



# TIBCO ActiveMatrix BusinessWorks™

## Getting Started

*Version 6.7*

*November 2020*

*Document Updated: December 2020*



# Contents

---

<b>Figures</b> .....	<b>4</b>
<b>TIBCO Documentation and Support Services</b> .....	<b>5</b>
<b>Changing Help Preferences</b> .....	<b>6</b>
<b>About the Getting Started Guide</b> .....	<b>7</b>
<b>Orientation</b> .....	<b>8</b>
TIBCO Business Studio™ for BusinessWorks™ .....	8
Application Development .....	8
Web Services .....	9
Shared Resources .....	10
Cheat Sheet .....	10
REST Support .....	11
REST Docucenter and Tester .....	12
Creating REST Services in the Rest Service Wizard .....	13
Discovering API Models from TIBCO Business Studio™ for BusinessWorks™ .....	14
Importing an API Model into your Workspace .....	15
Creating an XML Schema for a Swagger File .....	17
Consuming a REST Endpoint in TIBCO Business Studio for BusinessWorks .....	17
Synchronizing the Imported REST API Models in TIBCO Business Studio™ for BusinessWorks™ .....	18
Archive Files .....	18
Debugger .....	19
Run Time .....	20
Deployment .....	21
<b>File Poller Tutorial</b> .....	<b>22</b>
Creating a New Project .....	22
Configuring the FilePoller Project .....	23
Testing the FilePoller Application in the Debugger .....	25
<b>REST Service Tutorial</b> .....	<b>27</b>
Installing PostgreSQL .....	27
Creating a New Process .....	28
Building a REST Service .....	30
Testing the REST Service .....	35
Testing the POST and GET Operations .....	39
Troubleshooting .....	40
<b>REST Reference Tutorial</b> .....	<b>42</b>
<b>Administration Tutorial</b> .....	<b>45</b>
Running Admin Sample Scripts .....	45

Testing the Deployed RESTful BookStore Application from Admin UI ..... 49

Defining and Deploying Multiple AppSpaces and AppNodes ..... 53

Core Admin Sample Scripts ..... 54

**Legal and Third-Party Notices ..... 70**

# Figures

---

Shared Resources Folders in Project Explorer .....	10
REST Service Wizard .....	12
Swagger UI .....	12
REST Service Wizard .....	13
Export Project Dialog .....	19
Drag and Drop Project to File explorer .....	19
Debug Perspective .....	20
Console View .....	20
Stop Icon in Debug Perspective .....	20
Console View .....	21
Utils Package .....	46
Net Package .....	46

# TIBCO Documentation and Support Services

---

## How to Access TIBCO Documentation

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

The TIBCO Product Documentation website is updated frequently and is more current than any other documentation included with the product. To access the latest documentation, 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.

Access the following *TIBCO ActiveMatrix BusinessWorks™* guides on the TIBCO Documentation site:

- Concepts
- Installation
- Getting Started
- Application Development
- Administration
- Bindings and Palettes Reference
- Samples
- Error Codes
- Migration
- Performance Benchmarking and Tuning
- REST Reference Guide
- Refactoring Best Practices

## How to Contact TIBCO Support

You can contact TIBCO Support in the following ways:

- For an overview of TIBCO Support, visit <http://www.tibco.com/services/support>.
- For accessing the Support Knowledge Base and getting personalized content about products you are interested in, visit the TIBCO Support portal at <https://support.tibco.com>.
- For creating a Support case, you must have a valid maintenance or support contract with TIBCO. You also need a user name and password to log in to <https://support.tibco.com>. If you do not have a user name, you can request one by clicking Register on the website.

## How to Join TIBCO Community

TIBCO Community is the official channel for TIBCO customers, partners, and employee subject matter experts to share and access their collective experience. TIBCO Community offers access to Q&A forums, product wikis, and best practices. It also offers access to extensions, adapters, solution accelerators, and tools that extend and enable customers to gain full value from TIBCO products. In addition, users can submit and vote on feature requests from within the [TIBCO Ideas Portal](#). For a free registration, go to <https://community.tibco.com>.

# Changing Help Preferences

---

By default, documentation access from TIBCO Business Studio™ for BusinessWorks™ is online, through the [TIBCO Product Documentation](#) website that contains the latest version of the documentation. Check the website frequently for updates. To access the product documentation offline, download the documentation to a local directory or an internal web server and then change the help preferences in TIBCO Business Studio for BusinessWorks.

## Prerequisites

Before changing the help preferences to access documentation locally or from an internal web server, download the documentation.

1. Go to <https://docs.tibco.com/>
2. In the **Search** field, enter TIBCO ActiveMatrix BusinessWorks™ and press **Enter**.
3. Select the TIBCO ActiveMatrix BusinessWorks™ product from the list. This opens the product documentation page for the latest version.
4. Click **Download All**.
5. A compressed .zip file containing the latest documentation is downloaded to your web browser's default download location.
6. Copy the .zip file to a local directory or to an internal web server and unzip the file.

To point to a custom location:

## Procedure

1. Perform one of the following steps based on your operating system:
  - On Windows OS: In TIBCO Business Studio for BusinessWorks, click **Window > Preferences**
  - On macOS: In TIBCO Business Studio for BusinessWorks, click **TIBCO Business Studio > Preferences**.
2. In the Preferences dialog box, click **BusinessWorks > Help**.
3. Click **Custom Location**, and then browse to the `html` directory in the folder where you extracted the documentation or provide the URL to the `html` directory on your internal web server.
4. Click **Apply**, and then click **OK**.

# About the Getting Started Guide

---

This guide contains tutorials that are designed to familiarize you with a representational set of activities you might use to develop an application. By referring to these simple tutorials, you can understand how to use within each phase of the project life cycle.

These tutorials illustrate the basic activities for creating an application, building and testing a simple REST service, and basic information on deploying the Administration sample applications using the provided scripts.

For more information, see one of the following topics:

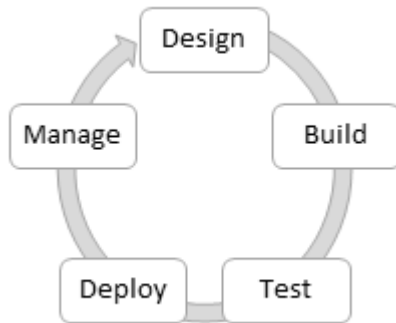
- [File Poller Module Tutorial](#): Guides you through creating a simple process and running it.
- [REST Service Tutorial](#): Walks you through the steps to build and test a simple REST Service using TIBCO Business Studio™ for BusinessWorks™ and the Swagger UI.
- [REST Reference Tutorial](#): Shows you how to create a simple REST Invoke to an existing REST Service defined by a Swagger specification.
- [Administration Tutorial](#): Provides information about the administration framework and deploying sample applications using the provided scripts.

The [Orientation](#) section introduces you to the TIBCO Business Studio for BusinessWorks development environment. Before you continue, read the *TIBCO ActiveMatrix BusinessWorks™ Concepts* guide to familiarize yourself with the ActiveMatrix BusinessWorks™ concepts and terminology.

## Orientation

TIBCO ActiveMatrix BusinessWorks™ is an integration product suite for enterprise, web, and mobile applications.

TIBCO Business Studio for BusinessWorks allows you to create services and integrate applications using a visual, model-driven development environment, and then deploy them in the ActiveMatrix BusinessWorks™ runtime environment.



This product uses Eclipse-based graphical user interface (GUI) provided by TIBCO Business Studio for BusinessWorks to define business processes and generate deployable artifacts in the form of archive files. These deployable artifacts can be:

- deployed and run in the product runtime, and
- managed using the console or bwadmin, or the web-based Admin UI.

## TIBCO Business Studio™ for BusinessWorks™

TIBCO Business Studio for BusinessWorks is the design-time IDE (based on Eclipse) where you create and test your processes.

You use TIBCO Business Studio for BusinessWorks for end-to-end application development. You can create new services, orchestrate business process, and integrate applications in a short time. A model-driven development approach is supported, with a rich set of palettes for process design. These palettes can be used to visually create and test business processes that connect to various technologies such as a database, messaging servers, and so on.

TIBCO Business Studio for BusinessWorks is installed as part of ActiveMatrix BusinessWorks™. To open TIBCO Business Studio for BusinessWorks:

- On Unix: Run the TIBCO Business Studio for BusinessWorks executable located in the `$TIBCO_HOME/studio/<version>/eclipse/` directory.
- On Windows: Click **Start > All Programs > TIBCO > TIBCO\_HOME > TIBCO Business Studio <version\_number> > Studio for Designers**

In the **Workspace Launcher** dialog, accept the default workspace or browse to create a new workspace, and click **OK**.

When TIBCO Business Studio for BusinessWorks starts, the default development environment, a *workbench*, is displayed.

For more information about samples in TIBCO Business Studio for BusinessWorks, see the section *Accessing Samples* in the *TIBCO ActiveMatrix BusinessWorks™ Samples* guide.

## Application Development

Applications solve integration problems of varying complexity. Using TIBCO Business Studio™ for BusinessWorks™, applications can be developed using an application-oriented integration style or a



service-oriented integration style. How you design your application's integration style depends on the following factors:

- Speed of integration
- Data abstraction
- Richness of operation primitives
- Typical endpoints

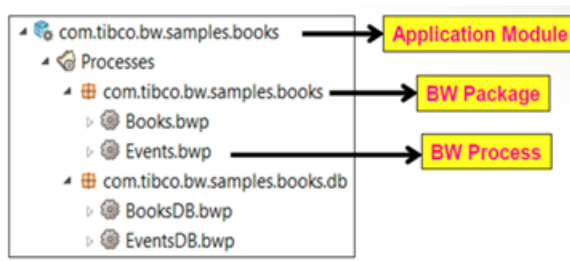
For more information about an application's integration style and other application design considerations, see "Application Design Considerations" in the *TIBCO ActiveMatrix BusinessWorks™ Application Development* guide.

Processes allow you to implement business logic that can obtain and manage the flow of information in an enterprise between a source and different destinations. In process-driven design, the business processes or integration flows are first realized and captured. For more information about process design, see "Process Design Considerations" in the *TIBCO ActiveMatrix BusinessWorks™ Application Development* guide.

Processes developed in TIBCO Business Studio for BusinessWorks are saved in an application module. Application modules are equivalent to projects and are saved to folders on the disk. The TIBCO Business Studio for BusinessWorks workspace contains one or more application modules.

- An application module contains one or more packages
- A package contains one or more processes, which in turn are main processes or subprocesses
- A process is stored as a single file with a .bwp extension

An application module contains a special folder called **Processes**. This folder contains the user created processes. In addition, an application module also contains folders to store WSDL files, schemas, and shared resources. The **Processes** folder is shown below.



A package should follow the Java naming convention.

Processes are designed in the **Process Editor**. Activities and shared resources help you rapidly design business processes. An activity is the individual unit of work in a process. There are multiple ways to add an activity to a process: from the right-click menu on the **Process Editor**, from the palettes, and from the **File Explorer** or **Project Explorer**. To add an activity from the palette, select it and drop it on the **Process Editor**.

Implemented services are shown as chevrons on the left side of the **Process Editor**. References that are invoked are shown on the right side of the **Process Editor**. For a simple process, services and references are optional.

## Web Services

Web services are application components that communicate using open standard protocols. You can develop SOAP-based web services using the Generate Concrete WSDL Wizard. The wizard generates a WSDL file and the appropriate response activities. You can develop REST-based web services using the REST Service Wizard in TIBCO ActiveMatrix BusinessWorks™.

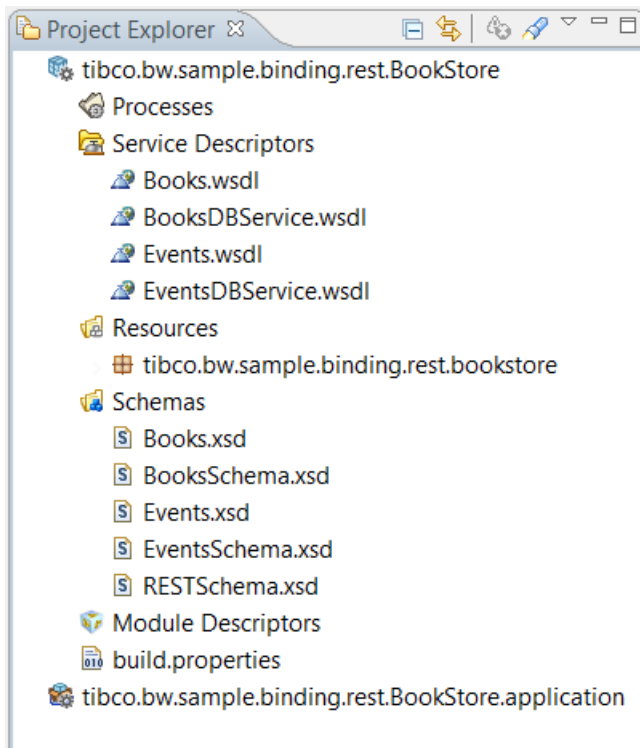
Select a WSDL file in the **Project Explorer** and drop it on the **Process Editor** to implement a web service. Dropping the WSDL file displays a menu for creating services or implementing operations. Response activities are automatically generated.

To create a REST service, select a path under the .json file in the **Service Descriptors** folder and drop it on the **Process Editor** to implement a web service. When you drop the path, it displays a menu with an option to create a service or a reference.

## Shared Resources

Shared resources are configurations that are shared among activities. These are resources such as database, JMS and HTTP connections, and connections to other servers. Resources are added to special folders in the **Project Explorer**. The following image shows these folder in the **Project Explorer**.

### *Shared Resources Folders in Project Explorer*



The following types of folders for shared resources can exist in a project.

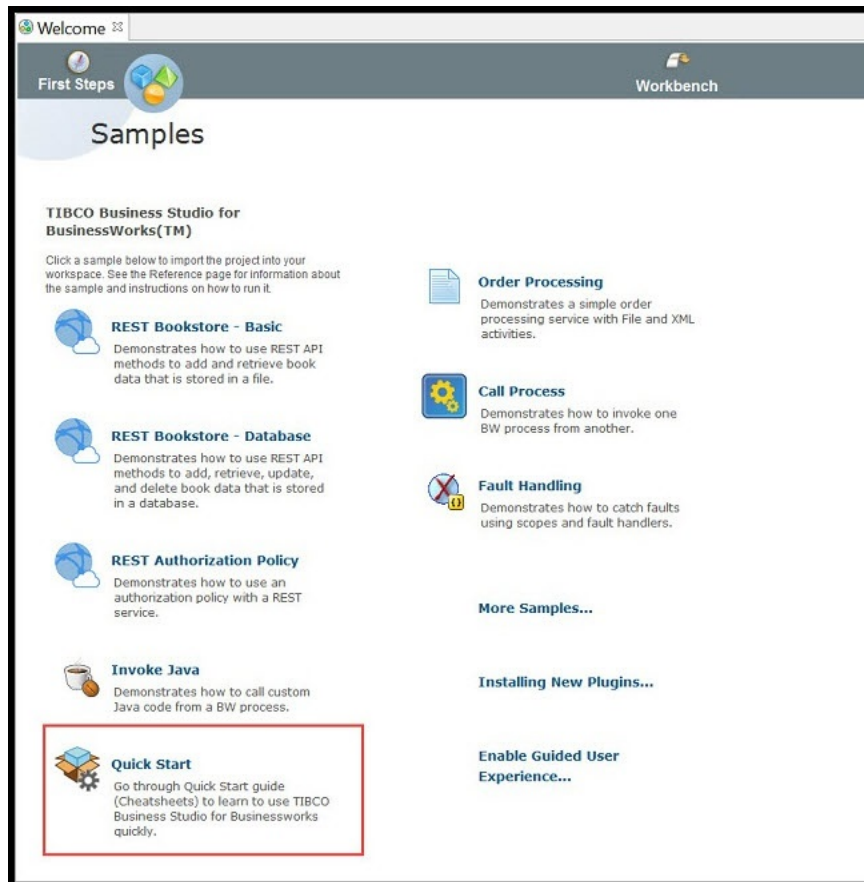
- **Resources:** Contains shared resources used by activities in a process.
- **Schemas:** Stores XSD (schema) files.
- **Service Descriptors:** Stores WSDL and JSON files.

## Cheat Sheet

A quick reference help is available in the form of cheat sheets for few common tasks in TIBCO Business Studio™ for BusinessWorks™.

Open a cheat sheet using any one of the following ways:

- On the Welcome page, click the Quick Start link. If you open TIBCO Business Studio for BusinessWorks in a new workspace, the Welcome screen is displayed. If you open ActiveMatrix BusinessWorks in an existing workspace, open the Welcome page by using **Help > Welcome**.



- Click **Help** > **Cheat Sheets**.

Cheat sheets are available for the following tasks:

- Adding unit testing to a project.
- Creating a basic SOAP application.
- Creating an application using Java Invoke.
- Configuring a JMS connection and creating a JMS send message application.
- Configuring a JDBC connection using custom driver.
- Developing a REST API to expose a SOAP service.
- Exposing a SOAP service via REST API.
- Using security policies in a project.

## REST Support

The REST Service wizard is used to build RESTful services.



When you create a REST service, make sure to edit the **Default Host** field in the HTTP Connection Resource to reflect the actual host name. By default, the Default Host field is set to localhost.

## REST Service Wizard

Developing a RESTful service is a simple three step process:

1. Name the REST resource.
2. Choose the resource definition (the XSD schema).
3. Choose the REST operations to implement.

The input and output messages for the operations are automatically generated along with a Response activity. An HTTP shared resource is also generated with the default configuration. You can then add activities and implement the business logic for each operation in the process.

## REST Documenter and Tester

A REST documenter and tester is automatically generated for a REST resource. The documentation is based on the Swagger specification and is rendered using the Swagger UI.

### Swagger UI

This Swagger based interface provides a convenient and quick way to:

- View REST endpoints and operations implemented by the REST resource service.
- Examine the inputs and outputs for each operation in JSON format.

- Enable **Input** fields to specify JSON or XML input for each operation.
- Invoke an operation and receive a live response for the input supplied.

## Creating REST Services in the Rest Service Wizard

You use the REST Service Wizard to create a RESTful service or simply drag and drop a process to the left of the **Process Editor** to create a REST service.



When you create a REST service, make sure to update the **Default Host** field in the HTTP Connection Resource to reflect the actual host name. By default, the **Default Host** field is set to `localhost`.

Follow these steps to create a REST service using the REST Service Wizard:

### Procedure

1. Right-click on a process in the **Project Explorer** and select **New > BusinessWorks REST Resource**. The REST Service Wizard opens.

#### *REST Service Wizard*

2. Specify a name for the REST resource.
3. Choose the resource definition (the XSD schema).
4. Select the REST operations to implement.
5. Click **Next** to configure the operations or click **Finish**.

## Discovering API Models from TIBCO Business Studio™ for BusinessWorks™

To view the APIs that reside on your local machine or on a remote server, use the **API Explorer** view in the TIBCO Business Studio for BusinessWorks.

### Prerequisites

For the API Explorer to discover the APIs residing on a remote server, the remote server must be up and running.

