the **Input** tab of the DatabaseCommand activity.

Input Item	Data Type	Description
`Document`	String	Enter a document with double quotation marks. The document contains the database command that you can use to perform operations in a database.

Output

The following table describes the item in the **Output** tab of the DatabaseCommand activity.

Output Item	Data Type	Description
`ResultDocument`	String	The document that records the returned value of the DatabaseCommand activity.

Fault

The following table describes the items in the **Fault** tab of the DatabaseCommand activity.

Error Schema Element	Data Type	Description
MongoDBPluginException		
msg	String	This error message is displayed when the MongoDB plug-in captures an exception for the DatabaseCommand activity.
msgCode	String	This error code is displayed when the MongoDB plug-in captures an exception for the DatabaseCommand activity.
MongoException		
msg	String	This error message is displayed when MongoDB encounters internal errors.
msgCode	String	This error code is displayed when MongoDB encounters internal errors.
NetWorkException		
msg	String	This error message is displayed when a network-related exception occurs.
msgCode	String	This error code is displayed when a network-related exception occurs.
JSONParseException		
msg	String	This error message is displayed when an invalid JSON object is passed to the JSON parser.
msgCode	String	This error code is displayed when an invalid JSON object is passed to the JSON parser.

MapReduce

You can use the MapReduce activity to perform map-reduce and finalize operations to perform aggregations in a collection by using custom JavaScript functions.

General

The following table describes the fields in the **General** tab of the MapReduce activity.

Field	Module Property?	Description
Name	No	The name of the activity displayed in the process. The default value is `MapReduce`.

Field	Module Property?	Description
MongoDB Connection	Yes	Select a shared resource by completing the following steps: 1. Click the **Choose/Create Default Resource** icon to open the **Select MongodbconnectionResource Resource Template** window, where all the available resources are listed. 2. In the Matching items panel, select one shared resource by clicking it. Click **OK**.
Collection Name	Yes	Specify the name for the collection where map and reduce operations are performed. Adhere to the following rules when specifying the collection name: • Do not start the collection name with `system.` • Do not contain empty strings, and characters such as `$`.
Read Preference	Yes	Select the read preference when the plug-in reads documents from a collection. Select one from the following list: • **PRIMARY** Reads from the primary member of a replica set. • **PRIMARY_PREFERRED** Reads from the primary member preferentially, but if the primary member is unavailable, reads from secondary members of a replica set. • **SECONDARY** Reads from secondary members of a replica set. • **SECONDARY_PREFERRED** Reads from secondary members preferentially, but if no secondary members are available, reads from the primary member of a replica set. • **NEAREST** Reads from a member of a replica set with the least network latency, irrespective of the member type of the replica set.
Map Function	No	Enter your JavaScript in the **Map Function** field. The map function maps an operation in every document in a collection. This operation emits key and value pairs.

Field	Module Property?	Description
Reduce Function	No	Enter your JavaScript in the **Reduce Function** field.  The reduce function reduces all values of a specific key to one element which contains the result.
Finalize Function	No	Enter your JavaScript in the **Finalize Function** field.  The finalize function follows the reduce function and is used to modify the output result.

Description

In the **Description** tab, add a short description for the MapReduce activity.

Input

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

Input Item	Data Type	Description
`CollectionName`	String	Enter the name for the collection, where map-reduce operations are performed, with single or double quotation marks.  The value specified in this field overwrites the value specified in the **Collection Name** field in the **General** tab.
`JavaScriptMapFunction`	JavaScript	Enter a JavaScript with double quotation marks to map an operation in every document in a collection. The operation emits key and value pairs.  The value specified in this field overwrites the value specified in the **Map Function** field in the **General** tab.
`JavaScriptReduceFunction`	JavaScript	Enter a JavaScript with double quotation marks to reduce all values of a specific key to one element which contains the result.  The value specified in this field overwrites the value specified in the **Reduce Function** field in the **General** tab.
`JavaScriptFinalizeFunction`	JavaScript	Enter a JavaScript with double quotation marks to modify the output result.  The value specified in this field overwrites the value specified in the **Finalize Function** field in the **General** tab.

Input Item	Data Type	Description
OutputTarget	String	Enter a collection name with double quotation marks to store a map-reduce operation result.  If the output type is inline, leave this field blank.
OutputType	String	Select one output type from the following lists: • **INLINE** Returns the result inline. • **REPLACE** Replaces the contents of the specified collection, if the specified collection exists . • **MERGE** Merges the new result with the existing result if the output collection already exists. If an existing document has the same key as the new result, the new result overwrites that existing document. • **REDUCE** Merges the new result with the existing result if the output collection already exists. If an existing document has the same key as the new result, the reduce function is applied to both, the new and the existing documents, and the existing document is overwritten by the result.
QueryDocument	String	Enter a document with double quotation marks. This document is used to query documents to be mapped and reduced.
Limit	String	Enter a number to decide how many documents to keep after the query document operation.
SortDocument	String	Enter a document with double quotation marks. This document is used to define the display order of documents after the limit operation.

Output

The following table describes the item in the **Output** tab of the MapReduce activity.

Output Item	Data Type	Description
ResultDocument	String	The document that records the returned value of the MapReduce activity. The document varies when you select different output type.

Fault

The following table describes the items in the **Fault** tab of the MapReduce activity.

Error Schema Element	Data Type	Description
MongoDBPluginException		
msg	String	This error message is displayed when the MongoDB plug-in captures an exception for the MapReduce activity.
msgCode	String	This error message is displayed when the MongoDB plug-in captures an exception for the MapReduce activity.
MongoException		
msg	String	This error message is displayed when MongoDB encounters internal errors.
msgCode	String	This error code is displayed when MongoDB encounters internal errors.
NetWorkException		
msg	String	This error message is displayed when a network-related exception occurs.
msgCode	String	This error code is displayed when a network-related exception occurs.
JSONParseException		
msg	String	This error message is displayed when an invalid JSON object is passed to the JSON parser.
msgCode	String	This error code is displayed when an invalid JSON object is passed to the JSON parser.

EventListener

You can use the EventListener activity to listen to write operations in a collection, such as inserting or removing documents or files. The EventListener activity is a start activity and it is used for a replica set.

General

The following table describes the fields in the **General** tab of the EventListener activity.

Field	Module Property?	Description
Name	No	The name of the activity displayed in the process. The default value is `EventListener`.

Field	Module Property?	Description
MongoDB Connection	Yes	Select a shared resource by completing the following steps: 1. Click the **Choose/Create Default Resource** icon to open the **Select MongodbconnectionResource Resource Template** window, where all the available resources are listed. 2. In the Matching items panel, select one shared resource by clicking it. Click **OK**.
Collection Filter	Yes	Enter the name for the collection which you want the EventListener activity to listen to, such as the `test` collection. If you leave this field blank, the activity listens to all the collections in the specified database.
Listen for Insert Event	Yes	Select this check box if you want to listen to insert events in the specified collection.
Listen for Remove Event	Yes	Select this check box if you want to listen to remove events in the specified collection.
Listen for Update Event	Yes	Select this check box if you want to listen to update events in the specified collection.

Description

In the **Description** tab, add a short description for the EventListener activity.

Advanced

The following table describes the fields in the **Advanced** tab of the EventListener activity.

Field	Module Property?	Description
`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 executed sequentially in the order in which the process instances are created.
`Custom Job Id`	No	This field contains an XPath expression that specifies a custom job ID for every job in the process instance.

Conversations

In the **Conversations** tab, you can click **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 items in the **Output** tab of the EventListener activity.

Output Item	Data Type	Description
`NameSpace`	String	The combination of database name and collection name.
`OperationType`	String	The operation type listened to by the EventListener activity.
`ResultDocument`	String	The document which records the returned value of the EventListener activity.

WaitForEvent

You can use the WaitForEvent activity to wait for write operations in a collection. When the WaitForEvent activity is performed, the process instance is suspended and waits for the specified change to occur before resuming. This WaitForEvent activity is a signal-in activity and it is used for a replica set.

General

The following table describes the fields in the **General** tab of the WaitForEvent activity.

Field	Module Property?	Description
Name	No	The name of the activity displayed in the process. The default value is `WaitForEvent`.
MongoDB Connection	Yes	Select a shared resource by completing the following steps: 1. Click the **Choose/Create Default Resource**  icon to open the **Select MongodbconnectionResource Resource Template** window, where all the available resources are listed. 2. In the Matching items panel, select one shared resource by clicking it. Click **OK**.
Collection Filter	Yes	Enter the name for the collection which you want to listen to, such as the `test` collection.
Listen for Insert Event	Yes	Select this check box if you want to listen to insert events in the specified collection.