You can set up the locations to which you want the API Explorer to connect and look for the APIs. To do so, follow the steps below.

### Procedure

1. In TIBCO Business Studio for BusinessWorks, go to the **API Explorer** view.
2. In the **API Explorer** tab, click the **View Menu** downward-facing triangle icon (▾) and select **Settings**. The Settings dialog box is displayed.

The registries for the ActiveMatrix BusinessWorks - API Modeler and the samples folder installed on your local machine are configured and appear in the API registry configurations box by default. In this dialog, you can specify how the discovered APIs appear in the API Explorer:

- **API Presentation** - specifies how the APIs appear in the **API Explorer**
  - Flat** - displays the APIs as a flat list with each API's version number displayed next to its name in parenthesis. If there are multiple versions of the same API, each version is shown as a separate API, hence multiple APIs with the same name but different version numbers.
  - Hierarchical** - displays every API as a hierarchy of API name label with version number folder under it and the actual API under the version folder. If there are multiple versions for an API, each version is listed in its own separate folder under the API name label.
  - Latest Version** - displays only the latest version of the API, even though there are multiple versions available.
- **Group by API registry** - groups the APIs according to the registry from which they were discovered
- **API registry configurations** - displays the list of API registries that are currently configured in your TIBCO Business Studio for BusinessWorks installation.

Select the API registry check boxes to display the APIs.







You can edit an existing registry by clicking the **Edit** button, delete the registry configuration by clicking **Remove**, or changing the order in which the registries show up in the API Explorer by using the **Up** and **Down** button. These button get activated when you click on an API registry name.

3. Click **New** to add a new registry.
4. In the **Create new API Registry client configuration** dialog do the following:
  - a) Enter a name for the API registry that you are mapping to in the **Name** text box.
  - b) Select the **Local** radio button to map a location where the APIs are stored on your local machine's hard drive and navigate to the location using the **Browse** button. Alternatively, select the **Remote** radio button if you want to map to a remote server that contains the APIs and enter the URL for the server in the **URL** text box.
5. Click **Finish**.  
You should now see the APIs displayed in the **API Explorer** in the format that you specified in the Settings dialog box. Expanding an API show you its version, the resource path, and the operations to perform on that resource.



Organizations can have multiple owners, and a list of owners is displayed in the Edit API Registry client configuration page.

The **API Explorer** view has the following quick-access buttons that you can use to format the way the APIs are listed:

-  **Refresh**
-  **Expand All**
-  **Collapse All**
-  **Group by API Registry**
-  **API Presentation**
-  **API Registries.** Selecting a registry from this drop-down list toggles between displaying and hiding the registry in the **API Explorer**.

Use the search filter that appears at the bottom of the **API Explorer** view to search for API names that match the string that you enter in the **Filter** text box. You can search by typing in the version number, the full API name, or a full word within an API name. Wildcards are not supported. The search is case insensitive.

## Importing an API Model into your Workspace

The APIs that are discovered from local and remote servers are displayed in the **API Explorer** tab of the TIBCO ActiveMatrix BusinessWorks™. You can use these APIs in your project by importing them into the **Service Descriptors** folder of the project. The .json file for the API gets copied into the application module.

To import the APIs from the **API Explorer** into your project follow these steps.

### Procedure

1. Right-click on one or more API names in the **API Explorer** and select **Import**. The Import API dialog opens.

**Import API**

Create a new project and import API to the new project.

Create a new project and import API to the new project.  
 Import to project:

API list to import:

API	
<input checked="" type="checkbox"/> TIBCO Cloud Integration PetStore Sample (1.1)	

Every API you selected in the **API Explorer** is listed in this dialog. If an API has multiple versions, all versions are listed. By default, all APIs listed here are selected. You can deselect APIs that you do not want to import by clearing its check box.

2. Select the appropriate action and click **Next**.

Option	Description
<b>Import to project</b>	Select the radio button to import the API into an existing project and browse to the project using the <b>Browse</b> button.
<b>Create a new project and import API to the new project</b>	To create a new project and import the API into that project select the radio button.
<b>API list to import</b>	Select the API or the appropriate version of the API when there are multiple versions of the API available.

The Change Swagger name dialog box opens.

Change the swagger file name if required. Click **Next**.

The New BusinessWorks Application Module dialog box opens.

3. Create a new application module with appropriate details and click **Finish**. You should see the API(s) under the **Service Descriptors** folder of the project. You can create sub-folders under the **Service Descriptors** folder and drag-and-drop APIs into them if you prefer to organize the APIs into a meaningful folder structure.

As an alternative to the above procedure, you can also drag and drop the API from the **API Explorer** into the project's **Service Descriptors** folder.



APIs that were created using a Swagger file must be implemented exactly as defined by the Swagger file. TIBCO Business Studio for BusinessWorks allows you to only view the parameters and operations that are defined in the Swagger file. You cannot create any new parameters or operations for such applications.



## Creating an XML Schema for a Swagger File

TIBCO Business Studio™ for BusinessWorks™ supports the creation of an XML schema for an imported Swagger 2.0 or a Swagger 3.0 file.

You can create an XML schema for a Swagger 2.0 or a Swagger 3.0 files in one of two ways described below.

### Prerequisites

A Swagger 2.0 or a Swagger 3.0 file must exist in the **Service Descriptors** folder of the project. Make sure to import the Swagger file into the **Service Descriptors** folder before you follow these steps:

### Procedure

1. Drop the Swagger file on the right side of the canvas to create a REST service binding. This action generates an XML schema for the Swagger file under the **Schemas** folder. The XML schema file has the same name as the Swagger file.  
Or
2. Right-click the Swagger file in the **Service Descriptors** folder and select **Refactor > Generate XSD Schema**.
  - To see which XML schema is related to the Swagger file, right-click the Swagger file and select **Refactor > Open XSD Schema**.
  - If you have multiple Swagger files all of which contain a definition for the same object, the definition for the object in all the Swagger files must be identical.
  - If you have multiple Swagger files with one file (a master file) containing a super set of definitions contained in the other files, generate an XSD file from the master Swagger file that contains the super set, and create links to the other files in the master Swagger file. If you create a link to the super set file in one of the subset files and then create an XSD from the subset file, then the XSD contains only those elements that are common to both files. It does not contain elements for definitions that exist only in the super set file.

## Consuming a REST Endpoint in TIBCO Business Studio for BusinessWorks

Create a REST Reference binding to consume a REST endpoint.



You cannot edit the REST reference binding configuration for APIs that are imported from a source external to TIBCO Business Studio for BusinessWorks.

### Prerequisites

Swagger API documents must be imported into the project's **Service Descriptors** folder. This gives you the ability to expand and collapse endpoints, operations, parameters and response codes in the **Project Explorer**.

To consume a REST API that exists in the **Service Descriptors** of the project, do the following:

### Procedure

1. Expand the Swagger file in the **Service Descriptors** special folder to view the endpoints, operations, parameters, and response codes.
2. Drag an endpoint on the right side of the canvas to create a REST reference binding. This creates a cloud shaped icon with a right facing arrow. The cloud is an indication that it is a REST reference whereas the arrow within the cloud indicates that it is a binding. Since the binding is within a cloud, it is an indication that it is a REST binding. You cannot convert a REST binding to a SOAP binding or vice versa.



When you create a REST reference for the service, make sure to edit the **Default Host** field in the HTTP Client Resource to reflect the actual host name. By default, the **Default Host** field is set to localhost.

3. Drag and drop an operation from the REST reference binding on to the canvas. This creates an **Invoke** activity which is pre-configured to invoke the operation. It also creates an HTTP Client Shared Resource with the host name and port number. The configuration for these entities is copied from the Swagger document from which you created the reference binding. The reference consists of the name of the API as well as the operations it supports.

When invoking a POST or PUT method, you must provide the request string in the **Input** tab. To do so, click the column next to **item** under **postRequest** in **XPath Expression** and provide the request string in the dropdown box.

4. Test the configured process using the TIBCO Business Studio for BusinessWorks debugger.

## Synchronizing the Imported REST API Models in TIBCO Business Studio™ for BusinessWorks™

If a REST service developer has made changes to the service API after creating the service, the changes need to be propagated to all the places where the service is used. You can check for updates to a Swagger file that has been imported into TIBCO Business Studio for BusinessWorks. The icon to the left of the Swagger file in the **Project Explorer** in the TIBCO Business Studio for BusinessWorks displays an indication that the file has been modified in its original location and the local copy of the file is not in synchronization with its source.

You can check for differences between the original Swagger file and its copy that was created when importing it into the TIBCO Business Studio for BusinessWorks. You can also compare the differences between the two and update your local copy if need be. To do so, follow these steps:

### Procedure

1. Right-click the Swagger file under **Service Descriptors** in the **Project Explorer**.

2. Select **Remote Interface**.

The **Check for Differences** menu option checks for differences between the imported copy and its original.

The **Compare Differences** menu option first checks for differences between the imported copy of the Swagger file and its original. If there is a difference, the file appears in the **Synchronize** tab and if you double-click it there it displays the two files side by side with the differences highlighted.

The **Update Local Copy** menu item updates the copy of the file in your workspace to match its original. It also regenerates the schema.



No changes are performed for processes that have already been created.

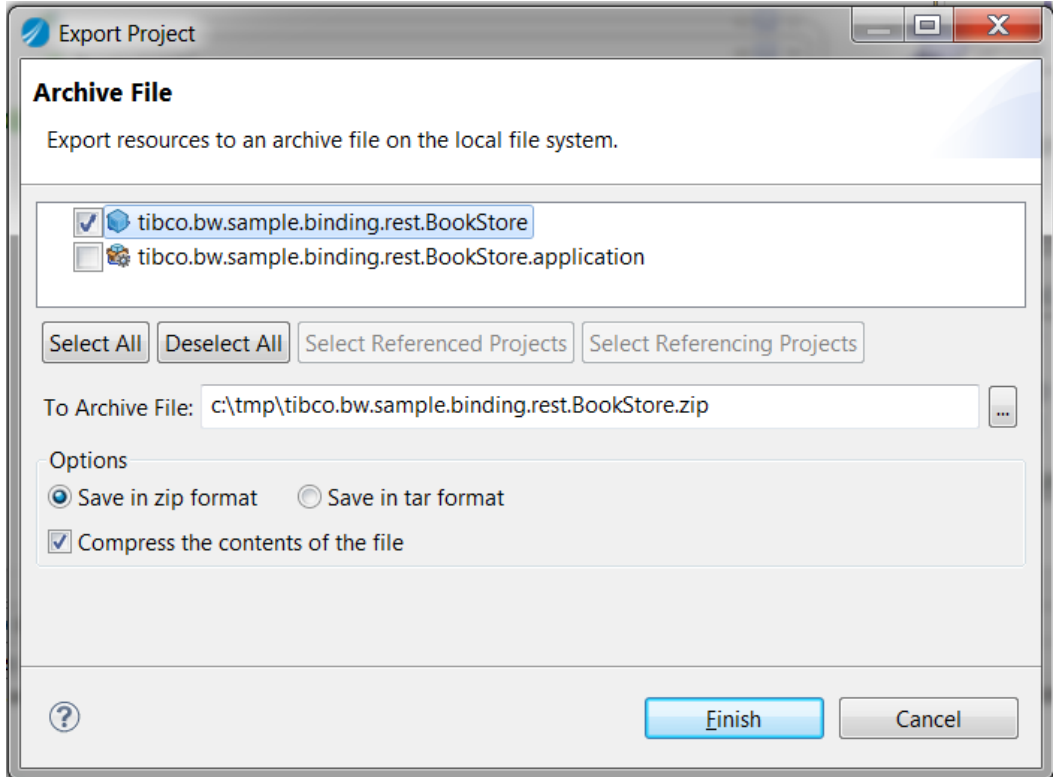
## Archive Files

After completing an application module, you must define an application to build a deployment archive file. An application defines all the processes, properties, and resources that must be included in the archive file. By default, all processes are included.

To create an archive, choose one of the following:

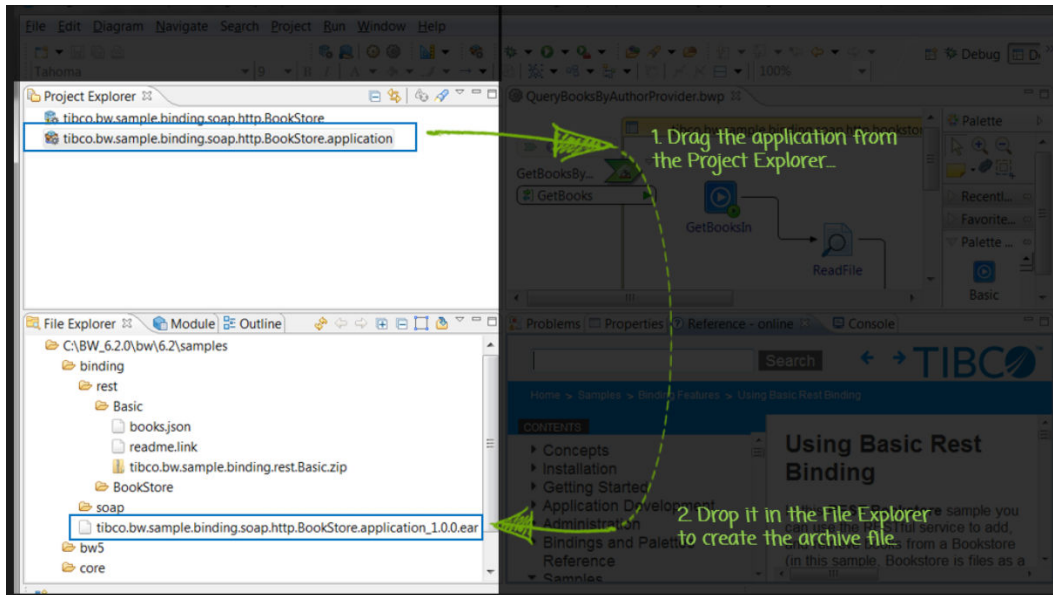
- Right-click the project in the **Project Explorer** and choose **Export > Studio Projects to Archive**. The **Export Project** dialog is displayed.

## Export Project Dialog



- Drag the project from the **Project Explorer** and drop it on a folder in the **File Explorer**.

### Drag and Drop Project to File explorer



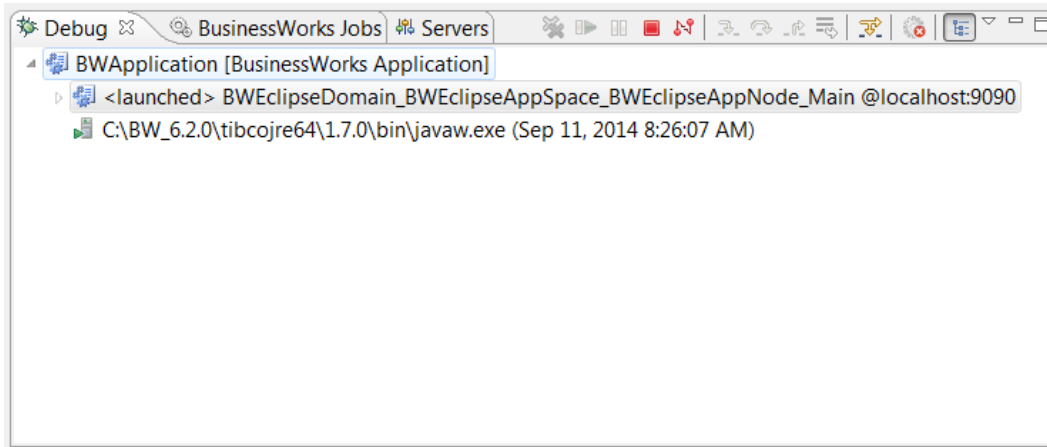
In both scenarios, an archive file is created with all required processes, properties, and resources. In the first scenario, you can name the archive file, select the format, and select the resources to include. In the second scenario, the archive is created for you in the format appropriate for your operating environment. All required elements are included.

## Debugger

The TIBCO Business Studio for BusinessWorks debugger is used to test processes during the process development stage. By default, the debugger supports all the processes and subprocesses of an application

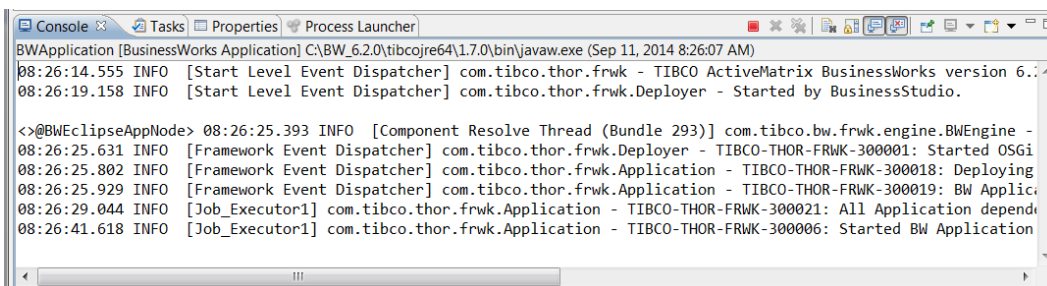
module, shared module and nested shared module. Starting the debugger brings up the **Debug** perspective. This perspective can be used to set breakpoints, step through processes, examine job variables, and activity input/output at each step.

### Debug Perspective



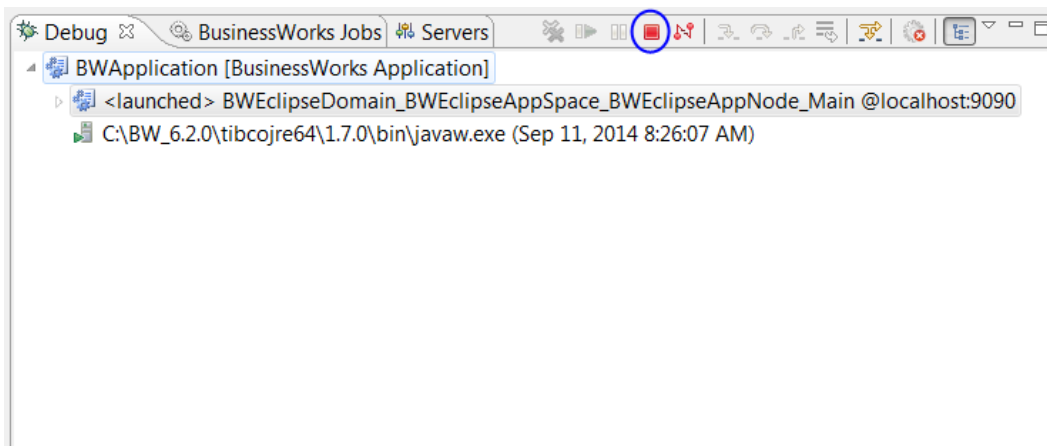
The **Console** view displays the messages and errors returned by the runtime.

### Console View



Start the debugger with the **Run > Debug** command. To stop the debugger, press the **Stop** icon on the **Debug** perspective toolbar:

### Stop Icon in Debug Perspective



## Run Time

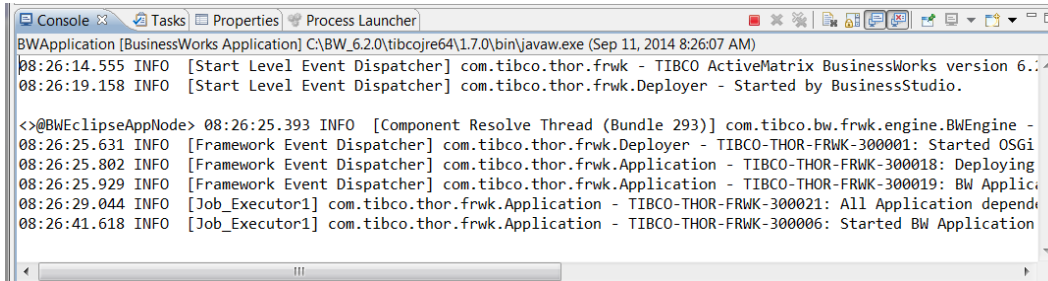
You can run applications in TIBCO Business Studio™ for BusinessWorks™ and test them in a runtime environment, which consists of a domain, AppSpace, and an AppNode on your local machine.

These runtime entities were created when you installed TIBCO ActiveMatrix BusinessWorks™. For more information about runtime entities, see the *TIBCO ActiveMatrix BusinessWorks™ Concepts* guide.

For information about the administration framework, see the *TIBCO ActiveMatrix BusinessWorks™ Administration* guide.

To run an application in TIBCO Business Studio for BusinessWorks, choose the **Run > Run** command. (Applications can also be run with the **Run > Run Configurations** command. This option allows you to manage and launch run configurations.) The **Run** command opens the **Console view** where progress messages and errors are displayed.


### Console View



```

Console [Tasks] Properties Process Launcher
BWApplication [BusinessWorks Application] C:\BW_6.2.0\tibcojre64\1.7.0\bin\javaw.exe (Sep 11, 2014 8:26:07 AM)
08:26:14.555 INFO [Start Level Event Dispatcher] com.tibco.thor.frwk - TIBCO ActiveMatrix BusinessWorks version 6.2.0
08:26:19.158 INFO [Start Level Event Dispatcher] com.tibco.thor.frwk.Deployer - Started by BusinessStudio.

<>@BWEclipseAppNode> 08:26:25.393 INFO [Component Resolve Thread (Bundle 293)] com.tibco.bw.frwk.engine.BWEngine -
08:26:25.631 INFO [Framework Event Dispatcher] com.tibco.thor.frwk.Deployer - TIBCO-THOR-FRWK-300001: Started OSGi
08:26:25.802 INFO [Framework Event Dispatcher] com.tibco.thor.frwk.Application - TIBCO-THOR-FRWK-300018: Deploying
08:26:25.929 INFO [Framework Event Dispatcher] com.tibco.thor.frwk.Application - TIBCO-THOR-FRWK-300019: BW Applic
08:26:29.044 INFO [Job_Executor1] com.tibco.thor.frwk.Application - TIBCO-THOR-FRWK-300021: All Application depende
08:26:41.618 INFO [Job_Executor1] com.tibco.thor.frwk.Application - TIBCO-THOR-FRWK-300006: Started BW Application
  
```

Click the **Businessworks Jobs** view in the top left to see the jobs created for the process. To stop the current job, click the **Stop** button  on the **Console view** toolbar.

From the **Console view**, you can use OSGi commands to monitor the running AppNode and gather metrics about your application. For information about OSGi commands, press Enter in the **Console view** to display the `<>@BWEclipseAppNode>` prompt. Type `help` to get a list of commands.

The scope is indicated along with the command. Commands with the scope `bw` return information about the running application. Type a command name followed by `-h` for information about the command. For example, the command `help bw:dsw` returns:

```

dsw - Diagnoses Shared Resource issues
  scope: bw
  parameters:
    String Partial or full name of a Shared Resource. Case is ignored.
  
```



Applications can also be run from the administration framework using the `bwadmin` command line utility or the Admin UI. For information about the administration framework, see the *TIBCO ActiveMatrix BusinessWorks™ Administration* guide.

## Deployment

Applications can be deployed from TIBCO Business Studio™ for BusinessWorks™ using either the **Deployment Wizard** or the **Deployment Server**.

The **Deployment Wizard** is available from the right-click menu once an archive file has been created. You need to provide the name and port for the network to which you want to deploy to, as well as the domain and AppSpace for deployment.

The Deployment Server is a pre-built deployment environment that consists of domains, AppSpaces, and AppNodes. Archive files have been uploaded to this server and applications are ready for deployment.

Network configuration for either a local network or the deployment server is available from the **Deploy view** in TIBCO Business Studio for BusinessWorks.

For more information about deployment in TIBCO Business Studio for BusinessWorks, see the *TIBCO ActiveMatrix BusinessWorks™ Application Development* guide.



Applications can also be deployed from the administration framework using the `bwadmin` command line utility or the Admin UI. For information about the administration framework, see the *TIBCO ActiveMatrix BusinessWorks™ Administration* guide.

# File Poller Tutorial

This tutorial guides you through creating a File Poller project. This simple project can perform many of the same tasks as those required for a project with a larger scope and more complexity. The File Poller project polls a directory for a specified file and writes a new file to the same directory each time the file is modified.

This tutorial involves:

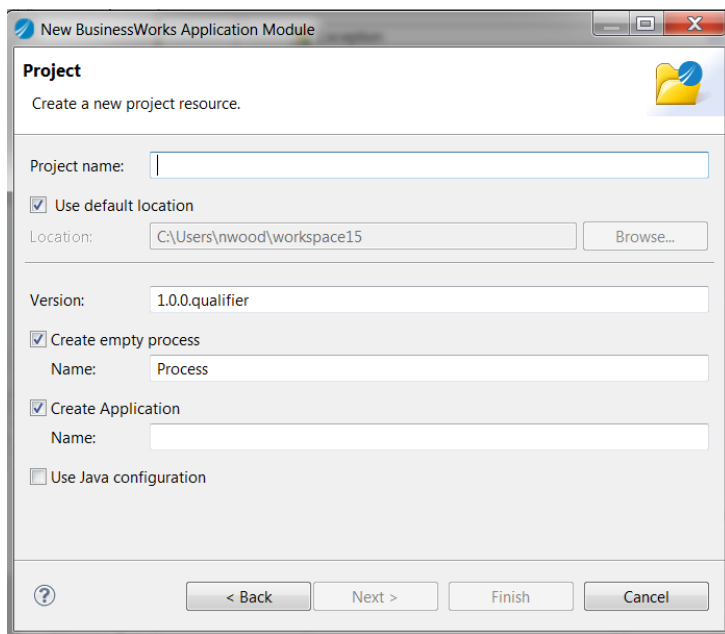
- [Creating a New Project](#)
- [Configuring the File Poller Project](#)
- [Testing the FilePoller Application in the Debugger](#)

## Creating a New Project

This section guides you through creating a simple project.

### Procedure

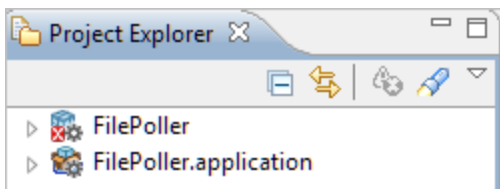
1. Start TIBCO Business Studio™ for BusinessWorks™:
  - a) On Unix: Select the TIBCO Business Studio for BusinessWorks executable located at `$TIBCO_HOME/studio/version/eclipse/`
  - b) On Windows: **Start > All Programs > TIBCO > TIBCO\_HOME > TIBCO Business Studio for Designers**
2. Launch the BusinessWorks Application Module wizard from **File > New > BusinessWorks Resources**. The BusinessWorks Resource Wizard opens.
3. In the Select a wizard dialog box, select **BusinessWorks Application Module** and click **Next**. The wizard is displayed:



4. In the **Project name** field, enter `FilePoller`  
Keep the **Use default location**, **Create empty process**, and **Create Application** check boxes selected.
5. Click **Finish**.

## Result

Two folders are created and are visible in the **Project Explorer**, one for the application and one for the application module.



## Configuring the FilePoller Project

The FilePoller project uses a **File Poller** activity and a **Write File** activity. This project creates a simple process that points to a specified file. The file is polled periodically to determine whether it was changed. The changed file comprises the text written to a new file.

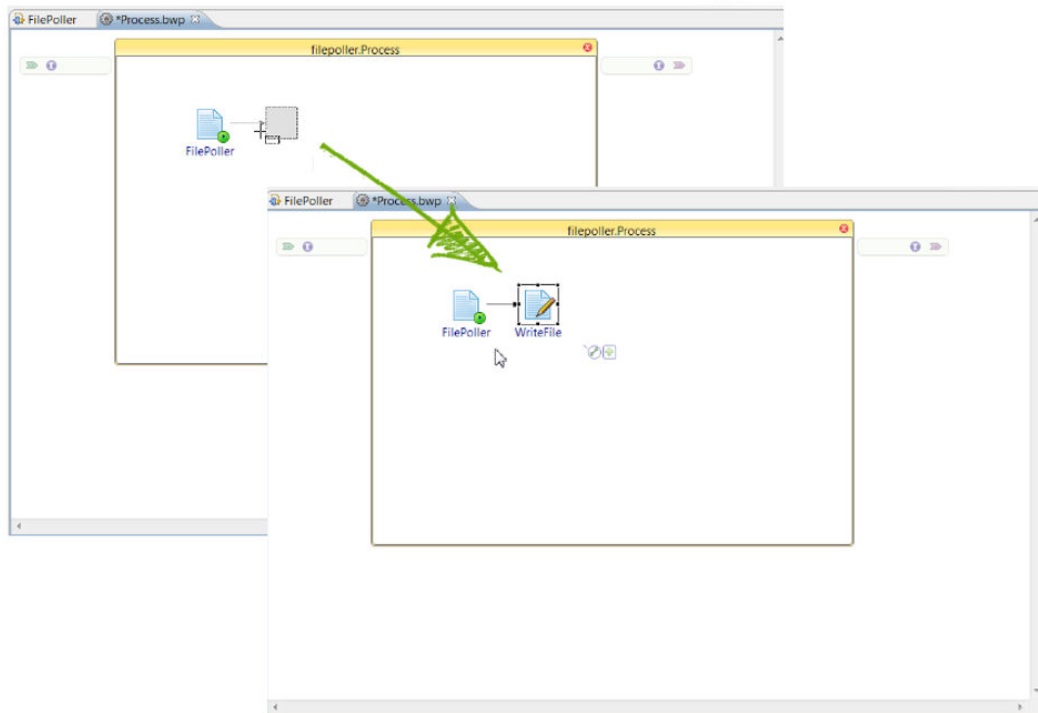
The **File Poller** and **Write File** activities in the **File** palette are used in this process.


### Prerequisites

A text file is required, for example, `c:\tmp\fileread.txt`. Type a few lines in the file and save it.

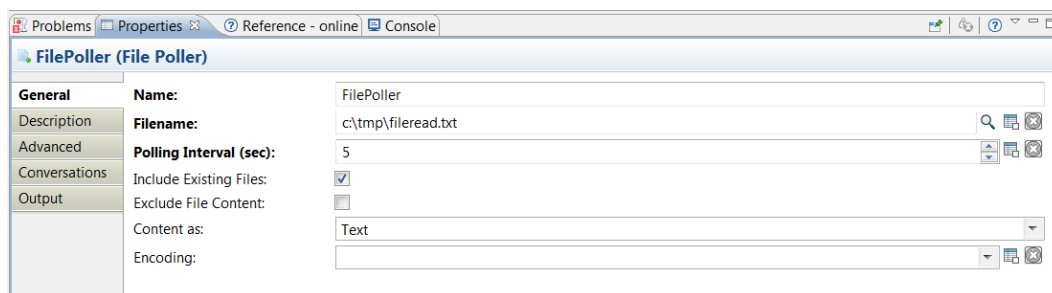
### Procedure

1. Select and drop a **File Poller** activity from the **File** palette to the **Process Editor** window.  
To add an activity to the **Process Editor**, click the activity and drop it on the **Process Editor**. Do not drag and drop the activity.
2. Select the **Write File** activity from the **File** palette. Click in the **Process Editor** next to the **File Poller** activity. You see a gray overlay indicating where you can place the activity, along with the transition arrow. When you drop the activity, the transition from the **File Poller** activity to the **Write File** activity is created.




To create a transition between two activities click the transition  icon and join both the activities in the Process Editor.

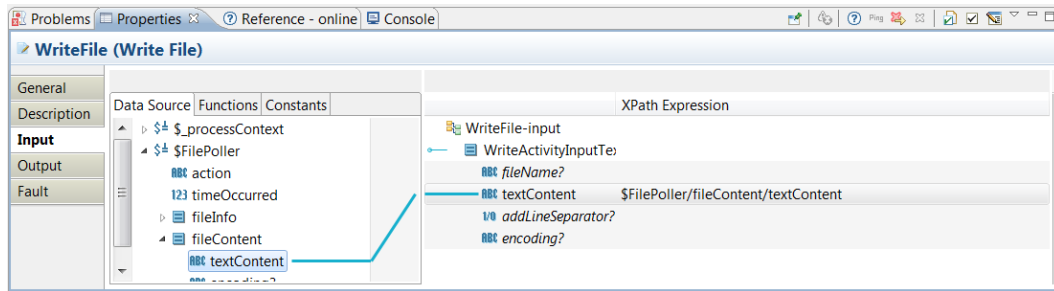
3. Select the **File Poller** activity. In the **Properties** tab, select the **General** tab. Point to the file you created as a prerequisite to this tutorial.
4. Select the **Include Existing Files** check box. When selected, the **File Poller** activity polls the existing file regardless of the changes made. The specified file is periodically polled at the specified interval even if the file has not changed. These changes may pertain to modifying, creating, or removing the file.



The Polling Interval (sec): 5 (default) indicates the Frequency with which the File Poller activity monitors this particular file. Any update to this file is transferred to the output .log file through the **Write File** activity.

5. Save your project. Click **File > Save** or the **Save** button on the tool bar.
6. Select the **Write File** activity and click the **General** tab.
7. Click the  button in the **Filename** field and specify the output location, such as, c:\tmp\FilePoller\output.log. Also, select the **Create Non-Existing Directories** check box.
8. Click the **Input** tab to specify input to the **Write File** activity. Drag the FilePoller\fileContent\textContent from the **Data Source** pane into the **textContent** field in the **XPath Expression** pane. This writes the content of the polled file to the output file.





9. Save your project.
10. Test this project in the debugger. For more information about the testing procedure, see [Testing the File Poller Application in Debugger](#).


## Result

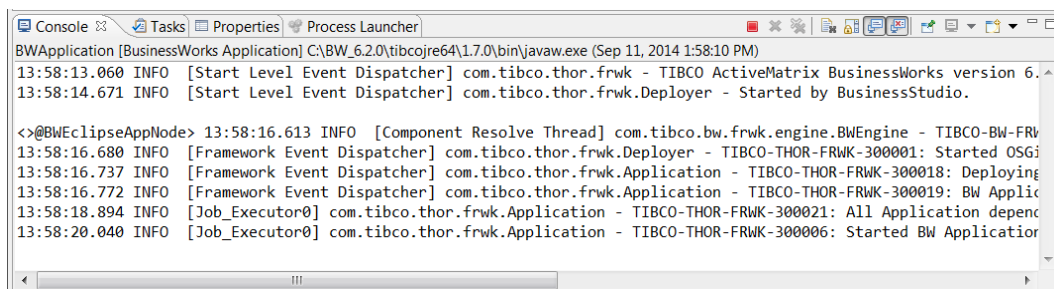
The **File Poller** activity polls the `fileread.txt` file located in `c:\tmp` file every 5 seconds and any changes made to the file content are written to the `output.log` file at `c:\tmp\FilePoller` by the **Write File** activity.

## Testing the FilePoller Application in the Debugger

The debugger provides a simple and fast way of debugging one or more TIBCO ActiveMatrix BusinessWorks™ applications in a local runtime environment. The applications must be in the workspace and selected before launching the debugger. After starting the debugging session, the debugger does not provide tooling support for deploying and debugging the application on the same runtime instance. The runtime starts when the debugger is started and stops when the debugger is stopped.

### Procedure

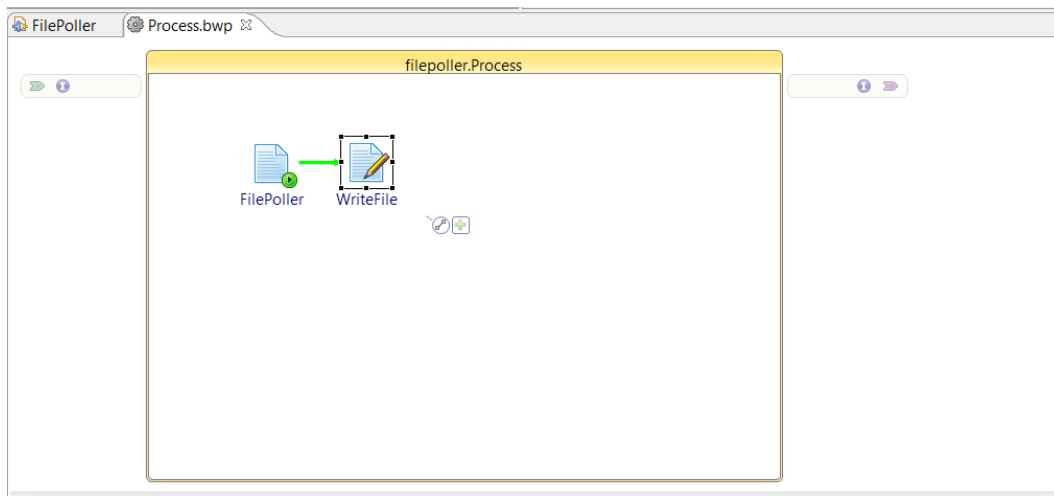
1. Right-click in the **Process Editor** and select the **Debug BusinessWorks Applications** option from the menu. You can also click  to start the debugger or choose **Run > Debug**. The following messages are displayed in the **Console view**.



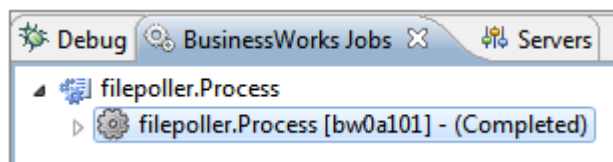
You are running the instance of the FilePoller application. When the debugger is launched, the perspective changes from **Modeling** to **Debug**.

2. After the application starts, the **File Poller** and **Write File** activities get triggered and the content of the `fileread.txt` is written to `c:\tmp\FilePoller\output.log`.
3. Modify the `c:\tmp\fileread.txt` file. For example, open the `c:\tmp\fileread.txt` file and write "Hello BusinessWorks!" and save the changes.  
The **File Poller** activity polls every 5 seconds and transitions the contents of the `fileread.txt` to the **Write File** activity.
4. Open the file `c:\tmp\FilePoller\output.log` and verify the updated information, for example, "Hello BusinessWorks!" printed in this file.  
While keeping the application running, explore adding and deleting the words in the `fileread.txt` file and notice the corresponding changes made to the `output.log` file.