Field	Module Property?	Description
Listen for Remove Event	Yes	Select this check box if you want to listen to remove events in the specified collection.
Listen for Update Event	Yes	Select this check box if you want to listen to update events in the specified collection.

Description

In the **Description** tab, add a short description for the WaitForEvent activity.

Event

The following table describes the fields in the **Event** tab of the WaitForEvent activity.

Field	Module Property?	Description
`Event Timeout (seconds)`	No	Specify the amount of time (in seconds) for the event. If the event times out, an error is logged and the event is discarded.
`Activity Timeout (msec)`	Yes	Specify the amount of time (in milliseconds) the WaitForEvent activity waits for an event message. If in the specified time the WaitForEvent activity does not receive an event message, the status of the WaitForEvent activity changes to unavailable.

Conversations

In the **Conversations** tab, you can click **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 items in the **Output** tab of the WaitForEvent activity.

Output Item	Data Type	Description
`NameSpace`	String	The combination of database name and collection name.
`OperationType`	String	The operation type listened to by the WaitForEvent activity.
`ResultDocument`	String	The document that records the result of operations in a specified collection.

Fault

The following table describes the items in the **Fault** tab of the WaitForEvent activity.

Error Schema Element	Data type	Description
`msg`	String	This error message is displayed when the MongoDB plug-in captures an exception for the WaitForEvent activity.
`msgCode`	String	This error code is displayed when the MongoDB plug-in captures an exception for the WaitForEvent activity.

Working with the Sample Projects

Working through the sample projects helps you understand how TIBCO ActiveMatrix BusinessWorks Plug-in for MongoDB operates.

TIBCO ActiveMatrix BusinessWorks Plug-in for MongoDB is packaged with sample projects within the installer. After installing the plug-in, you can find the sample projects which are located in the *TIBCO_HOME*\bw\palettes\mongodb*version*\samples directory.

Importing the Sample Projects

Before running the projects, you must import them to TIBCO Business Studio.

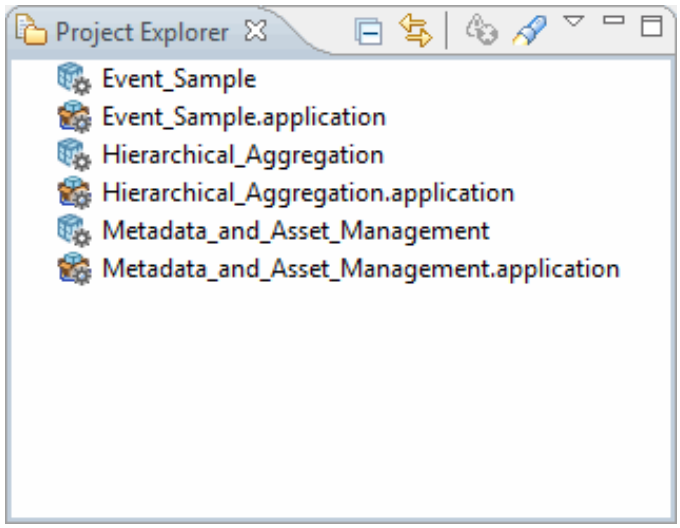
Procedure

1. Start TIBCO Business Studio using one of the following ways:
   - Microsoft Windows: click **Start** > **All Programs** > **TIBCO** > *TIBCO_HOME* > **TIBCO Business Studio** *version_number* > **Studio for Designers**
   - Mac OS and 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 samples file. Click **Finish**.

   The samples file is located in the *TIBCO_HOME*\bw\palettes\mongodb*version*\samples directory.

Result

The sample projects are imported to TIBCO Business Studio.



Configuring MongoDB Connection

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

Prerequisites

Importing the Sample Projects.

Procedure

1. In the Project Explorer view, expand **MongoDB_Sample**.
2. In the Resources folder, double-click **MongoDBConnectionResource.mongodbconnectionResource**.
3. In MongoDBConnection Editor, configure each field accordingly.
4. On the toolbar, click the **Save** icon to save your changes.

Running the Sample Projects

The sample projects show how to use the plug-in to insert, query, update and remove documents or files in MongoDB. The sample projects also show how to get a MongoDB connection instance, run database command, listen to write operations in MongoDB, and perform map-reduce operations in MongoDB.

Prerequisites

Ensure that you have imported the sample projects to TIBCO Business Studio, as described in Importing the Sample Projects, and configured MongoDB Connection, as described in Configuring MongoDB Connection.

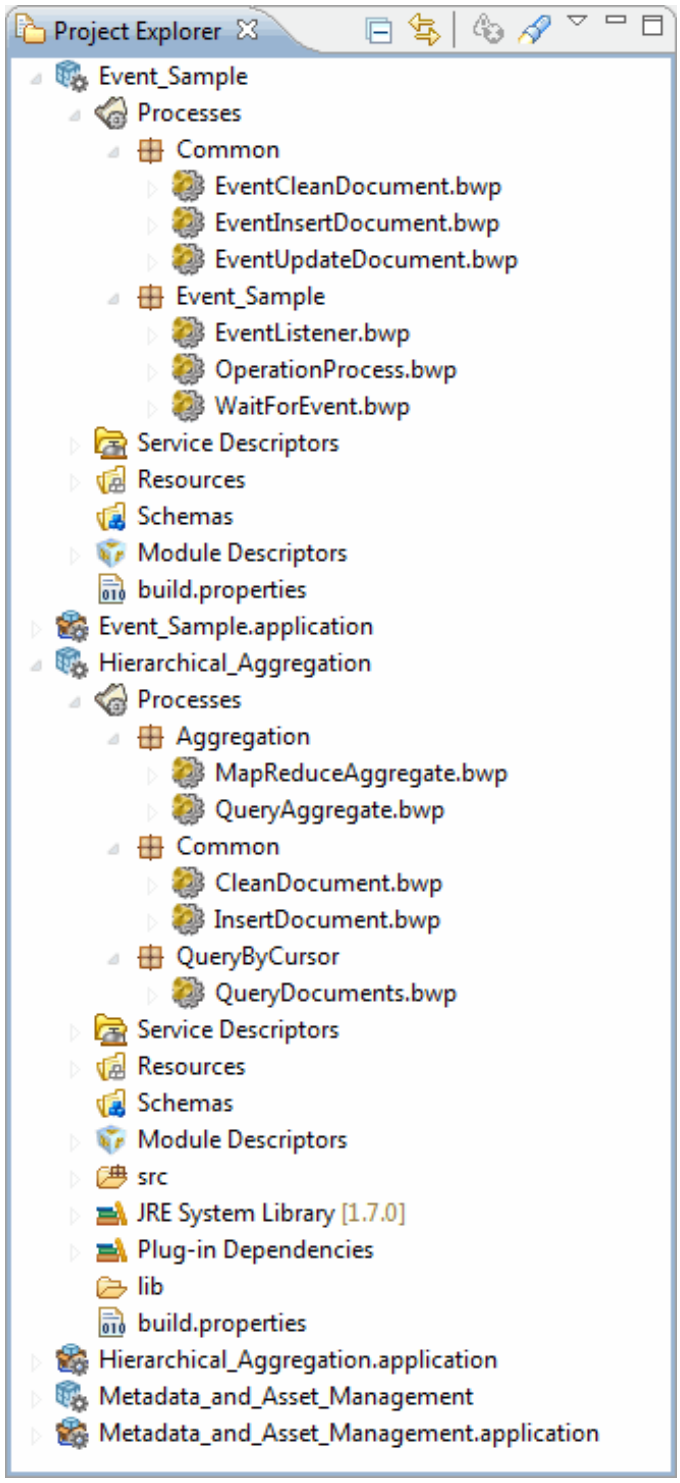
Procedure

1. In the Project Explorer view, expand the **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 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 **Event_Sample.application**, **Hierarchical_Aggregation.application**, or **Metadata_and_Asset_Management.application**.
7. Click **Run** to run the selected process.
8. Click the **Terminate** icon to stop the process.

Configuration of Sample Processes

The `samples` file contains three projects. Each process in the three projects has different functions.

After importing the sample projects, expand all projects in the Project Explorer view. All the processes are displayed. See Importing the Sample Projects for more information.