- The path taken by the engine for executing the process is displayed. Transitions turn green to specify that a path was executed.



- Click the **Businessworks Jobs** view on the top left to see the jobs created for the process.



- Click a particular activity, then click the **Job Data** view on the top right to see the input and output data of the activity.

The screenshot shows the 'Job Data' view for the 'FilePoller' activity. The output data is displayed in a table format.

Name	Value
FilePoller	<ns0:EventSourceOutputTextClass xmlns:ns0="http://www.tib...>
action	create
timeOccurred	1410973937194
fileInfo	<fileInfo>\r\n <fullName>c:\tmp\filerread.txt-</fullName>...
fileContent	<fileContent>\r\n <textContent>this is a test&#13;\r\n&#13...

- To stop the current job, click the **Stop** button  on the **Console view** toolbar.

# REST Service Tutorial

---

The REST Bookstore sample lets you explore the REST tooling in TIBCO Business Studio™ for BusinessWorks™. You can import this sample into TIBCO Business Studio for BusinessWorks through **File Explorer** and examine the project.

The processes in the sample implement different aspects of a bookstore, such as adding a book, deleting a book, and retrieving a list of books or a single book by its ISBN.

For more information about the sample, see "Using REST to Manage Books for a Bookstore" in *TIBCO ActiveMatrix BusinessWorks™ Samples*.

This tutorial walks you through the steps to build an additional REST service for the sample and test it in the debugger. You can use the Swagger UI to invoke the operations for the REST resource.

## Prerequisites

Have the following installed or created on your computer:

- PostgreSQL
- Required database and tables
- Most recent version of Google Chrome web browser

For more information, see [Installing PostgreSQL](#).

## Installing PostgreSQL

This topic explains how to install the PostgreSQL database and create the database and tables required for the Bookstore tutorial.

### Procedure

1. Download and install PostgreSQL from <http://www.postgresql.org/download/>

Note the superuser password that you create as part of the installation process.




If installing on Windows, do not install or run as Administrator.

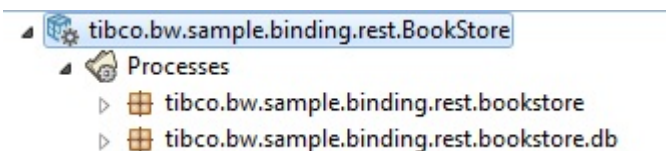
2. Open a terminal window and navigate to the root folder of the PostgreSQL installation. Open `pg-env.bat` and verify the path settings. Save the file if you make changes.
3. Start the server. Navigate to the `bin` folder of the install directory and type: `pg-ctrl start`  
Enter the password you created for the superuser.
4. Open another terminal window and navigate to the `BW_HOME\samples\binding\rest\BookStore\scripts` folder. Open `readme.txt`. On Unix systems, use the first command in the `readme` to start the script from the `psql` window. On Windows, copy the second command to start the script from the command line.
5. Navigate to the PostgreSQL `bin` folder and paste the command line into the terminal window. Modify the command as needed. For Windows, use forward slashes in the command.  
Run the command to create the database, the database tables, and to populate the database.
6. Open the PostgreSQL pgAdmin UI utility to see the database and tables.

## Creating a New Process

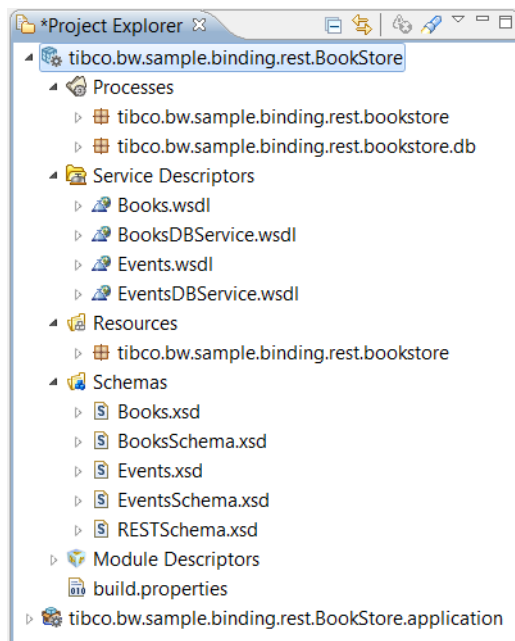
These steps show how to create a new process.

### Procedure

1. Open TIBCO Business Studio™ for BusinessWorks™.
2. Open the **Design** perspective by clicking  in the upper right.
3. Click the **File Explorer** tab. If the tab is not visible, click **Window > Show View > Other > FileSystem > File Explorer** and click **OK**.
4. Click **File > Switch Workspace** and select or open a clean new workspace.
5. In the samples directory, select **binding > rest > Bookstore** and double-click **tibco.bw.sample.binding.rest.BookStore.zip**. This opens the project in the **Project Explorer**.
6. In the **Project Explorer**, expand the **tibco.bw.sample.binding.rest.BookStore** project.
7. You can also import the sample using the **File > Import > General > Existing Studio Projects into Workspace > Select Archive File > Browse** option.
8. The project is displayed in the **Project Explorer** panel on the left.



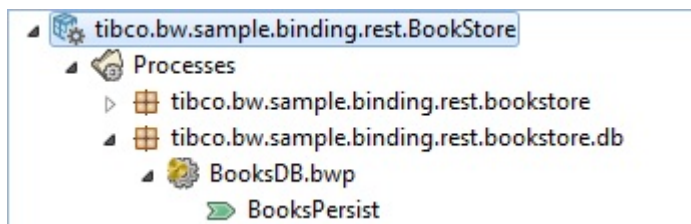
9. Expand the folders in the project to see all the project processes and resources. For more information on folder structure, see the *TIBCO ActiveMatrix BusinessWorks™ Application Development guide*.



10. Expand **Processes** and then expand **tibco.bw.sample.binding.rest.bookstore.db**. See **BooksDB.bwp**.



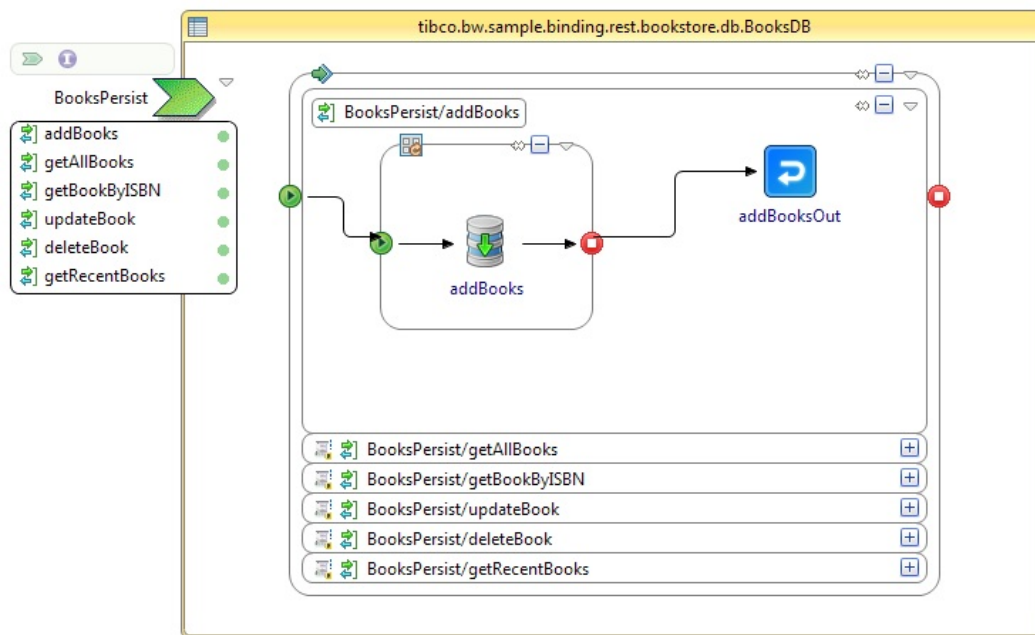
The .bwp extension indicates that it is an ActiveMatrix BusinessWorks™ process.



11. Double-click **BooksDB.bwp**.

The process comprises:

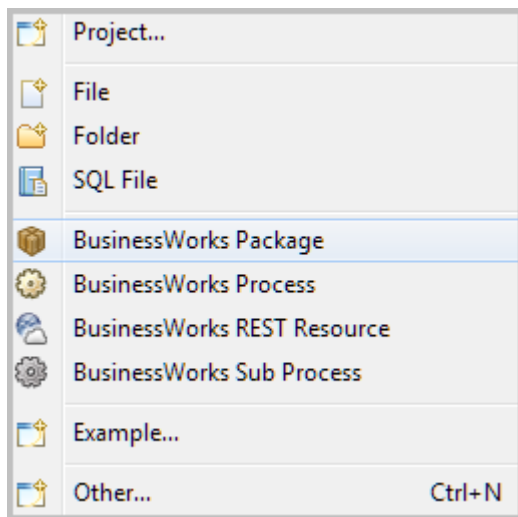
- Green chevron on the left indicates the service details.
- addBooks, getAllBooks, and so on indicate the operations implemented by this process.
- Each operation is implemented separately.



12. Double-click an operation to display the process for example, **BooksPersist > addBooks**.

- In the addBooks operation, you can see a JDBC activity.
- The activity is repeated using a ForEach group.
- addBooksOut represents the **Response** to the web service request.

13. To add a new process package named tibco.bw.sample.rest, right-click on **Processes** in the **Project Explorer** view, and select **New > BusinessWorks Package**.



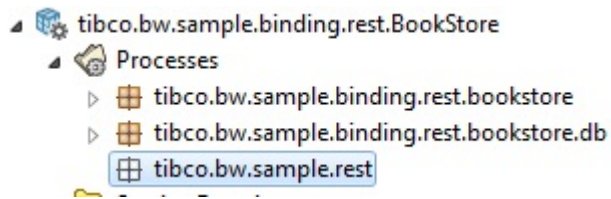
14. In the BusinessWorks Package screen, specify `tibco.bw.sample.rest` in the **Name** field.

Creates folders corresponding to packages.

BW Process Folder:

Name:

15. Click **Finish** and verify that the new package `tibco.bw.sample.rest` has been added in the **Project Explorer** view.



## Building a REST Service

This section details how to build a REST service.

### Prerequisites

The `tibco.bw.sample.binding.rest.BookStore` sample is loaded in the Project Explorer.

### Procedure

1. To define a REST Resource named `MyBooks`, select `tibco.bw.sample.binding.rest.BookStore` > **New** > **BusinessWorks REST Resource**.  
The REST Service Wizard window is displayed.

REST Service Wizard

## Create a new REST Resource

Configure the REST Service implementation.

Resource Name:

Summary:

Resource Service Path:

Resource Definition:

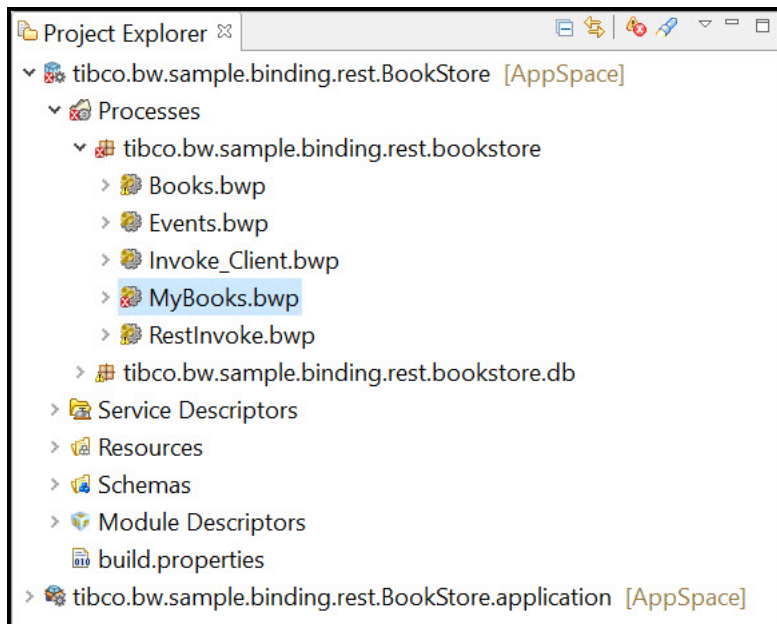
Operations:

POST  GET  PUT  DELETE  PATCH

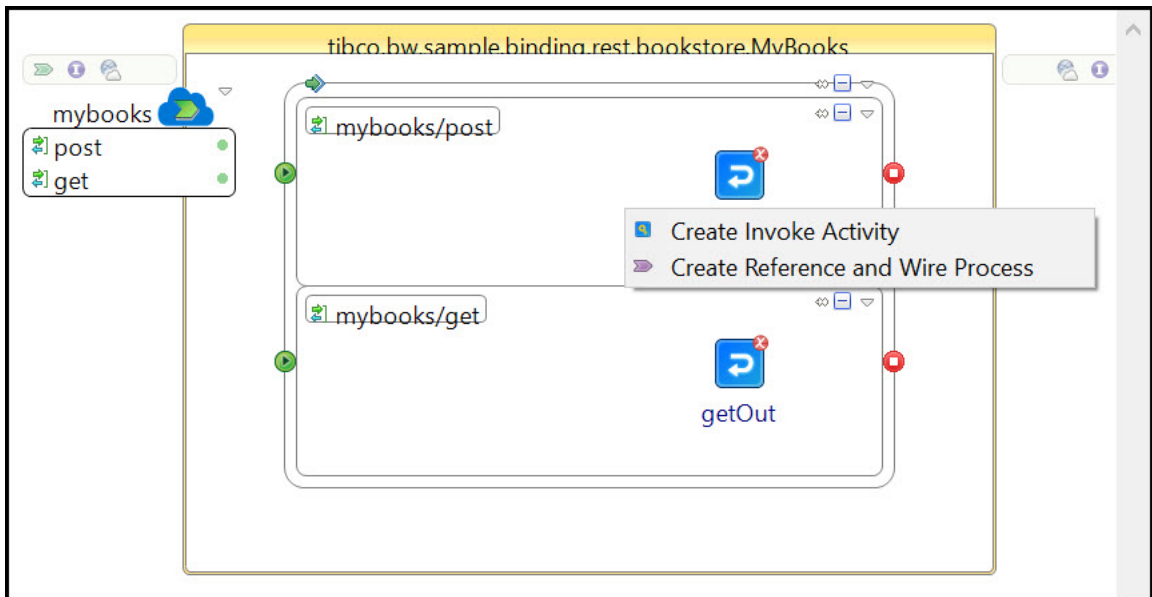
OPTIONS  HEAD

Implementation Data:  Structured  Opaque

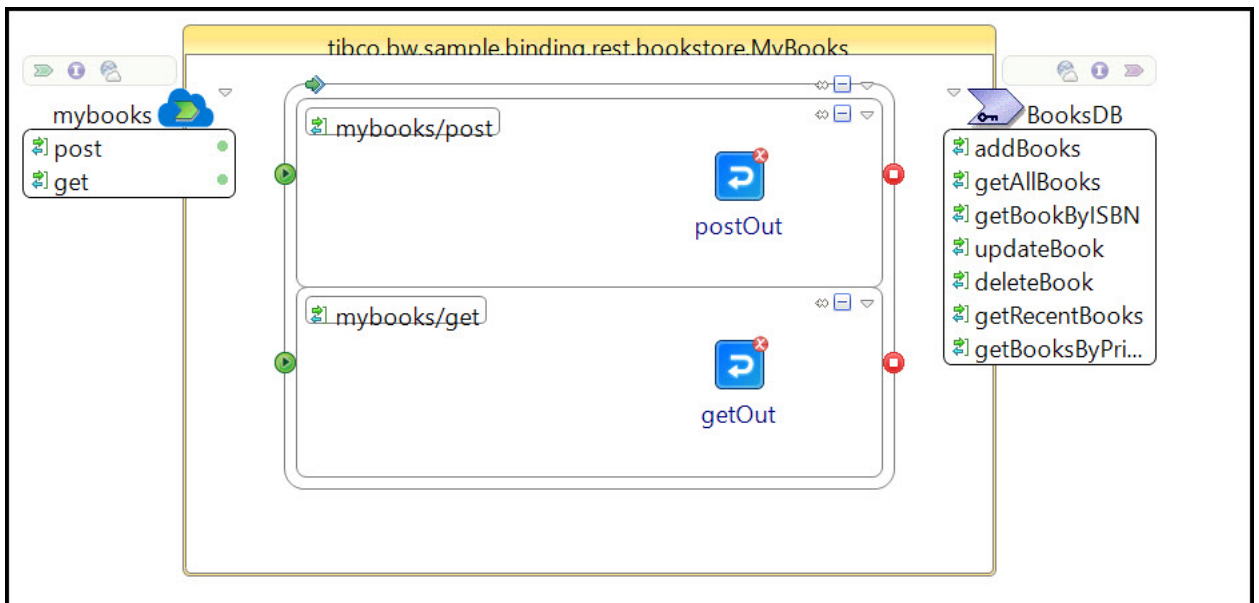
2. Specify the following values in the REST Service Wizard window.
  - a) **Resource Name:** MyBooks
  - b) **Summary:** Summary about the new REST service. (default)
  - c) **Resource Service Path:** Auto-filled
  - d) **Resource Definition:** Select **Browse > Schemas > Books.xsd > Books** in the Select Schema Element Declaration window.
  - e) **Operations:** Select POST and GET check boxes.
  - f) **Implementation Data:** Accept the default value of **Structured**.
3. Click **Finish**.  
This creates a new process **MyBooks.bwp** process is opened in the **Process Editor**.



4. Open the **tibco.bw.sample.binding.rest.bookstore.db** package in the **Project Explorer** and select the **BooksDB.bwp** process. Drag it to the **Process Editor** and drop it on the implemented POST operation. A menu is displayed with two options: **Create Invoke Activity** and **Create Reference and Wire Process**.

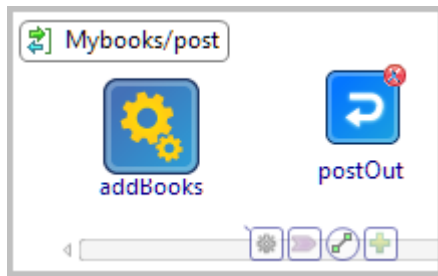


5. Select **Create References and Wire Process**. The references are added to the process. The purple chevron indicates the service and its operations that can be referenced by the process.



6. To update the POST process to invoke the appropriate external service operation:
  - a) Click the **addBooks** operation.
  - b) Select and drag the operation to the left of the **postOut** activity and drop it. An Invoke process activity is created.





- Click the newly added activity. Select the  icon and connect **addBooks** to **postOut**.



- Click the **getAllBooks** operation and select, drag, and drop the operation to the left of the **getOut** activity in the OUT process.
- Connect **getAllBooks** to **getOut**.

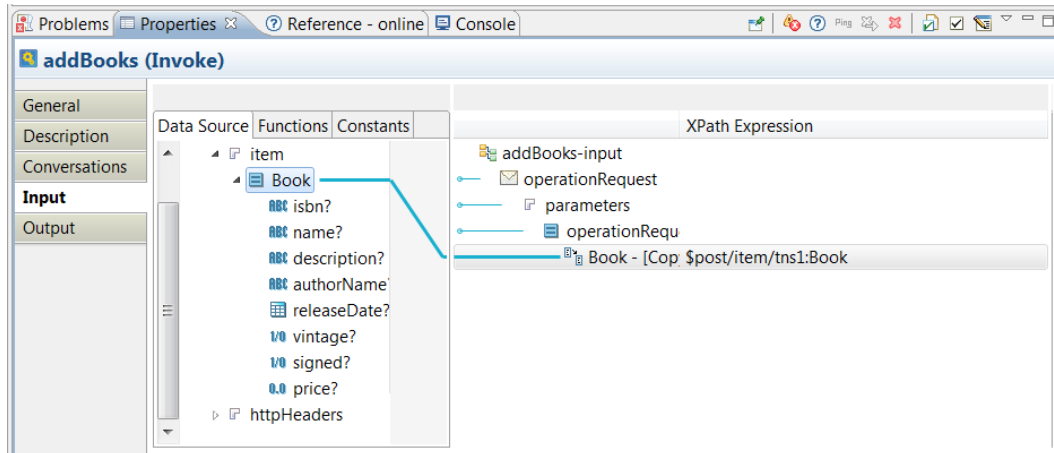


- Save your changes.
- Click the **addBooks** activity and select **Properties > Input**.
- Expand the data tree in the **Data Source** pane to locate the Book element.

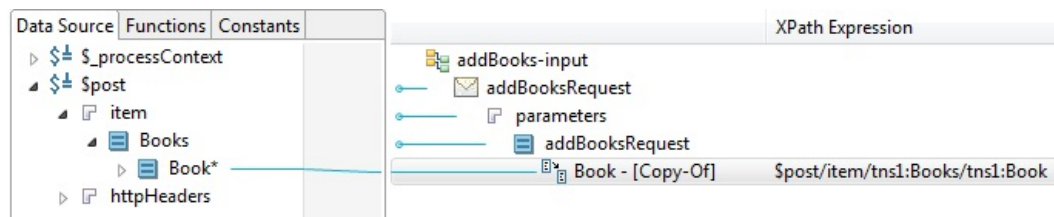
A screenshot of the "addBooks (Invoke)" properties dialog in the Input tab. The "Data Source" pane on the left shows a tree structure: "item" (expanded) contains "Book" (expanded), which lists attributes: "isbn?", "name?", "description?", "authorName", "releaseDate?", "vintage?", "signed?", and "price?". Below "Book" is "httpHeaders". The "XPath Expression" pane on the right shows a tree structure: "addBooks-input" contains "operationRequest" (expanded), which contains "parameters" (expanded), which contains "operationReque" (expanded), which contains "Book\*" (expanded), which lists attributes: "isbn?", "name?", "descriptio", "authorNa", "releaseDa", "vintage?", and "signed?".

- Drag the Book element from the left to the Book\* element on the right.
- In the pop-up window, select **Make a Copy of each "Book"** and click **Finish**.

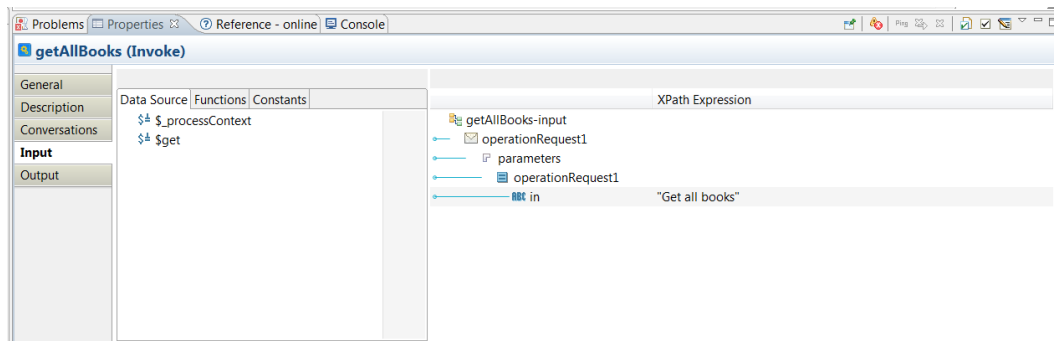
The **Input** tab looks like this:



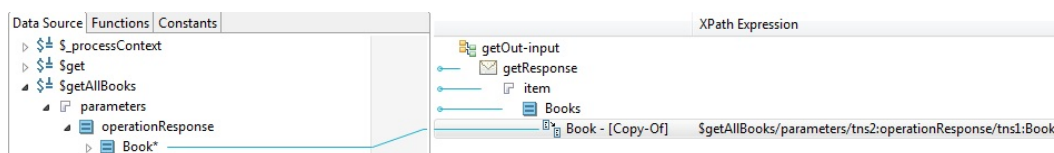
15. Save your changes.
16. Click the **postOut** activity and open the **Properties > Input** tab. Expand the **post** activity and drag the **Book\*** element from left to right.
17. In the pop-up window, select the **For each** option and click **Next**. Click **Finish** on the **Auto-Map** window. The **Properties > Input** tab looks similar to this:



18. Click **getAllBooks** and select **Properties > Input**.
19. In the **XPath Expression** pane, add a dummy value to the input element, such as, "Get All Books". The input must be in quotes.



20. Click the **getOut** activity in the **Process Editor**, and select the **Properties > Input** tab. Expand the **getAllBooks** activity and choose **Book\*** to map the **Book\*** element from left to right. In the pop-up window, choose **Make a Copy of each "Book"** and click **Finish**. The tab looks similar to this:



## Result

Your project is complete without any errors.

## Testing the REST Service

You can now test the REST service using the built-in tester and the Swagger UI.

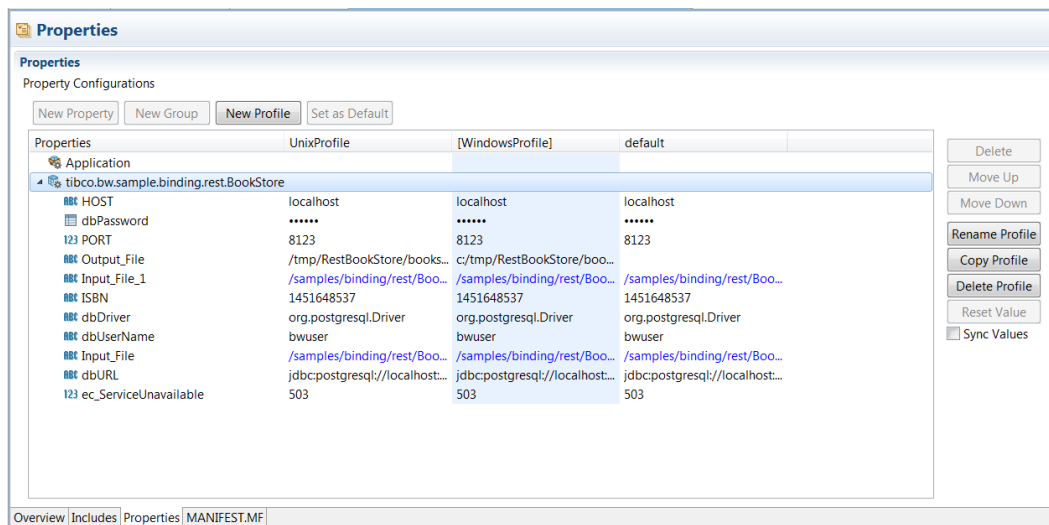
### Procedure



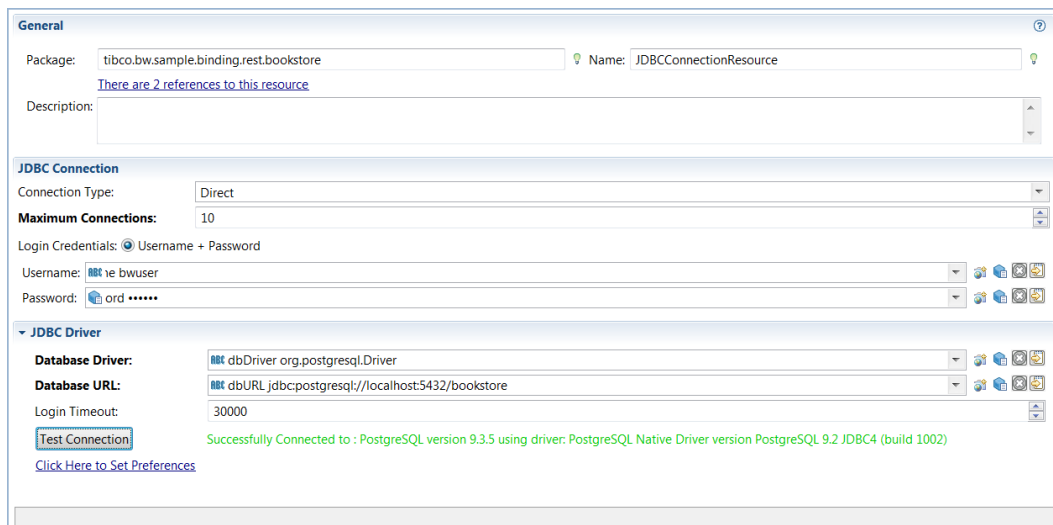
The Swagger framework used to test the REST service is accessed using the port specified by the `bw.rest.docApi.port` property in the `<TIBCO_HOME>\bw\<version>\domains\<domain name>\appnodes\<appspace name>\<appnode name>\config.ini` file. By default, this property is set to port 7777. If you have multiple REST applications running on separate AppNodes on a single machine or server, you must configure these port numbers to be unique for each AppNode. To do so, edit the `bw.rest.docApi.port` property in the `config.ini` file for each AppNode.

If there are multiple REST applications running within a single AppNode, there is no need to configure this port.

1. In the **Project Explorer**, expand the **tibco.bw.sample.binding.rest.BookStore.application** process and expand the **Package Unit > Properties** folder.
2. In the **Properties** window, open the **tibco.bw.sample.binding.rest.BookStore** process and set the default **ApplicationProfile** to match the operating system you are running on. The bracketed profile in the column head is the one that is selected:



3. Provide valid values for the application properties including a valid user name, password, and database URL to connect to your PostgreSQL database if different from the default setting.
4. Verify your JDBC connection.
  - a) Expand the **Resources** folder in the Project Explorer for the **tibco.bw.sample.binding.rest.BookStore** process.
  - b) Double-click **JDBCConnectionResource.jdbcsResource**.
  - c) In the **JDBC Driver** section of the window, click **Test Connection** to verify the connection. If you change the JDBC driver folder from the default, click **Click Here to Set Preferences** and set the JDBC driver folder to the folder where you downloaded PostgreSQL JDBC Driver.



5. Click **File > Save**.
6. In the **Project Explorer**, expand the **Processes** directory if it is not expanded and double-click **MyBooks.bwp**.
7. Click **Run > Debug Configurations**.
8. In the left-hand tree of the **Debug Configuration** wizard, expand **BusinessWorks Application** and select **BWApplication**.
9. Click the **Applications** tab and then click the **Deselect All** button if you have multiple applications. Select the check box next to **tibco.bw.sample.binding.rest.BookStore.application**.
10. Click **Debug**. This runs the sample in **Debug** mode.  
The **Console view** is opened and shows engine messages similar to: Started BW Application [tibco.bw.sample.binding.rest.BookStore.application:1.0].
11. In the **Console view**, press Enter to display the prompt: <>@BWEclipseAppNode>  
Enter the OSGi command **lrestdoc**. This lists the Swagger UI URL as the discovery URL:  
[Application Name]: tibco.bw.sample.binding.rest.BookStore.application [Discovery Url]: http://localhost:7777/tibco.bw.sample.binding.rest.BookStore.application
12. Launch the Google Chrome browser.
13. Open <http://localhost:7777/tibco.bw.sample.binding.rest.BookStore.application>
14. Click **Books** or **Events** to see the operations. Click **MyBooks** to see the REST service operations you just added. For more information, see the section [Testing the POST and GET Operations](#).

15. Expand the Books and Events headers, and test out the operations as listed below.

### Result

Click **Books** or **Events** in the Swagger UI to view the following operations for Books and Events:

#### Books

- Post books
- GET books
- GET book by ISBN
- PUT book by ISBN
- DELETE book by ISBN

#### Events

- POST Events
- GET Events
- GET Event by EventID
- PUT Event by EventID
- DELETE Event by EventID

**GET books** returns an output similar to the following:

```
{
  "Book": [
    {
      "isbn": "0061122416",
      "name": "The Alchemist",
      "description": "Every few decades a book is published that changes the lives of
its readers forever. The Alchemist is such a book",
      "authorName": "Paul Coelho",
      "releaseDate": "2006-04-25",
      "vintage": true,
      "signed": true,
      "price": 11.9
    },
    {
      "isbn": "0071450149",
      "name": "The Power to Predict",
      "description": "How Real Time Businesses Anticipate Customer Needs, Create
Opportunities, and Beat the Competition",
      "authorName": "Vivek Ranadive",
      "releaseDate": "2006-01-26",
      "vintage": false,
      "signed": true,
      "price": 15.999
    }
  ]
}
```

**GET books** by ISBN returns an output similar to the following for ISBN 0061122416:

```
{
  "isbn": "0061122416",
  "name": "The Alchemist",
  "description": "Every few decades a book is published that changes the lives of
its readers forever. The Alchemist is such a book",
  "authorName": "Paul Coelho",
  "releaseDate": "2006-04-25",
  "vintage": true,
  "signed": true,
  "price": 11.9
}
```

The `books.log` file is generated with the following information:

```
POST Books----->{"Book":[{"isbn":"1451648537","name":"Steve
Jobs","description":"Biography of Apple Co-Founder Steve Jobs","authorName":"Walter
Isaacson","releaseDate":"2012-10-24","vintage":false,"signed":false,"price":21},
{"isbn":"0385537859","name":"Inferno","description":"Robert Langdon returns in Dan
Brown's latest fast paced action thriller","authorName":"Dan
Brown","releaseDate":"2013-05-14","vintage":false,"signed":true,"price":14.09},
{"isbn":"0399103421","name":"The Godfather","description":"The Godfather is an epic
story of a New York's top mafia family, loyalty, and how men of honor live in their
own world, and die by their own laws.","authorName":"Mario
Puzo","releaseDate":"1969-03-10","vintage":true,"signed":true,"price":50}]}*****
*****

GET Books----->{"Book":[{"isbn":"1451648537","name":"Steve
Jobs","description":"Biography of Apple Co-Founder Steve Jobs","authorName":"Walter
Isaacson","releaseDate":"2012-10-24+05:30","vintage":false,"signed":false,"price":21},
{"isbn":"0385537859","name":"Inferno","description":"Robert Langdon returns in Dan
Brown's latest fast paced action thriller","authorName":"Dan
Brown","releaseDate":"2013-05-14+05:30","vintage":false,"signed":true,"price":14.09},
{"isbn":"0399103421","name":"The Godfather","description":"The Godfather is an epic
story of a New York's top mafia family, loyalty, and how men of honor live in their
own world, and die by their own laws.","authorName":"Mario
Puzo","releaseDate":"1969-03-10+05:30","vintage":true,"signed":true,"price":50}]}*****
*****

GET Book By ISBN----->{"isbn":"1451648537","name":"Steve
Jobs","description":"Biography of Apple Co-Founder Steve Jobs","authorName":"Walter
Isaacson","releaseDate":"2012-10-24+05:30","vintage":false,"signed":false,"price":21}**
*****
```

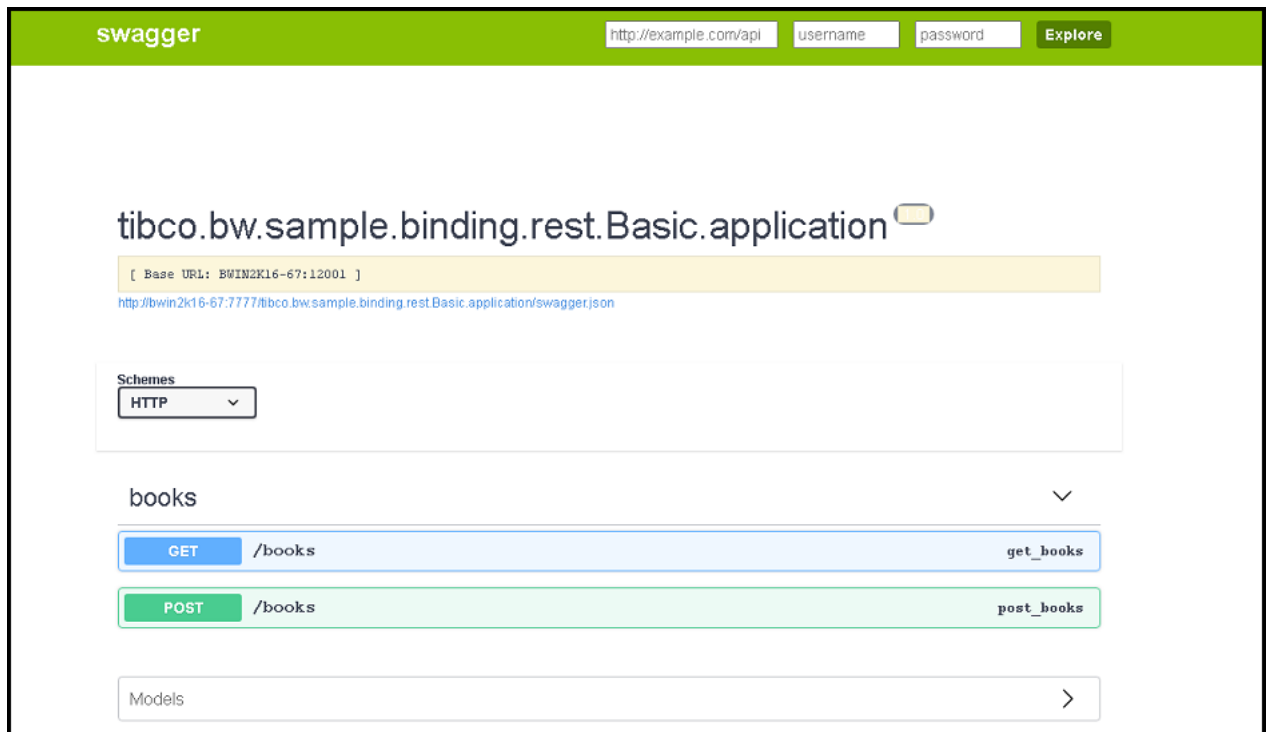
```
DELETE Book By ISBN----->"Deleted book with ISBN -
1451648537"*****
GET Events By ISBN---->{}*****
```

## Testing the POST and GET Operations

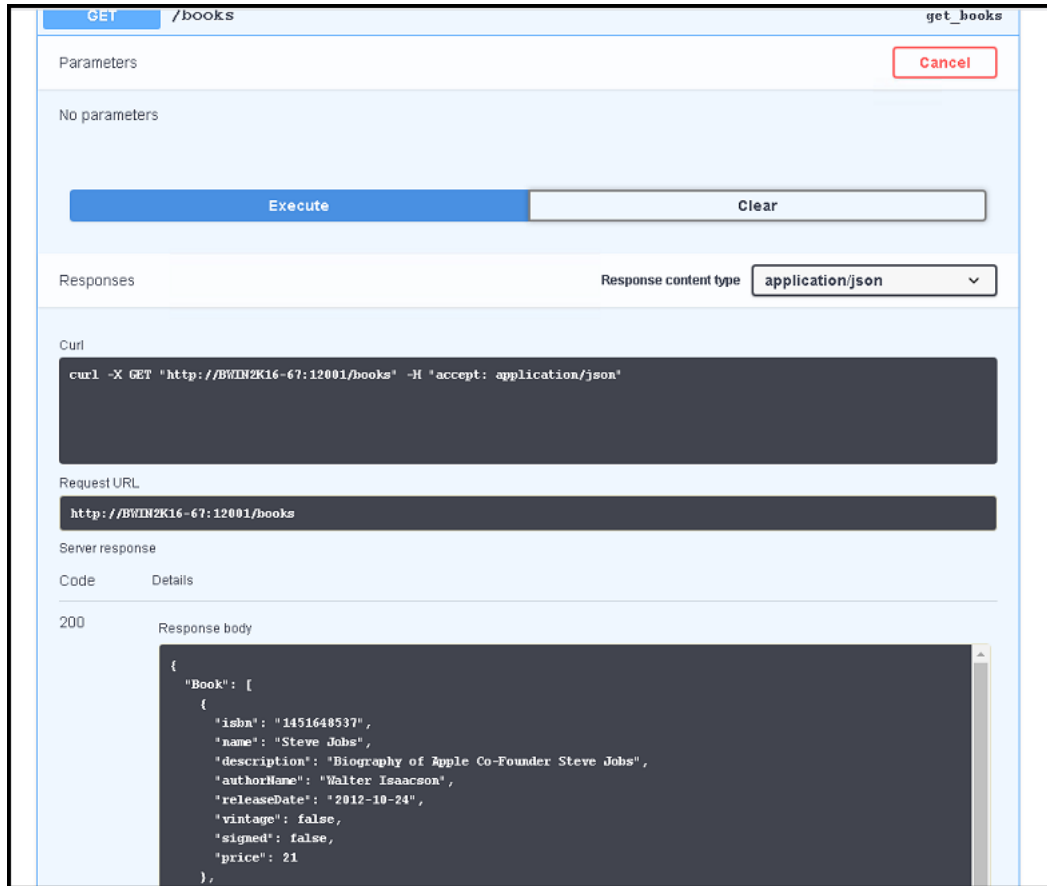
An available RESTful service displays the GET operation in the Swagger UI. The POST operation is tested using the JSON service. It is important to test these operations by doing some simple tasks. This section explains how to test the POST and GET operations you just added.