Event_Sample Project

The Event_Sample project contains the following processes:

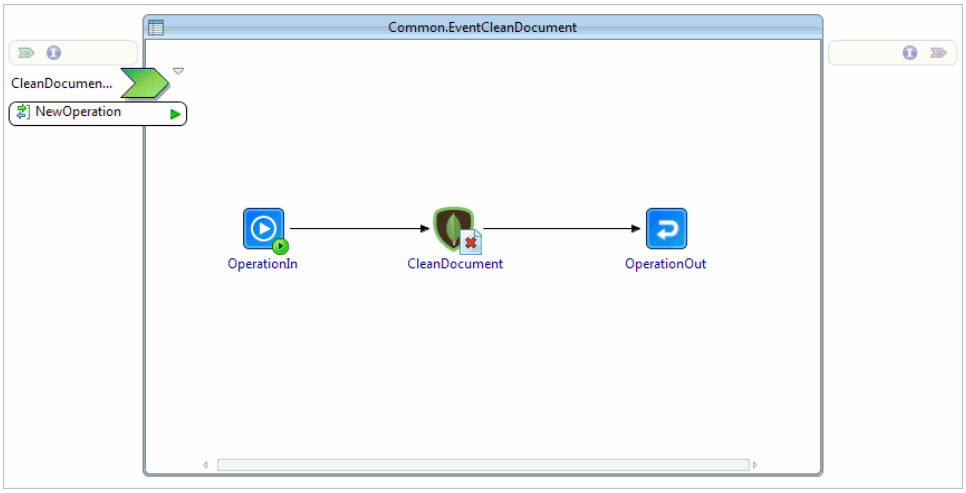
- EventCleanDocument.bwp
- EventInsertDocument.bwp
- EventUpdateDocument.bwp

- EventListener.bwp
- OperationProcess.bwp
- WaitForEvent.bwp

EventCleanDocument.bwp

This subprocess demonstrates how to use the plug-in to clean documents in a specified collection.

The subprocess is designed with the following activities:



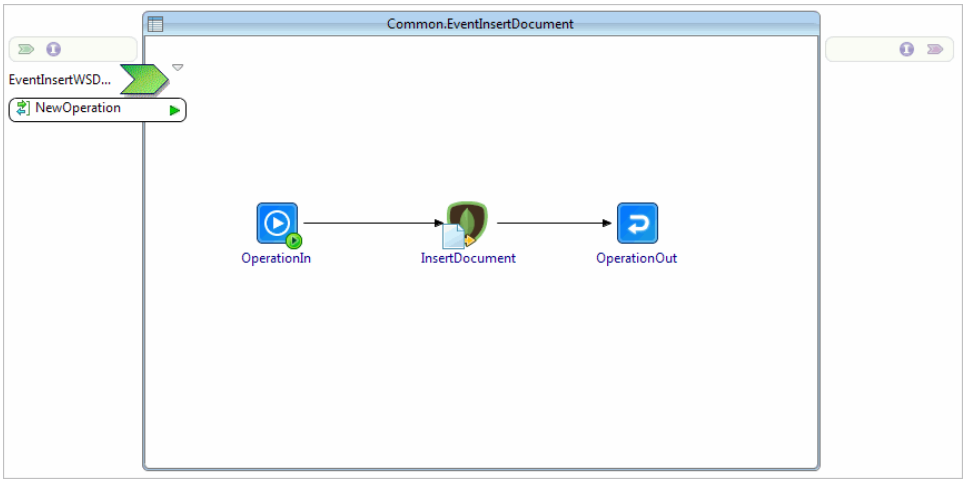
The following table describes the activities in the subprocess.

Activity	Description
OperationIn	Receives data and starts the subprocess.
CleanDocument	Deletes documents in the specified collection.
OperationOut	Delivers the output data and ends the subprocess.

EventInsertDocument.bwp

This subprocess demonstrates how to use the plug-in to insert documents into a specified collection.

The subprocess is designed with the following activities:

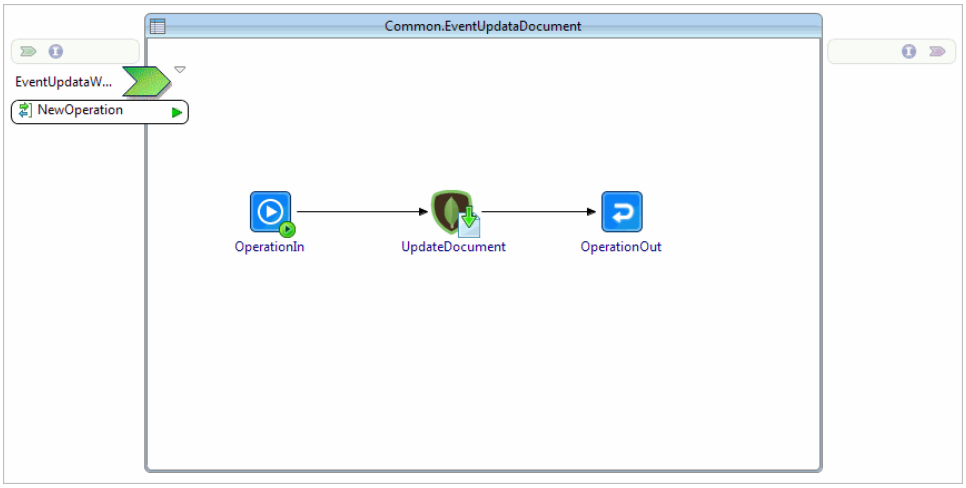


The following table describes the activities in the subprocess.

Activity	Description
OperationIn	Receives data and starts the subprocess.
InsertDocument	Inserts documents into the specified collection.
OperationOut	Delivers the output data and ends the subprocess.

EventUpdateDocument.bwp

This subprocess demonstrates how to use the plug-in to update documents in a specified collection. The subprocess is designed with the following activities:



The following table describes the activities in the subprocess.

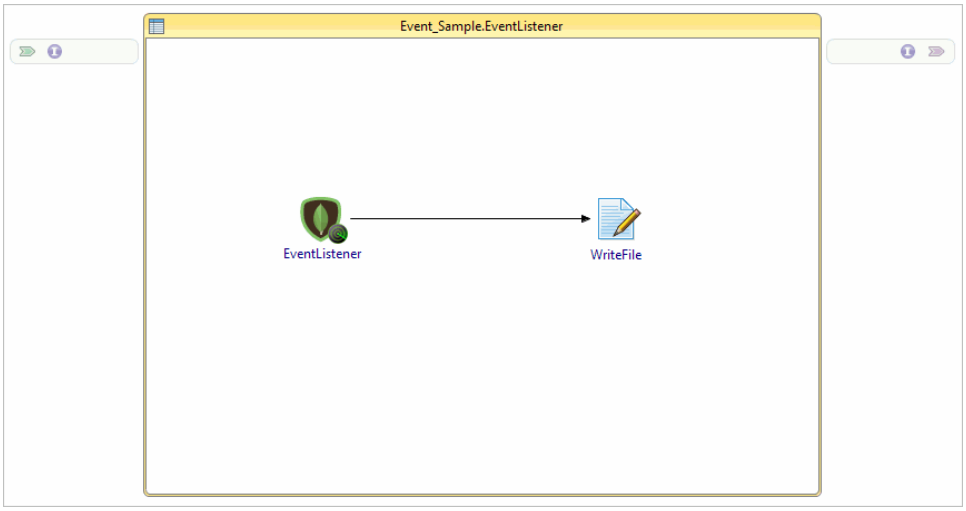
Activity	Description
OperationIn	Receives data and starts the subprocess.

Activity	Description
UpdateDocument	Updates documents in the specified collection.
OperationOut	Delivers data and ends the subprocess.

EventListener.bwp

This process demonstrates how to use the plug-in to listen to write operations in a specified collection.

The process is designed with the following activities:



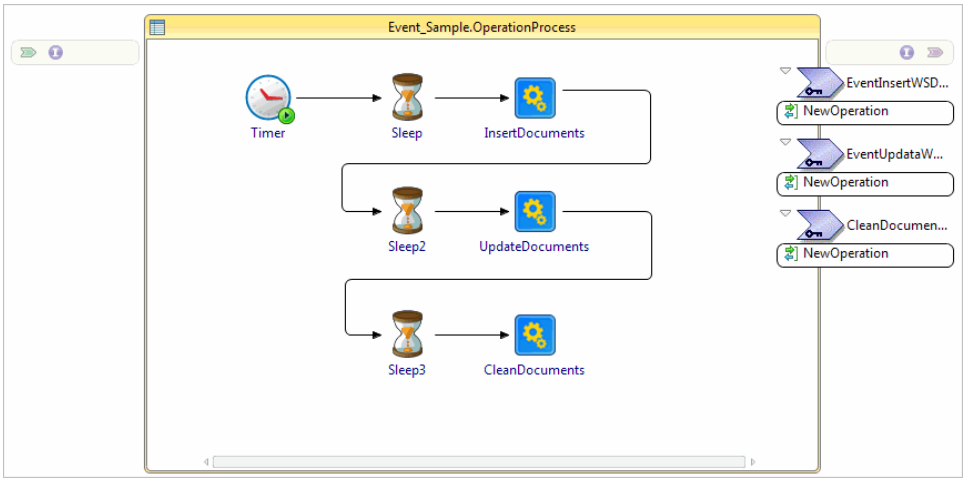
The following table describes the activities in the process.

Activity	Description
EventListener	Listens to write operations in the specified collection.
WriteFile	Writes the result obtained from the EventListener activity into a specified file.

OperationProcess.bwp

This process demonstrates how to use the plug-in to insert, update and delete documents in a specified collection.

The process is designed with the following activities:



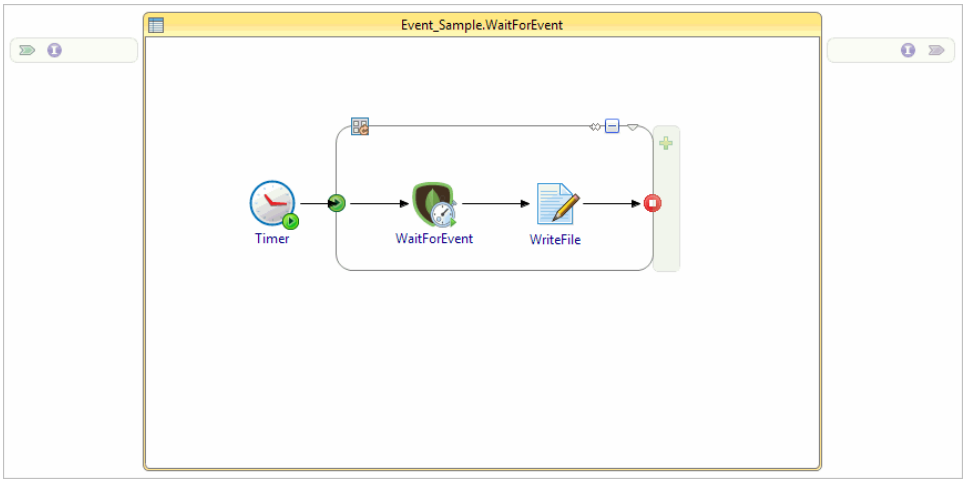
The following table describes the activities in the process.

Activity	Description
Timer	Starts the process.
Sleep	Sets the time interval for the InsertDocuments activity.
InsertDocuments	Invokes the EventInsertDocument subprocess to insert a group of documents in the specified collection.
Sleep2	Sets the time interval for the UpdateDocuments activity.
UpdateDocuments	Invokes the EventUpdateDocument subprocess to update one or more documents in the specified collection.
Sleep3	Sets the time interval for the CleanDocuments activity.
CleanDocuments	Invokes the EventCleanDocument subprocess to deletes one or more documents from the specified collection.

WaitForEvent.bwp

This process demonstrates how to use the plug-in to wait for and listen to write operations in a specified collection.

The process is designed with the following activities:



The following table describes the activities in the process.

Activity	Description
Timer	Starts the process.
WaitForEvent	Waits for and listens to write operations in the specified collection.
WriteFile	Writes the result obtained from the WaitForEvent activity into a specified file.

Hierarchical_Aggregation Project

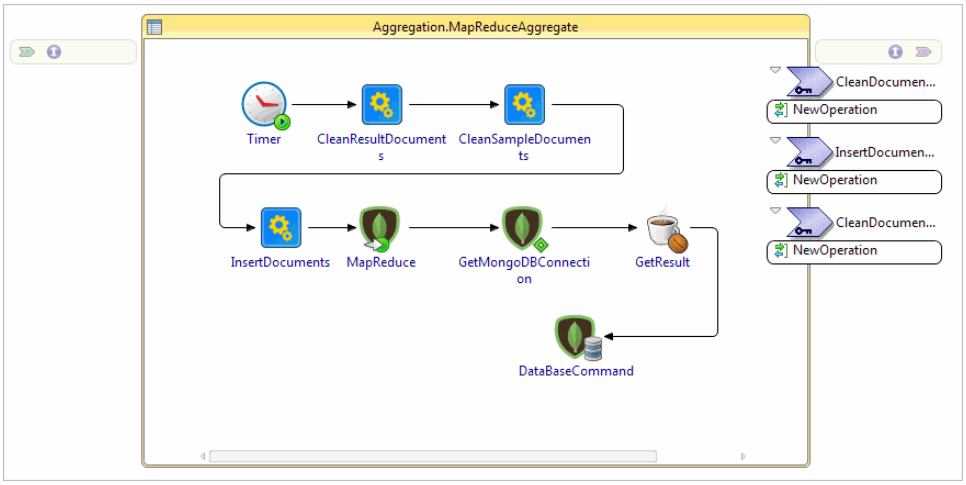
The Hierarchical_Aggregation project contains the following processes:

- MapReduceAggregate.bwp
- QueryAggregate.bwp
- CleanDocument.bwp
- InsertDocument.bwp
- QueryDocuments.bwp

MapReduceAggregate.bwp

This process demonstrates how to use the plug-in to perform map-reduce operations in a specified collection.

The process is designed with the following activities:



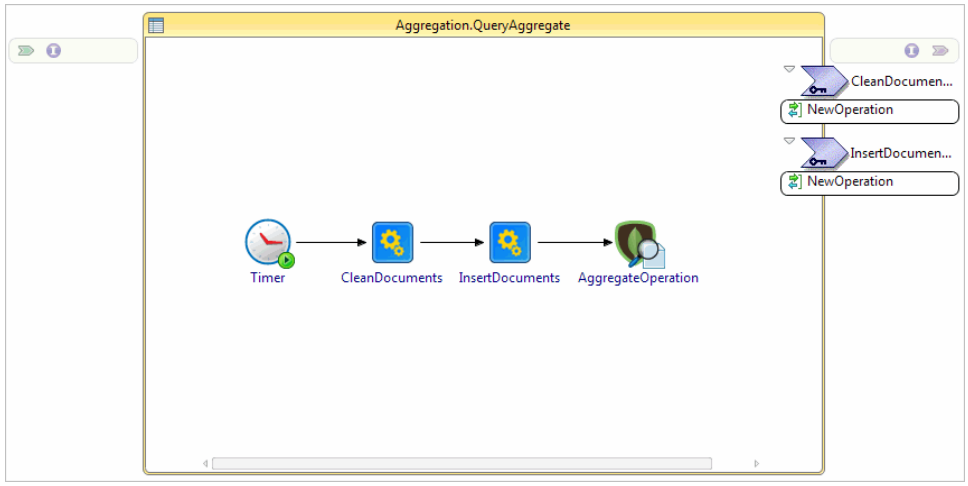
The following table describes the activities in the process.

Activity	Description
Timer	Starts the process.
CleanResultDocuments	Invokes the CleanDocument subprocess to delete all result documents from the specified collection.
CleanSampleDocuments	Invokes the CleanDocument subprocess to delete all sample documents from the specified collection.
InsertDocuments	Invokes the InsertDocument subprocess to insert a group of documents into the specified collection.
MapReduce	Performs a map-reduce operation on the specified collection and writes the result into another collection.
GetMongoDBConnection	Obtains a MongoDB Connection reference.
GetResult	Utilizes the obtained Connection reference to get information related to the collection where the result of the map-reduce operation is located.
DataBaseCommand	Performs database commands in the collection where the result of the map-reduce operation is located.

QueryAggregate.bwp

This process demonstrates how to use the plug-in to perform an aggregation query from a specified collection.