### Procedure

1. Click **books**.  
It expands and displays the POST and GET operations.



2. Click the **POST** icon to display its details.
3. Click the **Try it out!** button.
4. Provide values to the Books parameter and then click **Execute** button. You can use the JSON payload in the `<BW_HOME>\samples\binding\rest\BookStore\samplejson` folder.
5. Now click the **GET** icon to display its details.
6. Click the **Try it out!** button.
7. Click the **Execute** button.  
The response displays a list of books returned by the REST service from the database.



8. After you have finished, go back to TIBCO Business Studio™ for BusinessWorks™ and click  in the **Console view** to stop the process.

## Troubleshooting

You may encounter some errors while executing or running the process. The following are some of the possible errors you may encounter and their resolutions.

Error Encountered	Resolution
Unable to insert rows into the database using the <code>dbsetup.sql</code> script in the scripts folder.	Use the sample JSON payload from the sample json folder to post the data.
The REST Swagger UI page is not visible.	Verify that the application has started and that you are accessing the correct URL. Use the <code>1restdoc</code> command in the <b>Console view</b> to get the Swagger UI URL.
Problem markers are visible in the project.	Clean the project by invoking <b>Project &gt; Clean</b> or by switching to a clean new workspace.
Getting the File was not found exception.	Ensure that the <code>books.json</code> and <code>book_put.json</code> files are present at the location described in the <code>Input_File</code> and <code>Input_File_1</code> module properties.
The PostgreSQL server does not start.	Make sure you are not running as Administrator.



## Error Encountered

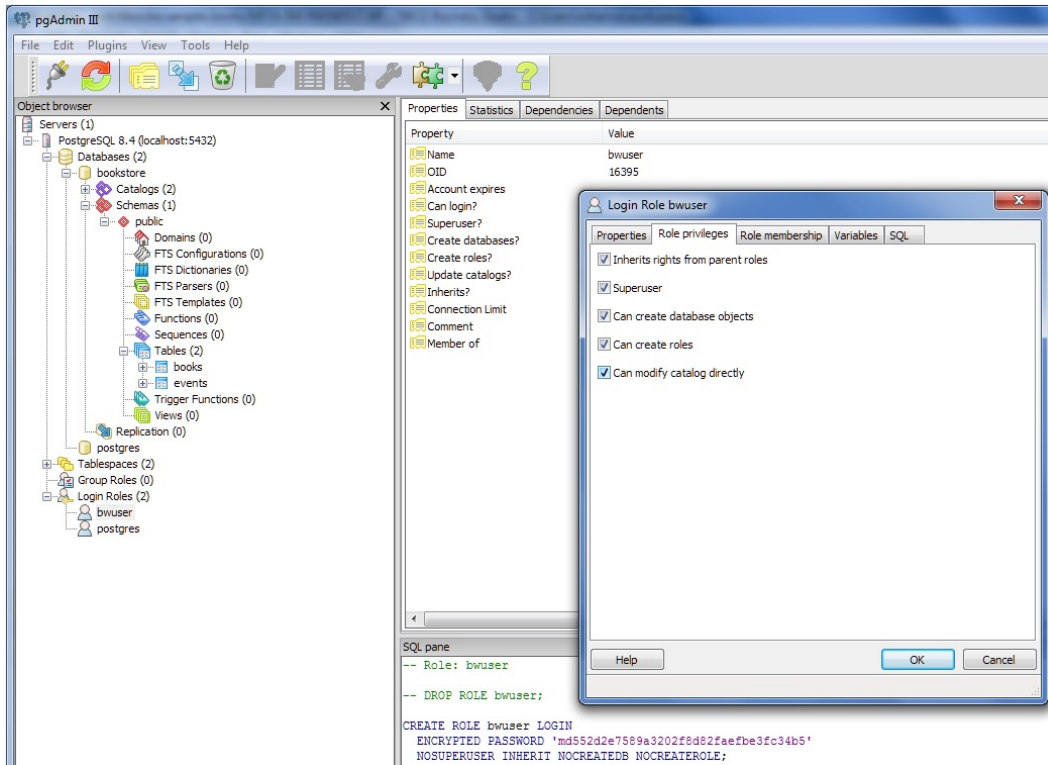
## Resolution

The database and database tables are not created.

Open the `readme.txt` file for the sample, located in the `BW_HOME\samples\binding\rest\BookStore\scripts` folder. Run the `dbsetup.sql` script from a command line, not the `psql` window.

Getting an unregistered user error message while running the process.

Select all the check boxes in the **Role Privileges** tab in the pgAdmin UI and run the process again. See the image below.



# REST Reference Tutorial

---

The REST reference tutorial shows you how to create a simple REST Invoke to an existing REST Service defined by a Swagger specification.

You cannot convert REST reference to SOAP or vice versa.

## Prerequisites

The REST service which you want to invoke must be accessible from the reference process at the time of its invocation.

## Creating a New Application

1. Open TIBCO Business Studio for BusinessWorks.
2. Open the **Design** perspective by clicking the **Design** icon in the upper right corner.
3. Click **File > New > Other > BusinessWorks > BusinessWorks Application Module** and click **Next**.
4. Enter `tibco.bw.sample.binding.rest` in the **Project Name** text box. Do not change the remaining default settings.
5. Click **Finish**. This step creates a new application module with an empty process.
6. Obtain the Swagger file from the Swagger UI of the running service.
7. Copy and paste the content into a new file, and call it `Books.json`.

## Importing the JSON File into your Project

1. In the **Project Explorer**, expand `tibco.bw.sample.binding.rest` application module.
2. Right-click **Service Descriptors** and select **Import > Import... > General > File System** and click **Next**.
3. In the File system dialog box, click the **Browse** button and browse to the location of the `Books.json` file.
4. Select the check box next to **Books.json** in the left pane and click **Finish**.

## Creating the REST Reference

1. In the **Project Explorer**, completely expand the `tibco.bw.sample.binding.rest` folder under **Service Descriptors**.
2. Select the `/books` under **Books.json** and drag and drop it to the right side of the process in the Process Editor. The references are added to the process. The purple chevron indicates the service and its operations.
3. In the **Process Editor**, right-click **Add Activity > General Activities > Timer**. Optionally, you can configure the **Sleep** activity with **IntervalInMillisec** as 3000 in a similar manner and connect the **Timer** with **Sleep**.
4. Drag the **get** operation under the purple chevron and drop it on the right of **Timer** activity (or **Sleep** if configured) and connect the **Timer** activity with the **get** activity.
5. Drag the **post** operation under the purple chevron and drop it on the right of the **get** activity, connect the **get** activity with the **post** activity.
6. Right-click the **get** activity select **Show Properties View**.
7. In the **Properties** view, select the **Input** tab and click **Show Check and Repair** icon in the icon bar on the upper right corner of the **Properties** view.
8. Select the check box under **Fix** and click **OK**.

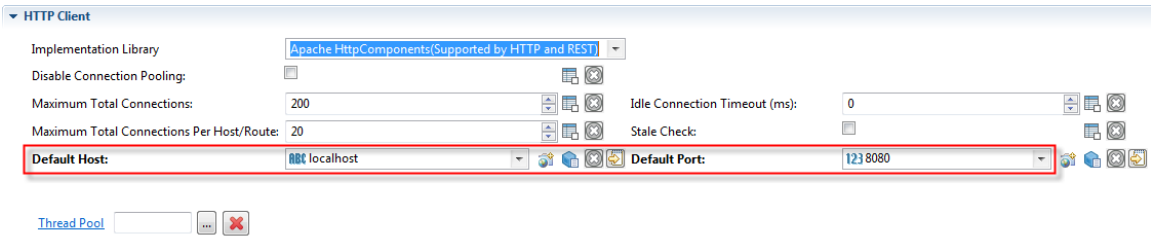
9. Click **Show Check and Repair** icon again.
10. Select the check box under **Fix** and click **OK**.
11. Select the **post** activity and right-click and select **Show Properties View**. In the **Properties View**, select the **Input** tab and select **Data Source** tab.
12. Expand **\$get** in the **Data Source** tab completely.
13. In the XPath Expression pane, expand the **post-input** completely.
14. Drag and drop **Book\*** from the **Data Source** tab to the **Book\*** under **post-input** in the **XPath Expression** pane.
15. In the Drop dialog, select **Make a copy of each "book"** radio button and click **Finish**.
16. Click **Show Check and Repair** icon in the icon bar on the upper right corner of the Properties view.
17. Select the check box under **Fix** and click **OK**.
18. Click **Show Check and Repair** icon again. Select the check box under **Fix** and click **OK**.
19. In the **Project Explorer**, select **Books.json** under **Service Descriptors** of **tibco.bw.sample.binding.rest.basic** application module, and right-click **Open With > Text Editor** and locate the "host" attribute. Make a note of the host name and port number.

```

{
  "swagger" : "2.0",
  "info" : {
    "version" : "1.0",
    "title" : "Summary about the new REST service.",
    "description" : "Summary about the new REST service."
  },
  "host" : "localhost:8080",
  "basePath" : "/",
  "schemes" : [ "http" ],
  "consumes" : [ "application/json" ],
  "produces" : [ "application/json" ],
  "paths" : {
    "/books" : {
      "post" : {
        "description" : "",
        "operationId" : "postbooks",
        "consumes" : [ "application/json" ],
        "produces" : [ "application/json" ],
        "parameters" : [ {
          "name" : "body",
          "in" : "body",
          "description" : "",
          "schema" : {
            "$ref" : "#/definitions/Books"
          }
        } ],
        "required" : true,
        "allowMultiple" : false
      },
      "responses" : {
        "200" : {
          "description" : "a Books to be returned",
          "schema" : {
            "$ref" : "#/definitions/Books"
          }
        }
      }
    }
  }
},

```

20. Expand the Resources folder under the **tibco.bw.sample.binding.rest.basic** application module.
21. Double-click **HttpClientResource.httpClientResource**.
22. In the **HTTP Client** section, change the Default Host and Default Port to the values in the **Books.json** file and select the **Default Confidentiality** check box.

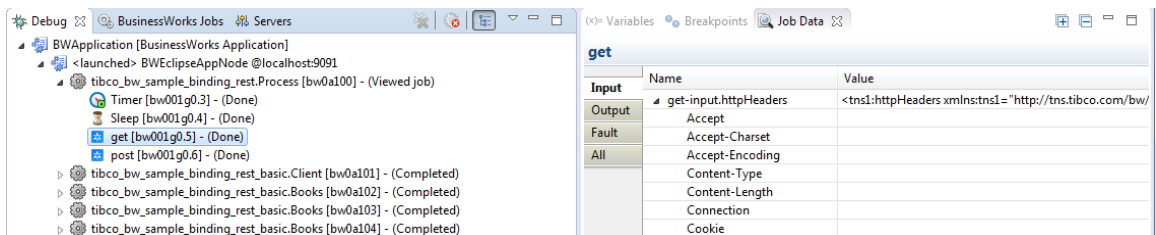


23. Click **File > Save All**.

## Testing the REST Reference

You can now test the REST service using the built-in tester and the Swagger UI. To do so follow these steps:

1. Click **Run > Debug Configuration**.
2. In the left pane of the **Debug Configuration** wizard, expand **BusinessWorks Application** and select **BWApplication**.
3. Click the **Applications** tab, then click **Deselect All** if you have multiple applications.
4. Select the check boxes next to **tibco.bw.sample.binding.rest.basic.application** and **tibco.bw.sample.binding.restapp**.
5. Click **Debug**. This runs the sample in debug mode. The Console view is opened and shows engine messages similar to: `Started BW Application [ tibco.bw.sample.binding.rest.Basic.application:1.0 ]`
6. In the **Debug** view, expand **BWApplication [BusinessWorks Application] > <launched> BWEclipseAppNode > tibco.bw.sample.binding.rest.Process** and select **get**.
7. In the **JobData** view, you can see the job data of the **get** activity.



# Administration Tutorial

---

The administration framework supports application deployment either through the `bwadmin` command line utility or the Admin UI. The scripts provided can be used to set up runtime entities that are useful for testing purposes. This tutorial walks you through running scripts and navigating runtime entities in the Admin UI.

The administration framework contains:

- The Admin UI hosted on TIBCO® Enterprise Administrator
- A powerful back-end `bwagent` designed to scale across large numbers of actual or virtual machines to manage large scale deployment.
- A simple, flexible, and easy-to-use `bwadmin` command line utility.

This section shows how to:

- Create runtime entities (Domains, AppSpaces, and AppNodes) and upload and deploy archive files using scripts.
- Start and stop applications using the Admin UI.
- Navigate runtime entities using the Admin UI.

For more information on runtime entities, see Administration Concepts in the *TIBCO ActiveMatrix BusinessWorks™ Concepts* guide.

For more information on the commands used in this sample, see Getting Started in the *TIBCO ActiveMatrix BusinessWorks™ Administration* guide.

## Running Admin Sample Scripts

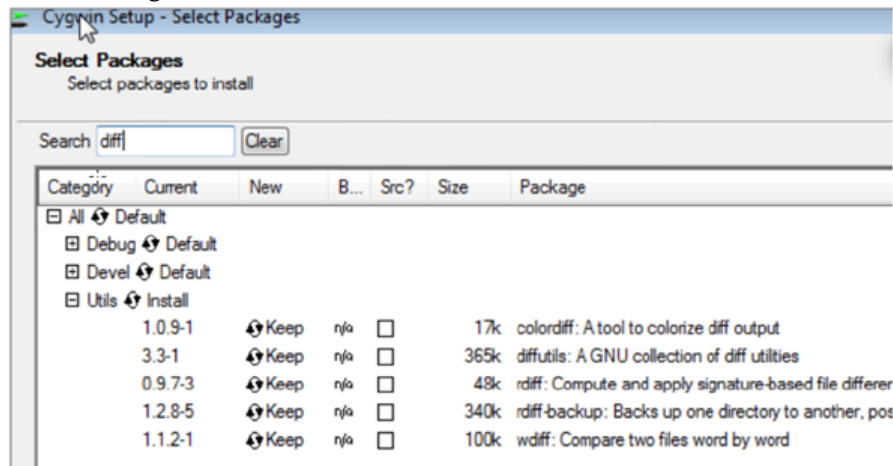
This tutorial walks you through running the Bookstore and Acme administration scripts.

### Prerequisites

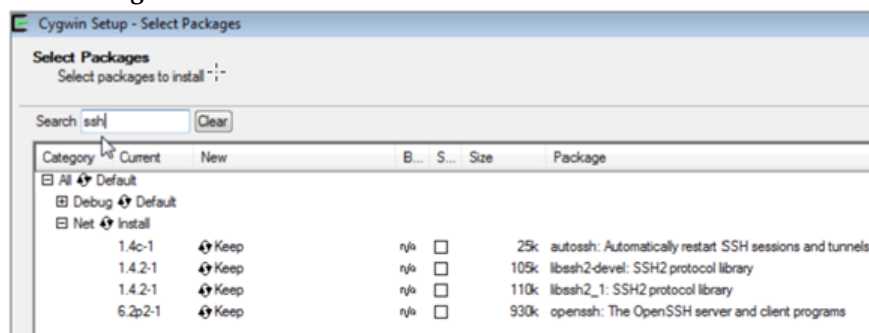
If you want to run scripts, the installation directory cannot contain one or more spaces. For example, on Windows, TIBCO ActiveMatrix BusinessWorks™ should not be installed in the `Program Files` folder.

1. Optional. Install Enterprise Message Service™ 8.x or higher if you want to run the `Acme.com` applications.
2. Optional. Install PostgreSQL 9.3.x if you want to run the Bookstore sample.
  - For more information, see [Installing PostgreSQL](#).
3. On Windows, you must install the latest Cygwin 64-bit version from <http://www.cygwin.com/>. Scripts must be executed with Cygwin. Install the **Utils** and **Net** package which contains the `diff.exe` and `ssh.exe` executables.
  - Select the **Utils** and **Net** packages and change the option from `Default` to `Install`. Refer to the following images that show the **Utils** and **Net** packages.

### Utils Package



### Net Package



### Procedure

1. Install ActiveMatrix BusinessWorks™. For example,
  - a) On Unix, install ActiveMatrix BusinessWorks into `/opt/tibco/bw-6.x` as `TIBCO_HOME`.
  - b) On Windows, install ActiveMatrix BusinessWorks into `/opt/tibco/bw-6.x` as `TIBCO_HOME`.
 From here onwards, the following instructions use the directory path mentioned above to refer to certain installed files. Replace this path with the install directory of your choice.



On Windows OS, avoid installing ActiveMatrix BusinessWorks in a directory with space. The product binary supports it, however, the scripts in `${BW_HOME}/scripts/admin` and `${BW_HOME}/samples/core/admin` are bash scripts and work best with directory paths without space.

2. Install TIBCO Enterprise Administrator 2.0 or compatible version, in the same `TIBCO_HOME`. When TIBCO Enterprise Administrator installer prompts for `JAVA_HOME` path, point to `/opt/tibco/bw-6.x/tibcojre64/1.7.0` on UNIX. On Windows, point to `c:/tibco/bw-6.x/tibcojre64/1.7.0`. You can install TIBCO Enterprise Administrator in a separate `TIBCO_HOME`. If you choose to do that, refer to [Step 4](#).
3. Install TIBCO Enterprise Message Service 8.1, or compatible version, in the same `TIBCO_HOME`. You can install TIBCO Enterprise Message Service in a separate `TIBCO_HOME`. If you have installed TIBCO Enterprise Message Service in a separate `TIBCO_HOME`, refer to [Step 4](#).
4. On UNIX: If you have installed TIBCO Enterprise Administrator and/or TIBCO Enterprise Message Service in a separate `<TIBCO_HOME>` than ActiveMatrix BusinessWorks, open `/opt/tibco/bw-6.x/bw/6.x/scripts/bashrc.sh` in a text editor and adjust `TEA_HOME` and `EMS_HOME` to point to where you have installed them.
5. This step is applicable only to Windows OS. From a proper text editor (Do not use Notepad.exe), open and edit `c:/tibco/bw-6.x/bw/6.x/scripts/bashrc.sh`. Search and replace all occurrences of `C:/` (or

whatever letter drive you have installed ActiveMatrix BusinessWorks) with `/cygdrive/c/` (or `/cygdrive/<drive-letter-where-you-installed-bw6>`).

Adjust `TEA_HOME` and `EMS_HOME` to point to the location, where you have installed them.

6. Source the `bashrc.sh` script from your `~/.bashrc` or `~/.profile` files.
  - **On UNIX:** Locate your `~/.bashrc` or `~/.profile` file and add the following line at the end:
 

```
source /opt/tibco/bw-6.x/bw/6.x/scripts/bashrc.sh
```
  - **On Windows:** Edit the `.bashrc` file located at `CYGWIN_HOME/home/<UserName>` to add the following line at the end:
 

```
source c:/tibco/bw-6.x/bw/6.x/scripts/bashrc.sh
```
7. To verify that the environment variables are configured correctly, open a new command line window, navigate to `<TIBCO_HOME>/bw/6.x/scripts/admin/`, and run `./bw6env.sh`. This script prints the required and optional environment variable configurations on the screen.



The script returns the environment variable configurations only if the environment variables are set correctly.

The following is a sample output on UNIX:

```
Admin@WINAA-2:admin admin$ ./bw6env.sh
BW 6 Environment Configurations

Required Environment Variables:
  TIBCO_HOME = /opt/tibco/bw-6.x
  BW_HOME    = /opt/tibco/bw-6.x/bw/6.x
  JAVA_HOME  = /opt/tibco/bw-6.x/tibcojre64/1.7.0

Optional Environment Variables:
  TEA_HOME   = /opt/tibco/bw-6.x/tea/2.1
  EMS_HOME   = /opt/tibco/bw-6.x/ems/8.1

Required Binaries on $PATH:

Optional Binaries on $PATH:
```

8. Change to the `admin` folder by typing: `admin`
9. Issue the following command to create the BookStore sample, create the domains for the Acme sample, create the Samples-Domain, and deploy all sample archive files:
 

```
runAll.sh
```



- Running the `runAll.sh` script automatically updates the Admin "local" mode to "enterprise" mode.
- You can use `-clean` option which cleans TIBCO Enterprise Administrator Server Data Store and ActiveMatrix BusinessWorks™ Domain Data Store.
- This script may take up to 10 or 15 minutes to complete. To see how long it takes to run the sample, run the `time runAll.sh` command to measure the time the script takes to complete.
- If you don't want to run all the samples at the same time, then run the following commands:

1. `bootstrap.sh -clean`



Running the `bootstrap.sh -clean` with the `-clean` option cleans the TIBCO Enterprise Administrator server and the ActiveMatrix BusinessWorks™ domain data store. Use the `-clean` option only if you want to clear the TIBCO Enterprise Administrator server and ActiveMatrix BusinessWorks™ domain data store.

2. `runBookStore.sh` - to run REST BookStore sample , or
3. `runSamples.sh` - to run other samples

10. Open a web browser and access the Admin UI at <http://localhost:8777/tea>

11. Log in using:

**Username:** admin

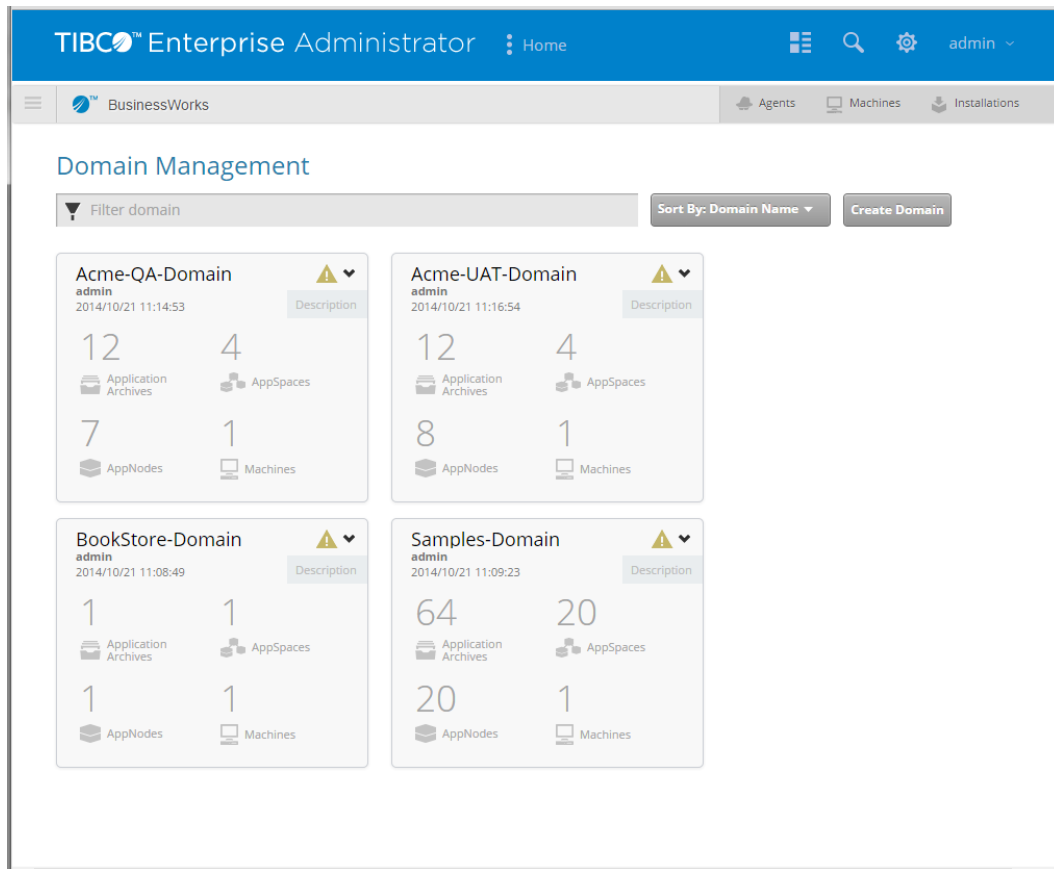
**Password:** admin

12. Click **BusinessWorks** in the **Products** list to see the following domains in the **Domain Management** screen.

- Acme-QA-Domain
- Acme-UAT-Domain
- BookStore-Domain
- Samples-Domain

Drill down into any of the domains to see the created runtime entities, such as AppSpaces, AppNodes, application archives, and applications.





## Testing the Deployed RESTful BookStore Application from Admin UI

The runAll script creates four domains, with multiple AppSpaces and AppNodes. The instructions in this topic show how to navigate the Admin UI to view the runtime entities.

### Procedure

1. In the TIBCO® Enterprise Administrator browser window, click **TIBCO Enterprise Administrator** at the top of the page to open the home page.
2. Click the **BusinessWorks** icon in the **Products** list to display the **Domains** page. The **Domains Management** page displays.

The screenshot shows the 'Domain Management' interface in TIBCO Enterprise Administrator. It features a search bar, a 'Sort By: Domain Name' dropdown, and a 'Create Domain' button. Below are four domain cards, each with a warning icon and a 'Description' link. The data for each domain is as follows:

Domain Name	Application Archives	AppSpaces	AppNodes	Machines
Acme-QA-Domain	12	4	10	1
Acme-UAT-Domain	12	4	9	1
BookStore-Domain	1	1	1	1
Samples-Domain	64	20	20	1

3. Click **BookStore-Domain** to drill down into domain details, then click **Monitor** to see the dashboard.

The screenshot shows the 'Monitor' dashboard for the 'BookStore-Domain'. The main area features a large green donut chart with the number '1' and the text 'AppSpaces' inside. Below the chart, it indicates '1 Running'. At the bottom, there are three smaller icons representing 'AppSpaces', 'AppNodes', and 'Applications'. On the left, a vertical sidebar contains navigation icons for 'Monitor', 'Application Archives', 'AppSpaces', 'Applications', 'AppNodes', 'Agents', 'Machines', and 'Installations'.

4. Click an entity in the side bar, such as **AppNodes**, to pivot views.

TIBCO Enterprise Administrator BusinessWorks

BusinessWorks BookStore-Domain Help Agents Machines Installations

AppNodes (1) Last updated 15:17:47

Create AppNode Delete Filter

Name	Status	Actions	AppSpace	Config State	Uptime	Applications	Machine	Agent
binding-rest-AppNode-6001	Running		binding-rest-AppSpa...	In sync	0d 00:00:43	1	WINAA-2	WINAA-2

- To view the BookStore application deployed earlier, select **Applications** on the left. A single application archive is displayed.

Application Archives (1) Last updated 15:19:36

Upload Delete Filter by name

Name	Version	App name	Uploaded on	Description
tibco.bw.sample.binding.rest.BookStore...	1.0.0.201411131234	tibco.bw.sample.binding.rest.BookStore...	2015/03/11 14:56:40	Using REST to Manage Books for a Bookstore

- Go to **Applications** to view the deployed applications

Applications (1) Last updated 15:20:29

Undeploy Filter

Name	Status	Actions	Version	Description	AppSpace	Deployment State	AppNodes
tibco.bw.sample.binding.rest...	Running		1.0	tibco bw sample binding rest B...	binding-rest-AppSpace	Deployed	1

- In the Applications view, click the link for **REST Doc URL**.

The screenshot shows the TIBCO Enterprise Administrator BusinessWorks interface. The top navigation bar includes the TIBCO logo, 'Enterprise Administrator', and 'BusinessWorks'. Below the navigation bar, the application name 'tibco.bw.sample.binding.rest.BookStore.application' is displayed, along with its status 'Running' and a 'Last updated' timestamp of 15:22:01. The interface includes a sidebar with navigation options like Monitor, Application Archives, AppSpaces, Applications, AppNodes, and Agents. The main content area shows the application's configuration, including 'App Instances' (Total: 1, Min: 1, Running: 1), 'Version' (1.0), 'AppSpace' (binding-rest-AppSpace), 'Jobs' (Created: 2, Running: 0, Faulted: 1), and 'Profile' (WindowsProfile). The 'Deployment State' is 'Deployed', and the 'REST Doc URL' is 'http://WINAA-2:7777/tibco.bw.sample.binding.rest.BookStore.application'. The 'Description' is 'tibco bw sample binding rest BookStore Application'. Below this, there are tabs for 'General' and 'App Instances'. The 'App Instances' tab is active, showing a table with columns for 'Property', 'Type', and 'Actual Value'. The table contains one entry: 'tibco.bw.sample.binding.rest.BookStore' with the property 'BW.APPNODE.NAME' of type 'string' and value 'ABC'.

8. Test the deployed application in the Google Chrome browser using Swagger UI. Open **localhost:7777/tibco.bw.sample.binding.rest.BookStore.application**.

The screenshot shows the Swagger UI interface for the BookStore application. The top navigation bar includes the TIBCO logo, the URL 'http://localhost:7777/tibco.bw.sample.binding.rest.BookStore', an 'api\_key' input field, and an 'Explore' button. Below the navigation bar, there are input fields for 'For Basic Auth only: username' and 'password'. The main content area is divided into two sections: 'Books' and 'Events'. The 'Books' section lists several REST operations: GET /book/{ISBN}, PUT /book/{ISBN}, DELETE /book/{ISBN}, POST /books, GET /books, and GET /book/{ISBN}/events. The 'Events' section lists several REST operations: GET /event/{EventID}, PUT /event/{EventID}, DELETE /event/{EventID}, POST /events, and GET /events. Each operation is represented by a colored button (blue for GET, orange for PUT, red for DELETE, green for POST) and a corresponding URL.

9. To execute the REST operations exposed by the BookStore application, click the **GET/books** icon and then click **Try it Out!**.  
This displays a list of all books. Locate and copy an ISBN.
10. Click the **GET/books{ISBN}** icon to get a book by its ISBN.
11. Enter the ISBN and then click **Try it Out!**.  
The book details can be seen in the Response Body.

## Defining and Deploying Multiple AppSpaces and AppNodes

The runAll script defines multiple AppSpaces and AppNodes and deploys multiple applications to these AppNodes. This topic shows how to navigate the Admin UI to locate these runtime entities.

### Procedure

1. Navigate to the **Domain Management** page by clicking the **TIBCO Enterprise Administrator** icon and choosing **BusinessWorks** in the **Products** area.
2. Choose the **Acme-QA-Domain** to see the **Monitor** page that shows the status of AppSpaces, AppNodes, and applications.
3. Click **AppSpaces** in the side bar to display the **AppSpaces** page. (You can also click the **AppSpace** icon on the **Monitor** page.) Change the view of the page by clicking the icons in the upper right of the screen.

Name	MinNodes	Status	Actions	Description	Deployment State	Applications	AppNodes	Machines
acme.acct.AppSpace	2	Running		acme.acct.AppSpace	In sync	2	2	1
acme.fulfillment.AppSpace	2	Stopped		acme.fulfillment.AppSpace	In sync	2	2	1
acme.hr.AppSpace	2	Stopped		acme.hr.AppSpace	In sync	2	2	1
acme.mkt.AppSpace	2	Stopped		acme.mkt.AppSpace	In sync	5	2	1

4. Now, view the AppNodes. Click **AppNodes** in the side bar.

Name	Status	Actions	AppSpace	Config State	Uptime	Applications	Machine	Agent
acme.acct.AppNode-5010	Running		acme.acct.AppSpace	In sync	00:00:412	2	NW00Q-VS30	NW00Q-VS30.ca.sba0.com
acme.acct.AppNode-5011	Running		acme.acct.AppSpace	In sync	00:00:412	2	NW00Q-VS30	NW00Q-VS30.ca.sba0.com
acme.fulfillment.AppNode-5012	Stopped		acme.fulfillment.AppSpace	In sync	00:00:000	2	NW00Q-VS30	NW00Q-VS30.ca.sba0.com
acme.fulfillment.AppNode-5013	Stopped		acme.fulfillment.AppSpace	In sync	00:00:000	2	NW00Q-VS30	NW00Q-VS30.ca.sba0.com
acme.hr.AppNode-5014	Stopped		acme.hr.AppSpace	In sync	00:00:000	2	NW00Q-VS30	NW00Q-VS30.ca.sba0.com
acme.hr.AppNode-5015	Stopped		acme.hr.AppSpace	In sync	00:00:000	2	NW00Q-VS30	NW00Q-VS30.ca.sba0.com
acme.mkt.AppNode-5016	Stopped		acme.mkt.AppSpace	In sync	00:00:000	5	NW00Q-VS30	NW00Q-VS30.ca.sba0.com
acme.mkt.AppNode-5017	Stopped		acme.mkt.AppSpace	In sync	00:00:000	5	NW00Q-VS30	NW00Q-VS30.ca.sba0.com

5. View the application archives by clicking **Application Archives** in the side bar.

Name	Version	App name	Uploaded on	Description
acme.mkt.retail.mgmt.file.application.ear	1.0.0.201309261119	acme.mkt.retail.mgmt.file.application	2014-05-06 14:09:33	
acme.mkt.retail.mgmt.application.ear	1.0.0.201309261118	acme.mkt.retail.mgmt.application	2014-05-06 14:09:33	
acme.mkt.loyalty.mgmt.application.ear	1.0.0.201309261118	acme.mkt.loyalty.mgmt.application	2014-05-06 14:09:33	
acme.mkt.inventory.mgmt.application.ear	1.0.0.201309261118	acme.mkt.inventory.mgmt.application	2014-05-06 14:09:33	
acme.mkt.distribution.mgmt.application.ear	1.0.0.201309261118	acme.mkt.distribution.mgmt.application	2014-05-06 14:09:33	
acme.hr.payroll.application.ear	1.0.0.201309261115	acme.hr.payroll.application	2014-05-06 14:09:33	
acme.hr.hiring.application.ear	1.0.0.201309261114	acme.hr.hiring.application	2014-05-06 14:09:33	
acme.hr.benefits.mgmt.application.ear	1.0.0.201309261114	acme.hr.benefits.mgmt.application	2014-05-06 14:09:33	
acme.fulfillment.iphone.prov.application.ear	1.0.0.201309261114	acme.fulfillment.iphone.prov.application	2014-05-06 14:09:33	
acme.fulfillment.android.prov.application.ear	1.0.0.201309261114	acme.fulfillment.android.prov.application	2014-05-06 14:09:33	
acme.acct.ar.application.ear	1.0.0.201309261113	acme.acct.ar.application	2014-05-06 14:09:32	
acme.acct.ap.application.ear	1.0.0.201309261113	acme.acct.ap.application	2014-05-06 14:09:32	

6. Execute the `killall.sh` command to shut down all running components.
7. Execute the `bwc1ean.sh` command to clean up.

## Core Admin Sample Scripts

The sample scripts provide a simple and fast way to run the core Admin samples.

These are bash scripts. On Windows, install Cygwin64.

Admin scripts are located in the following folders: `$BW_HOME/samples/core/admin` and `$BW_HOME/scripts/admin`

For information about running the sample scripts, see the "Running Admin Sample Scripts" section in the *TIBCO ActiveMatrix BusinessWorks™ Getting Started* guide.

This sets the `TIBCO_HOME`, `BW_HOME`, `TEA_HOME`, `EMS_HOME`, and `JAVA_HOME` environment variables necessary to run the admin scripts.

All scripts support the `-h` and command-line argument with full documentation of what each script does.