The process is designed with the following activities:



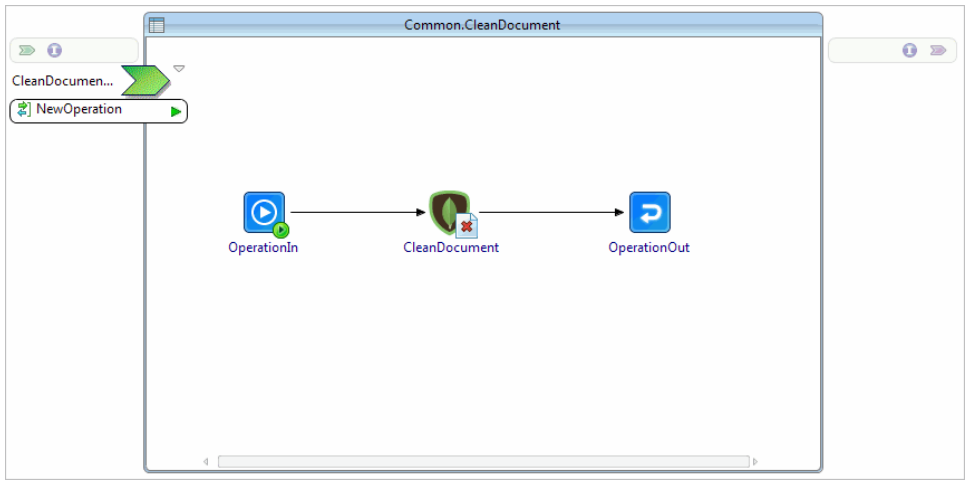
The following table describes the activities in the process.

Activity	Description
Timer	Starts the process.
CleanDocument	Invokes the CleanDocument subprocess to delete all documents from the specified collection.
InsertDocument	Invokes the InsertDocument subprocess to insert a group of documents into the specified collection.
AggregateOperation	Performs an aggregation query from the specified collection.

CleanDocument.bwp

This subprocess demonstrates how to use the plug-in to delete all documents from a specified collection.

The subprocess is designed with the following activities:



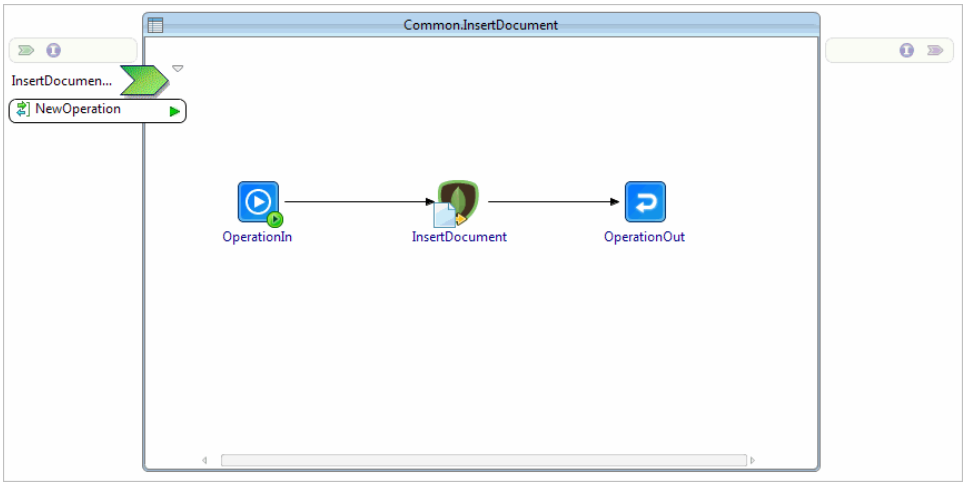
The following table describes the activities in the subprocess.

Activity	Description
OperationIn	Receives data and starts the subprocess.
CleanDocuments	Deletes all documents from the specified collection.
OperationOut	Delivers the output data and ends the subprocess.

InsertDocument.bwp

This subprocess demonstrates how to use the plug-in to insert a group of documents into a specified collection.

The subprocess is designed with the following activities:



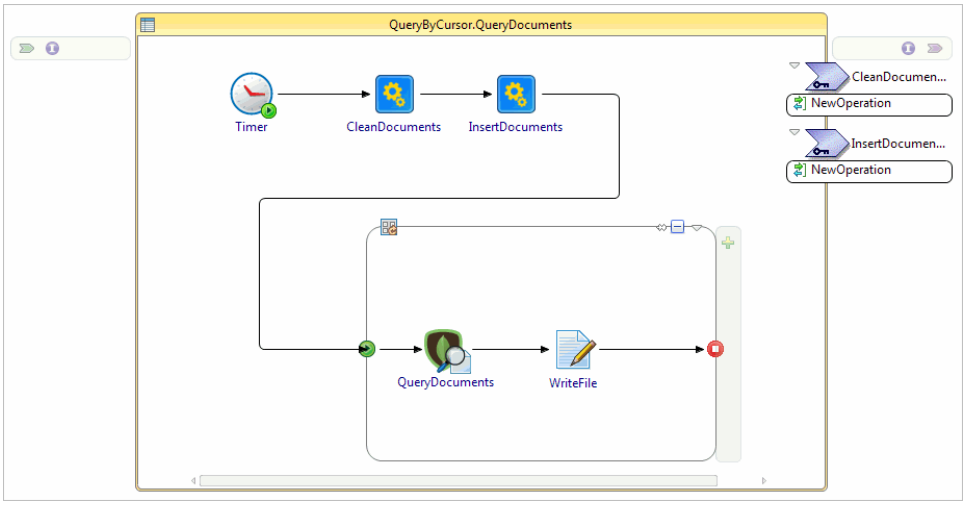
The following table describes the activities in the subprocess.

Activity	Description
OperationIn	Receives data and starts the subprocess.
InsertDocument	Inserts a group of documents into the specified collection.
OperationOut	Delivers the output data and ends the subprocess.

QueryDocuments.bwp

This process demonstrates how to use the plug-in to query multiple documents from a specified collection.

The process is designed with the following activities:



The following table describes the activities in the process.

Activity	Description
Timer	Starts the process.
CleanDocuments	Invokes the CleanDocument subprocess to delete all documents from the specified collection.
InsertDocuments	Invokes the InsertDocument subprocess to insert a group of documents into the specified collection.
QueryDocument	Queries documents from the specified collection and returns one document each time.
WriteFile	Writes each document returned from the QueryDocument activity into the specified file.

Metadata_and_Asset_Management Project

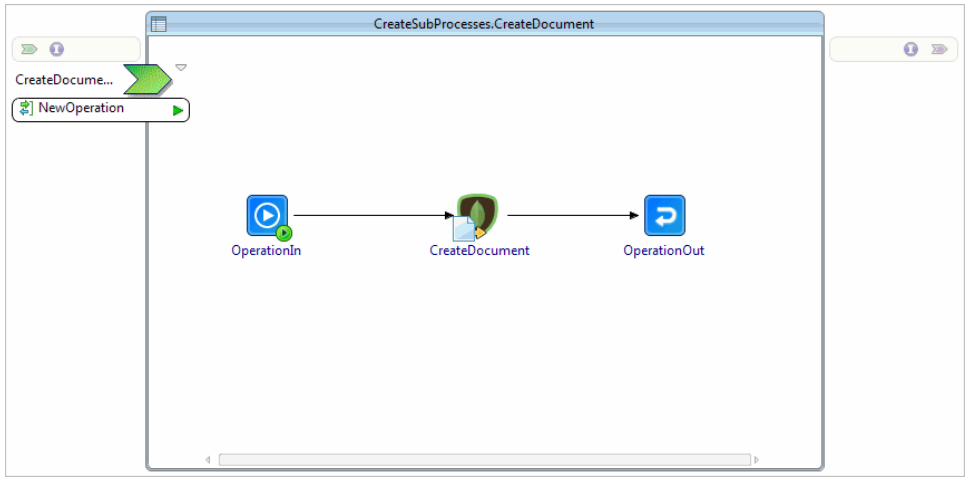
The Metadata_and_Asset_Management project contains the following processes:

- CreateDocument.bwp
- CreateGridFSFile.bwp
- DeleteDocument.bwp
- DeletePhoto.bwp
- QueryDocument.bwp
- QueryDocumentbyIndex.bwp
- DropIndex.bwp
- UpdateDocument.bwp
- GridFSPhotoOperate.bwp
- QuerybyIndex.bwp
- QueryJSONtoXML.bwp
- SavaXMLtoMongoDB.bwp

CreateDocument.bwp

This subprocess demonstrates how to use the plug-in to insert a group of documents into a specified collection.

The subprocess is designed with the following activities:



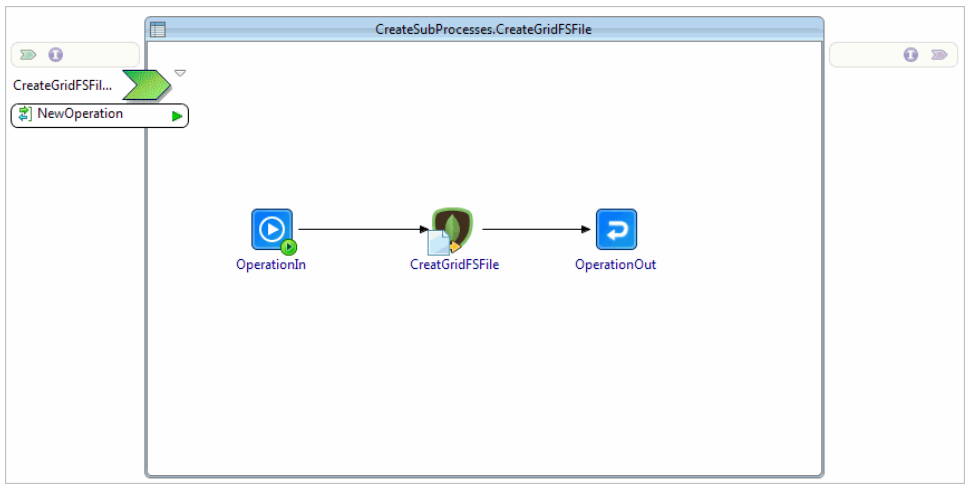
The following table describes the activities in the subprocess.

Activity	Description
OperationIn	Receives data and starts the subprocess.
CreateDocument	Inserts a group of documents into the specified collection.
OperationOut	Delivers the output data and ends the subprocess.

CreateGridFSFile.bwp

This subprocess demonstrates how to use the plug-in to insert a file into a specified bucket.

The subprocess is designed with the following activities:



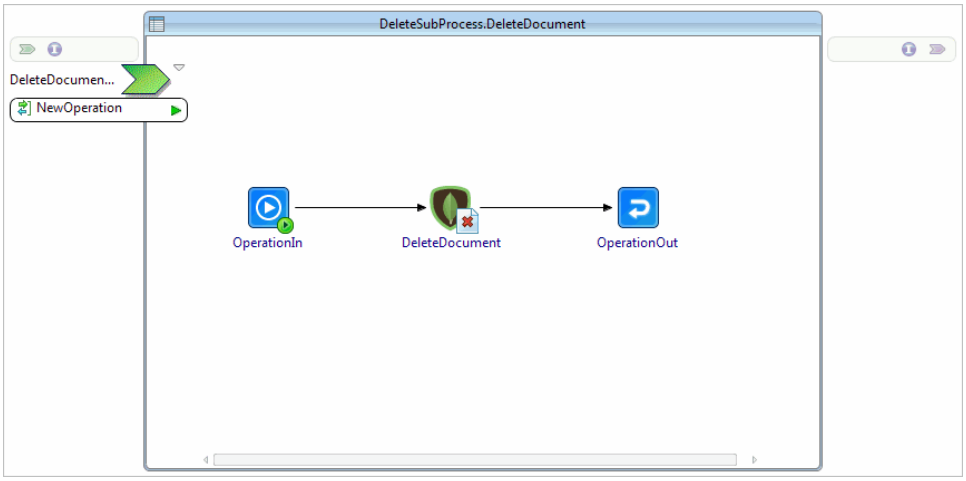
The following table describes the activities in the subprocess.

Activity	Description
OperationIn	Receives data and starts the subprocess.
CreateGridFSFile	Inserts a file into the specified bucket.
OperationOut	Delivers the output data and ends the subprocess.

DeleteDocument.bwp

This subprocess demonstrates how to use the plug-in to delete all documents from a specified collection.

The subprocess is designed with the following activities:



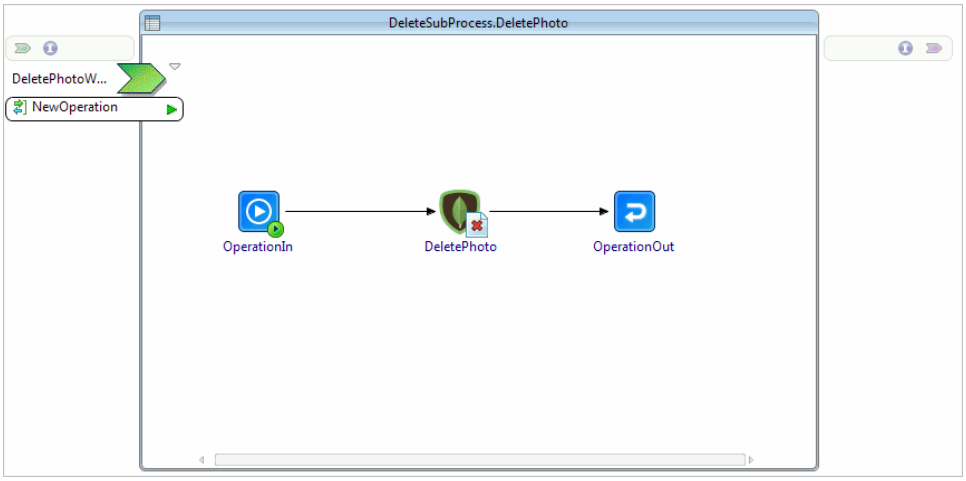
The following table describes the activities in the subprocess.

Activity	Description
OperationIn	Receives data and starts the subprocess.
DeleteDocument	Deletes all documents from the specified collection.
OperationOut	Delivers the output data and ends the subprocess.

DeletePhoto.bwp

This subprocess demonstrates how to use the plug-in to delete a file from a specified bucket.

The subprocess is designed with the following activities:

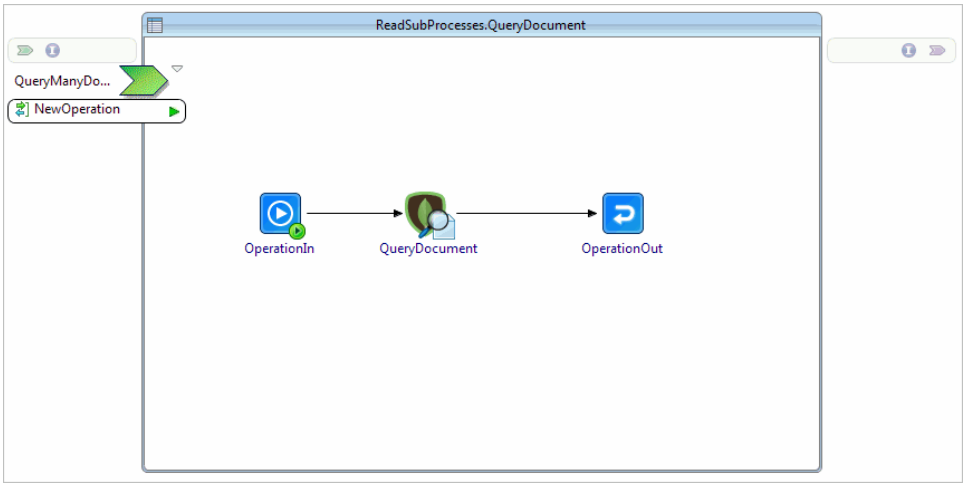


The following table describes the activities in the subprocess.

Activity	Description
OperationIn	Receives data and starts the subprocess.
DeletePhoto	Deletes a file from the specified bucket.
OperationOut	Delivers the output data and ends the subprocess.

QueryDocument.bwp

This subprocess demonstrates how to use the plug-in to query a document from a specified collection. The subprocess is designed with the following activities:



The following table describes the activities in the subprocess.

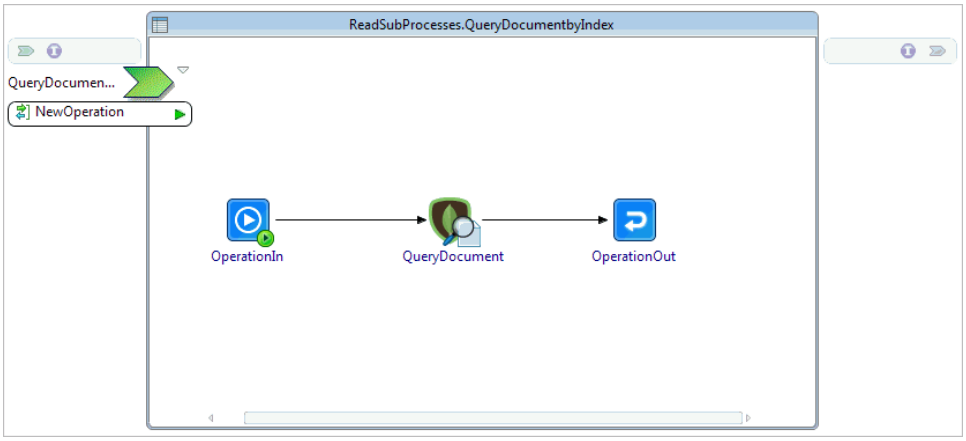
Activity	Description
OperationIn	Receives data and starts the subprocess.

Activity	Description
QueryDocument	Queries a document from the specified collection.
OperationOut	Delivers the output data and ends the subprocess.

QueryDocumentbyIndex.bwp

This subprocess demonstrates how to use the plug-in to query a document by index from a specified collection.

The subprocess is designed with the following activities:



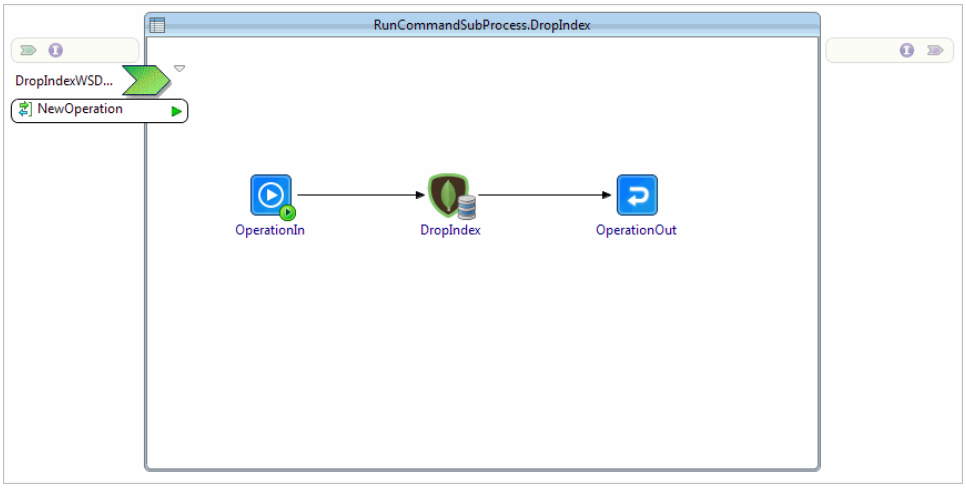
The following table describes the activities in the subprocess.

Activity	Description
OperationIn	Receives data and starts the subprocess.
QueryDocument	Queries a document from the specified collection.
OperationOut	Delivers the output data and ends the subprocess.

DropIndex.bwp

This subprocess demonstrates how to use the plug-in to delete all indexes from a specified collection.

The subprocess is designed with the following activities:

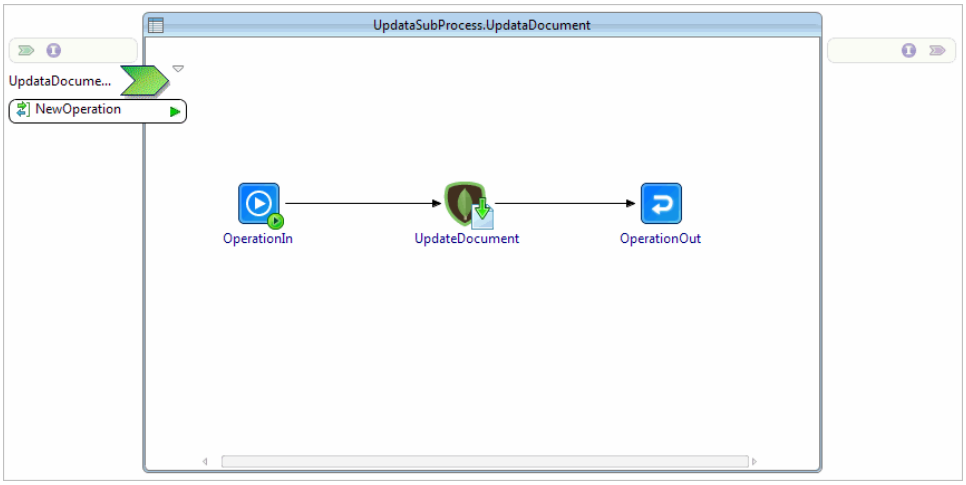


The following table describes the activities in the subprocess.

Activity	Description
OperationIn	Receives data and starts the subprocess.
DropIndex	Deletes all indexes from the specified collection.
OperationOut	Delivers the output data and ends the subprocess.

UpdateDocument.bwp

This subprocess demonstrates how to use the plug-in to update a document in a specified collection. The subprocess is designed with the following activities:



The following table describes the activities in the subprocess.

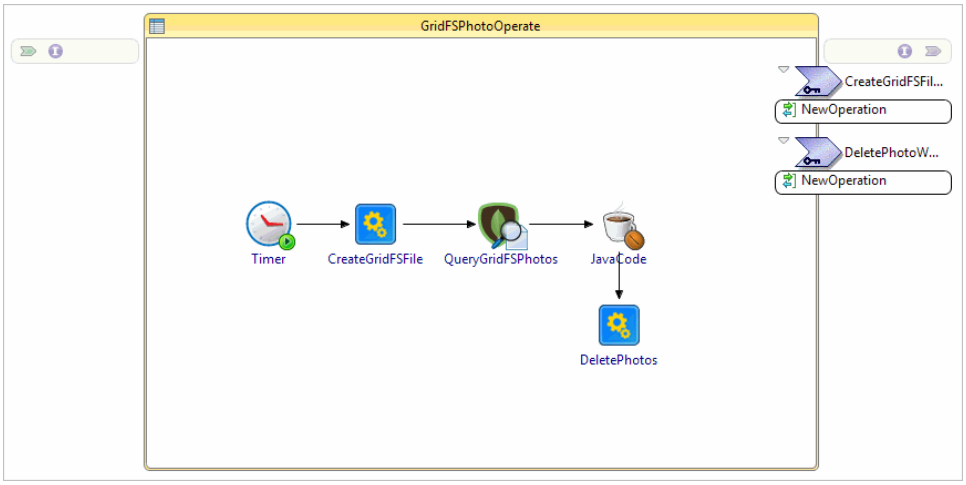
Activity	Description
OperationIn	Receives data and starts the subprocess.

Activity	Description
UpdateDocument	Updates a document in the specified collection.
OperationOut	Delivers the output data and ends the subprocess.

GridFSPhotoOperate.bwp

This process demonstrates how to use the plug-in to insert, query and delete a file from a specified bucket.

The process is designed with the following activities:



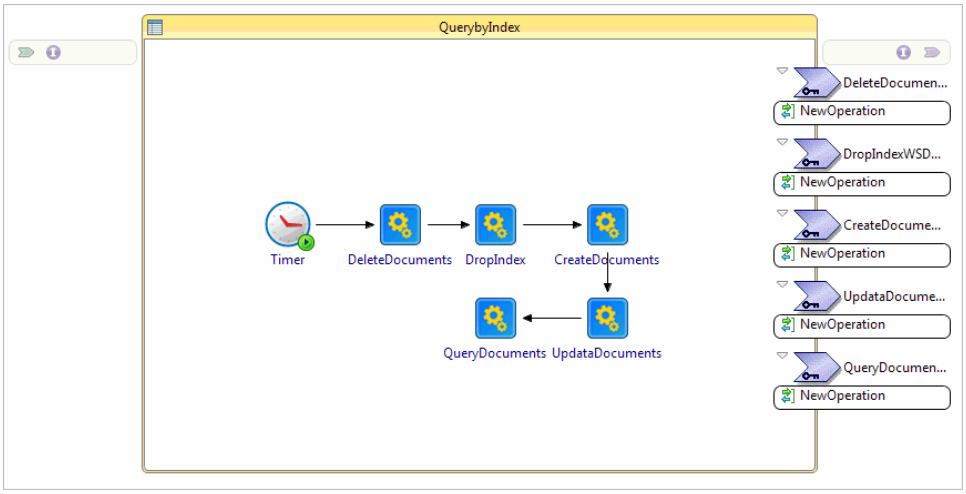
The following table describes the activities in the process.

Activity	Description
Timer	Starts the process.
CreateGridFSFile	Invokes the CreateGridFSFile subprocess to insert a file into the specified bucket and specify a new name for the inserted file.
QueryGridFSPhotos	Queries the inserted file from the specified bucket.
JavaCode	Obtains an instance of InputStream and writes the file stream into the inserted file.
DeletePhotos	Invokes the DeletePhoto subprocess to delete the inserted file from the specified bucket.

QuerybyIndex.bwp

This process demonstrates how to use the plug-in to query documents from a specified collection using a newly created index.

The process is designed with the following activities:



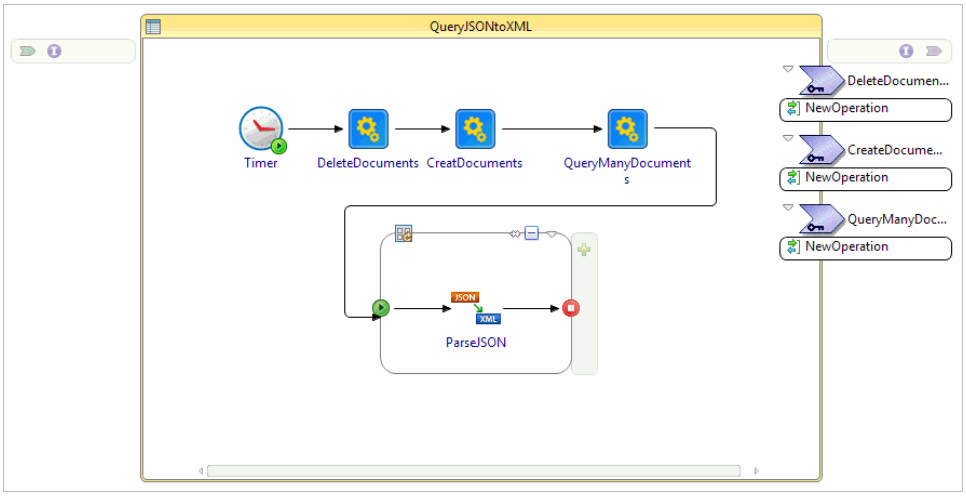
The following table describes the activities in the process.

Activity	Description
Timer	Starts the process.
DeleteDocument	Invokes the DeleteDocument subprocess to delete all documents from the specified collection.
DropIndex	Invokes the DropIndex subprocess to delete all indexes from the specified collection.
CreateDocument	Invokes the CreateDocument subprocess to insert a group of documents into the specified collection and create a new index for the inserted documents.
UpdateDocument	Invokes the UpdateDocument subprocess to update a document in the specified collection.
QueryDocument	Invokes the QueryByIndex subprocess to query the updated document from the specified collection using the newly created index.

QueryJSONtoXML.bwp

This process demonstrates how to use the plug-in to parse JSON documents into XML documents.

The process is designed with the following activities:



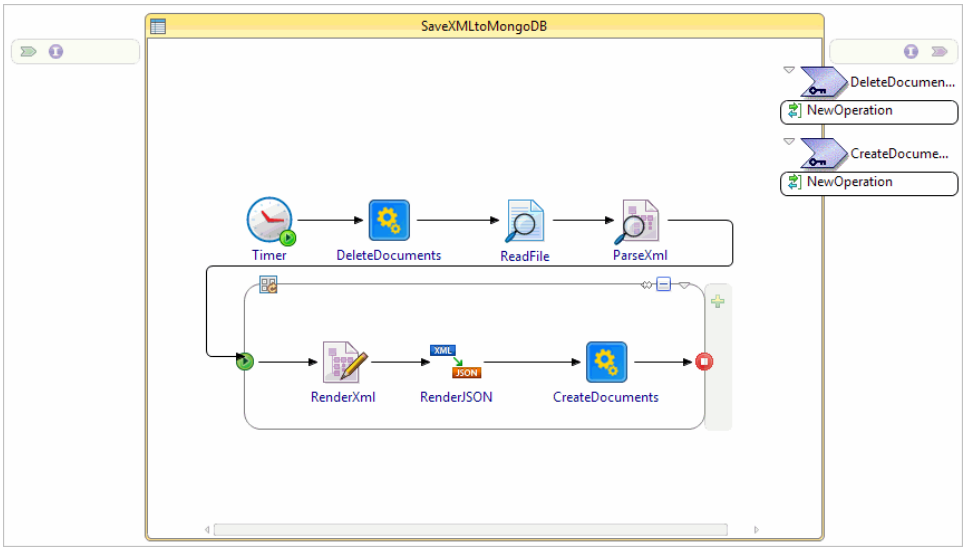
The following table describes the activities in the process.

Activity	Description
Timer	Starts the process.
DeleteDocuments	Invokes the DeleteDocument subprocess to deletes documents from the specified collection.
CreateDocuments	Invokes the CreateDocument subprocess to insert documents into the specified collection.
QueryDocuments	Invokes the QueryDocument subprocess to query documents from the specified collection.
ParseJSON	Parses the JSON documents into XML documents.

SaveXMLtoMongoDB.bwp

This process demonstrates how to use the plug-in to save XML documents into MongoDB.

The process is designed with the following activities:



The following table describes the activities in the process.

Activity	Description
Timer	Starts the process.
DeleteDocuments	Invokes the DeleteDocument subprocess to deletes documents from the specified collection.
ReadFile	Reads the specified XML file and converts it to a string.
ParseXml	Parses the string and converts it into an `XML` format file.
RenderXml	Generates a new XML file.
RenderJSON	Converts the newly generated XML file into a JSON file.
CreateDoucments	Invokes the CreateDocument subprocess to insert documents into the specified collection.

Managing Logs

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 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

You can configure a different log level for the plug-in, plug-in activities and shared resources 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 loggers** area to specify a log level for the plug-in:
   ```
   <logger name="com.tibco.bw.palette.mongodb.runtime">
      <level value="DEBUG"/>
   </logger>
   ```
   The value of the **`level`** element can be `Error`, `Info`, or `Debug`.

   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 different nodes in the **BusinessWorks Palette and Activity loggers** area to specify different log levels for the activities and shared resources.
   For the MongoDB Connection shared resource, add the following node:
   ```
   <logger name="com.tibco.bw.sharedresource.mongodb.runtime">
      <level value="DEBUG"/>
   </logger>
   ```

For the EventListener and the WaitForEvent activities, add the following node:

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

For other activities, add the following node:

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

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.

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-mongodb.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 exceptions that are thrown by the plug-in are listed with their corresponding descriptions and solutions.

Error Code and Error Message	Role	Category	Description	Solutions
TIBCO-BW-PALETTE-MONGODB-500001 `Activity {0} initialization error.`	Error	BW-Plug-in	Occurs when the activity is initiated.	Ensure that the activity is configured correctly.
TIBCO-BW-PALETTE-MONGODB-500002 `IOException occurred while retrieving XML Output for activity [{0}].`	Error	BW-Plug-in	Occurs when the XML output for the activity is retrieved.	Ensure that the activity is configured correctly.
TIBCO-BW-PALETTE-MONGODB-500003 `Exception occurred while invoke execute method for activity [{0}].`	Error	BW-Plug-in	Occurs when the execute method for the activity is invoked.	Ensure that the activity is configured correctly.
TIBCO-BW-PALETTE-MONGODB-500004 `Configuration Error.{0}`	Error	BW-Plug-in	Occurs when the configuration for the activity is not correct.	Ensure that the activity is configured correctly.
TIBCO-BW-PALETTE-MONGODB-500005 `Exception occurred while operate the activity wrong.{0}`	Error	BW-Plug-in	Occurs when the activity is operated incorrectly.	Ensure that the activity is configured correctly.

Error Code and Error Message	Role	Category	Description	Solutions
TIBCO-BW-PALETTE-MONGODB-500006 `Exception occurred while operate the function of IsGridFs.{0}`	Error	BW-Plug-in	Occurs when the function of IsGridFs is operated incorrectly.	Ensure that the activity is configured correctly.
TIBCO-BW-PALETTE-MONGODB-500007 `There is a problem with network.`	Error	BW-Plug-in	Occurs when there is a problem with network .	Ensure that the network works correctly.
TIBCO-BW-PALETTE-MONGODB-500008 `There is a problem with IO.`	Error	BW-Plug-in	Occurs when the IO operation is not correct.	Ensure that the IO is operated correctly.
TIBCO-BW-PALETTE-MONGODB-500009 `There is a problem with the cursor.`	Error	BW-Plug-in	Occurs when the cursor operation is not correct.	Ensure that the query is operated correctly.
TIBCO-BW-PALETTE-MONGODB-500010 `Valid output type are : {REPLACE, MERGE, REDUCE, INLINE}`	Error	BW-Plug-in	Occurs when the input for the MapReduce activity is not correct.	Ensure that the MapReduce activity is operated correctly.
TIBCO-BW-PALETTE-MONGODB-500011 `Exception occurred while parsing the JSON string {0}, please ensure your input is correct.`	Error	BW-Plug-in	Occurs when the input is not correct.	Ensure that your input is correct.
TIBCO-BW-PALETTE-MONGODB-500012 `{0}`	Error	BW-Plug-in	Occurs when the listening operation is not correct.	Ensure that your host is correct.