### Location of the Admin Scripts

The admin scripts are located in the following folders:

- The sample scripts are located in `$BW_HOME/samples/core/admin`
- The scripts that are generic for Activematrix BusinessWorks™ are located in `$BW_HOME/scripts/admin`



The scripts are updated to rely on the `PATH` setting to find the generic scripts. To make this easier to configure, after installation you can generate `$BW_HOME/scripts/bashrc.sh` that can be sourced from your `~/ .bashrc`.


Source the `$BW_HOME/scripts/bashrc.sh` to setup the following environment variables required to run the scripts mentioned in the table below:

Variable	Required
TIBCO_HOME	Yes
BW_HOME	Yes
TEA_HOME	No. But required if you run TIBCO® Enterprise Administrator on this machine.
EMS_HOME	No. But required if Enterprise Message Service™ is configured on this machine.
PATH	This variable is auto-populated based on the values set for the above variables.

### Core Admin Scripts

The following table lists some of the available scripts; browse the folder to see the complete list.

Script	Description	Script Location
<b>AppManage.sh</b>	<p>This is a ActiveMatrix BusinessWorks™ 6.x utility program that emulates ActiveMatrix BusinessWorks™ 5.x AppManage commands.</p> <p>The main purpose of this utility is to demonstrate how the AppManage commands from ActiveMatrix BusinessWorks™ 5.x translate to corresponding TIBCO ActiveMatrix BusinessWorks 6.x bwadmin commands.</p> <p>This utility creates a cmd/AppManage_deploy.cmd folder that contains bwadmin commands and uses bwadmin -f cmd/AppManage_deploy.cmd to run it.</p> <div style="border-left: 1px solid black; padding-left: 10px; margin-left: 20px;">  Not all AppManage commands are implemented in this emulation utility. </div> <p><b>ActiveMatrix BusinessWorks™</b>  <b>Augmented Options:</b></p> <ul style="list-style-type: none"> <li>• -appSpace or - a - AppSpace name to be used for Application lifecycle.</li> <li>• -profile or -p - Configuration Profile to use for deployment. This profile must be available in the EAR file.</li> <li>• -profileFile- Configuration Profile file to use for deployment.</li> <li>• -debug - Turn on debug tracing for this utility.</li> <li>• -sapp - Single Application per AppSpace deployment mode. Each AppSpace supports only one application deployment.</li> <li>• -mapp - Multiple Applications per AppSpace deployment mode. Each AppSpace supports one or more application deployments.</li> </ul> <div style="border-left: 1px solid black; padding-left: 10px; margin-left: 20px;">  ActiveMatrix BusinessWorks™ supports both -sapp and -mapp modes. The default is -mapp mode. </div>	<p><i>\$BW_HOME/samples/core/admin</i></p>

Script	Description	Script Location
<b>bootstrap.sh</b>	<p>Usage: <b>bootstrap.sh</b> [-h -help] [-clean] [-forceClean -force -forceclean]</p> <p>This utility is a wrapper script around the following scripts:</p> <ul style="list-style-type: none"> <li>• killtea.sh</li> <li>• killbwagent.sh</li> <li>• teaclean.sh only if -clean or -forceClean options is used.</li> <li>• bwclean.sh if and only if -clean or -forceClean options is used.</li> <li>• genbwagentini.sh</li> <li>• tea.sh</li> <li>• bwagent.sh</li> <li>• registeragent.sh</li> </ul> <p>[-h] or [-help] - Prints this usage message.</p> <p>-clean Cleans TIBCO Enterprise Administrator Server Data Store and ActiveMatrix BusinessWorks™ Domain Data Store.</p> <div style="border-left: 1px solid black; padding-left: 10px; margin-left: 20px;">  <p>The -clean command on the data store is not reversible, so back up your data stores before using the command. Use this option carefully, as you may lose all your configurations if you do not have a backup.</p> </div> <p>-forceClean Same as -clean, except it avoids prompting user to confirm with clean.</p> <p>-force Same as -forceClean</p> <p>-forceclean Same as -forceClean</p> <p>This script assumes that the following products are installed correctly and the environment variables are set accordingly:</p> <p><i>TIBCO_HOME</i> = TIBCO_HOME directory where you installed ActiveMatrix Businessworks™.</p> <p><i>TEA_HOME</i> = Parent directory to TIBCO Enterprise Administrator's /bin directory.</p> <p>Supports generation of bwagent.ini file for either Database/ TIBCO EMS™, or</p>	<i>\$BW_HOME/scripts/admin</i>



Script	Description	Script Location
	Database/ TIBCO FTL® as the technology type.	
<b>bounce.sh</b>	<p>This utility does the following:</p> <ol style="list-style-type: none"> <li>1. Stops TIBCO Enterprise Administrator Server and bwagent Processes.</li> <li>2. Restarts TIBCO Enterprise Administrator Server and bwagent Processes.</li> <li>3. Registers bwagent to TIBCO Enterprise Administrator Server.</li> </ol> <p>[-h] or [-help] - Prints this help message and exits.</p>	\$BW_HOME/scripts/admin
<b>bounceagent.sh</b>	<p>Kills and restarts bwagent Process.</p> <p>[-h] or [-help] - Prints this help message and exits.</p>	\$BW_HOME/scripts/admin

Script	Description	Script Location
<b>bwadmin.sh</b>	<p>This is a utility script that wraps around the bwadmin executable.</p> <p>[-h] or [-help] - Prints this help message and exits.</p> <p>[-network &lt;bwagent Network Name&gt;] - Connects to a named bwagent Network. This is an optional argument.</p> <p>By default, this script uses \$BW_HOME/config/bwagent.ini</p> <p>[&lt;bwadminArgs&gt; ...] - Use bwadmin to run commands found in the input files.</p> <p>Start bwadmin in the interactive mode if cmdFile is not specified.</p> <p>A bwagent Network Name is a named directory under \${TIBCO_HOME}/bw/networks and contains the corresponding bwagent.ini.</p> <p><b>How to Set Up a Newly Named Network</b></p> <ol style="list-style-type: none"> <li>1. Obtain a bwagent.ini created for the named bwagent network. For example, a named network called "acmeNetwork"</li> <li>2. Create the acmeNetwork directory under \${TIBCO_HOME}/bw/networks. For example, mkdir \$ {TIBCO_HOME}/bw/networks/acmeNetwork</li> <li>3. Copy bwagent.ini to the above directory.</li> <li>4. Rerun bwadmin.sh -network acmeNetwork</li> </ol>	<p>\$BW_HOME/scripts/admin</p>

Script	Description	Script Location
<b>bwagent.sh</b>	<p>This script starts bwagent in the background and waits until it is fully initialized, or the maxWait time (&lt;n&gt; * 2 sec) expires.</p> <p><b>[-h] or [-help]</b> - Prints this usage message.</p> <p><b>[-network &lt;Network&gt;]</b> - Starts up bwagent using the configuration of a named network.</p> <p><b>[-maxWait &lt;n&gt;]</b> - Maximum amount of wait time (2 sec increment) for bwagent start up success.</p> <p>The dDefault value for &lt;n&gt; is 30, which means 30 * 2 sec = 60 seconds</p>	\$BW_HOME/scripts/admin
<b>bwclean.sh</b>	<p>This utility script cleans up ActiveMatrix BusinessWorks™ Domain Data and internal Data Store. The end effect of this clean up is similar to a fresh installation of ActiveMatrix BusinessWorks™.</p> <p><b>[-force] or [-forceClean]</b> - Proceeds with wiping ActiveMatrix BusinessWorks™ Domain Data and internal Data Store without prompting user reconfirmation.</p> <p>By default, the script prompts user confirmation.</p>	\$BW_HOME/scripts/admin

Script	Description	Script Location
<b>configureBWEngineGroup.sh</b>	<p>This utility configures AppNodes in a Domain/AppSpace to form a fault-tolerant group and cross engine persistence</p> <p>[-h] or [-help] - Prints this usage message.</p> <p>[-c] or [-clean] - Cleans up and drops all the previously configured database tables.</p> <p>Use this option carefully. This operation cannot be undone. Do not specify both -setup and -cleanup on the same run.</p> <p>[-s] or [-setup] - Does the one time setup of bwengine Database. When this option is used, -domain and -appspace arguments are not needed and are not used even if specified. <code>\${BW_HOME}/config/sqlscripts/&lt;dbtype&gt;/create.sql</code> is used to set up the database tables and configuration.</p> <p>[-b] or [-bootstrap] - Does clean up then setup.</p> <p>[-t] or [-dbtype] - This is the default value is postgresql.</p> <p>-cf &lt;config.sh&gt; - Sources configuration from the specified &lt;config.sh&gt; file.</p> <p>By default, <code>&lt;BW_HOME&gt;/scripts/admin/config/bwengine-group-&lt;dbtype&gt;.sh</code></p> <p>[-d] or [-domain] - Domain Name</p> <p>[-a] or [-appspace &lt;appspace&gt;] - AppSpace Name. All AppNodes in the specified Domain and AppSpace are configured to form a Fault-Tolerant group and across engine persistence.</p>	<code>\$BW_HOME/scripts/admin</code>


Script	Description	Script Location
<b>deploy.sh</b>	<p>Usage: <b>deploy.sh</b> <b>-ear</b> &lt;EARFile&gt; [<b>-h</b>   <b>-help</b>] [<b>-domain</b> &lt;DomainName&gt;] [<b>-appspace</b> &lt;AppSpaceName&gt;] [<b>-redeploy</b>   <b>-force</b>] [<b>-profile</b> &lt;Profile&gt;]</p> <p>Deploys the specified ActiveMatrix BusinessWorks™ EAR File into -domain &lt;DomainName&gt; -appspace &lt;AppSpaceName&gt;</p> <p>[<b>-h</b>] or [<b>-help</b>]- Prints this help message.</p> <p><b>-ear</b> &lt;EARFile&gt; - Enterprise Archive file to deploy</p> <p>[<b>-domain</b> &lt;DomainName&gt;] - Domain Name - Optional parameter</p> <p>If it is not specified, DomainName is computed from \${USER}-Domain</p> <p>This utility creates the Domain if it does not already exist.</p> <p>[<b>-appspace</b> &lt;AppSpaceName&gt;] - AppSpace Name - Optional parameter</p> <p>If it is not specified, AppSpaceName is computed from the name of the EAR file.</p> <p>This utility creates the AppSpace and AppNode if they do not already exist.</p> <p>[<b>-redeploy</b>   <b>-force</b>] - Redeploy if the application has been previously deployed.</p> <p>The application is not redeployed if it already exists and this option is not specified.</p> <p>[<b>-profile</b> &lt;Profile&gt;] : Profile name to use for this deployment.</p> <p>If it is not specified, the default Profile as packaged in the Enterprise Archive file is used.</p> <p>[<b>-mapp</b>] - Optional flag to set Multiple Applications per AppSpace Mode. This is the default mode for ActiveMatrix BusinessWorks™.</p> <p>[<b>-debug</b>] - Prints debug tracing for this script <b>./deploy.sh</b></p>	<i>\$BW_HOME</i> /scripts/admin


Script	Description	Script Location
<b>genbwagentini.sh</b>	<p>This script auto generates <code>\${BW_HOME}/config/bwagent.ini</code> based on configurations defined in <code>./config/bwadmin-default-config.sh</code></p> <p><code>-h</code> or <code>-help</code> - Prints this help message.</p> <p>The following variables are required from <code>./config/bwadmin-default-config.sh</code>:</p> <ul style="list-style-type: none"> <li>• <code>BWAgentNetworkName</code> - Name of BWAgent Network.</li> <li>• <code>BWMachines</code> - Defined as a list of machine names (as obtained through <code>hostname -f</code>). If you have only one machine to configure, do not add it to this list because this script auto-configures it as a standalone BWAgent Network.</li> </ul> <p>This script uses <code>hostname -f</code> to determine the name of the machine it is run on. It then determines whether this machine is in the <code>BWMachines</code> list.</p> <p>You can assume that the <code>discoveryURL</code> of the <code>bwagent.ini</code> is comparable to that of a Database Server's URL, and <code>BWAgentNetworkName</code> is then comparable to the Database Name. You can configure both to uniquely access the specific instance of the Database.</p> <p>If the <code>KEEP_BWAGENT_INI</code> environment variable is defined, <code>bwagent.ini</code> generation is skipped.</p> <p>You can edit either the <code>./config/bwadmin-default-config.sh</code> file, or make a copy of it, edit it, and then set environment variable <code>BWADMIN_CONFIG</code> to point to it. For example, export <code>BWADMIN_CONFIG=~/.config/bwadmin-my-config.sh</code></p> <p>Generates <code>bwagent.ini</code> file for either Database/EMS, or Database/ TIBCO FTL<sup>®</sup> as the technology type.</p>	<code>\$BW_HOME/scripts/admin</code>



Script	Description	Script Location
<b>kill.sh</b>	Kills all processes that match the specified <i>&lt;process name&gt;</i>  -h or -help - Prints this help message.  <i>&lt;process name&gt;</i> - name of the process you want to kill. This script kills all instances of the pid that matches this name.	<i>\$BW_HOME/scripts/admin</i>
<b>killall.sh</b>	This script finds and kills all instances of processes that match the following names:  <ul style="list-style-type: none"> <li>• tea</li> <li>• bwagent</li> <li>• bwappnode</li> <li>• bwadmin</li> </ul> -h or -help - Prints this help message.	<i>\$BW_HOME/scripts/admin</i>
<b>killbwagent.sh</b>	This script finds and kills all instances of processes that match "bwagent".  -h or -help - Prints this help message.	<i>\$BW_HOME/scripts/admin</i>
<b>killbwappnodes.sh</b>	This script finds and kills all instances of processes that match "bwappnode".  -h or -help - Prints this help message.	<i>\$BW_HOME/scripts/admin</i>
<b>killtea.sh</b>	This script finds and kills all instance of processes that matches "tea".  -h or -help - Prints this help message.	<i>\$BW_HOME/scripts/admin</i>
<b>killtibemsd64.sh</b>	This script finds and kills all instances of processes that match "tibemsd".  -h or -help - Prints this help message.	<i>\$BW_HOME/scripts/admin</i>
<b>recreatedb.sh</b>	This script cleans up and recreates the Postgres DB needed by ActiveMatrix BusinessWorks™ BookStore REST sample located in: <i>{BW_HOME}/samples/binding/rest/BookStore</i>  -h and -help - Prints this help message.	<i>\$BW_HOME/samples/core/admin</i>

Script	Description	Script Location
<b>registeragent.sh</b>	<p>This utility registers the local bwagent with TIBCO Enterprise Administrator server.</p> <p>-h or -help - Prints this help message.</p> <p>This utility assumes that the following environment variables have been set:</p> <pre>export TIBCO_HOME="<i>&lt;Where ActiveMatrix BusinessWorks™ is installed&gt;</i>"</pre> <p>At least one of the following environment variable is set:</p> <pre>export TEA_HOME="<i>Where TIBCO Enterprise Administrator is installed in the form of \$TIBCO_HOME/tea/&lt;version&gt;</i>"</pre> <p>Or,</p> <pre>export TEA_HOSTNAME=<i>&lt;HostName&gt;</i></pre> <p>If <i>TEA_HOSTNAME</i> environment variable is set, it assumes the TIBCO Enterprise Administrator server is running remotely from the local bwagent instance.</p> <p>If <i>TEA_HOSTNAME</i> environment variable is not set, this script registers the local bwagent to the locally running TIBCO Enterprise Administrator server.</p>	<i>\$BW_HOME/scripts/admin</i>



Script	Description	Script Location
<p><b>runAcme.sh</b></p>	<p>Creates &lt;domain&gt; and deploys all EAR files found under <code>\${BW_HOME}/samples/core/admin/ears/acme</code>.</p> <p>-h or -help : Displays this usage message</p> <p>&lt;domain&gt; - can be "Acme-QA-Domain" or "Acme-UAT-Domain". When not specified, the default is "Acme-QA-Domain"</p> <p>&lt;mode&gt; - [-sapp ] or [-mapp]</p> <p>-sapp- Single App AppSpace deployment mode. Each AppSpace supports only one application deployment.</p> <p>-mapp - Multiple App AppSpace deployment mode. Each AppSpace supports one or more application deployment.</p> <div style="border-left: 1px solid black; padding-left: 10px; margin-left: 20px;">  <p>ActiveMatrix BusinessWorks™ supports both -sapp and -mapp modes. The default is -mapp mode.</p> </div> <p>This script dynamically creates a <b>bwadmin</b> command file in <code>cmd/&lt;domain&gt;-&lt;mode&gt;.cmd</code> and executes it.</p>	<p><code>\$BW_HOME/samples/core/admin</code></p>

Script	Description	Script Location
<b>runAll.sh</b>	<p>This utility is a wrapper script that performs the following:</p> <ul style="list-style-type: none"> <li>• bootstrap.sh - only if running in a single machine setup</li> <li>• runBookStore.sh</li> <li>• runSamples.sh</li> <li>• runAcme.sh -domain Acme-QA-Domain</li> <li>• runAcme.sh -domain Acme-UAT-Domain</li> </ul> <p>-h or -help - Displays this usage message and exits</p> <p>-clean - Cleans the TIBCO Enterprise Administrator Server Data Store and ActiveMatrix BusinessWorks™ Domain Data Store.</p> <p>These data store clean is not reversible. Make sure you back up your data stores before running this command. Use this option with utmost care, otherwise you risk losing all your configurations.</p> <p>-forceClean - Same as -clean, except it avoids prompting you to confirm with clean.</p> <p>-force - Same as -forceClean</p> <p>&lt;mode&gt; - [-sapp   -mapp]</p> <p>-sapp - Single Application per AppSpace deployment mode. Each AppSpace supports only one application deployment.</p> <p>-mapp - Multiple Applications per AppSpace deployment mode. Each AppSpace supports one or more application deployments.</p> <div style="border-left: 1px solid black; padding-left: 10px; margin-left: 20px;">  ActiveMatrix BusinessWorks™ supports both -sapp and -mapp modes. The default is -mapp mode. </div> <p>Generates the bwagent.ini file for either Database/EMS™, or Database/FTL® technology type.</p>	<p><i>\$BW_HOME/samples/core/admin</i></p>

Script	Description	Script Location
<b>runBookStore.sh</b>	<p>Creates BookStore-Domain and deploys all EAR files found under <code>\${BW_HOME}/samples/core/admin/ears/bookstore</code></p> <p><code>-h</code> or <code>-help</code> - Displays this usage message.</p> <p><code>&lt;mode&gt;</code> - [<code>-sapp</code>   <code>-mapp</code>]</p> <p><code>-sapp</code> - Single Application per AppSpace deployment mode. Each AppSpace supports only one application deployment.</p> <p><code>-mapp</code> - Multiple Application per AppSpace deployment mode. Each AppSpace supports one or more application deployments.</p> <p>ActiveMatrix BusinessWorks™ supports both <code>-sapp</code> and <code>-mapp</code> modes. The default is <code>-mapp</code> mode.</p> <div style="border-left: 1px solid black; padding-left: 10px; margin-top: 10px;">  This script dynamically creates a <b>bwadmin</b> cmd file in <code>cmd/Samples-Domain-&lt;mode&gt;.cmd</code> and executes it. </div>	<code>\$BW_HOME/samples/core/admin</code>
<b>runSamples.sh</b>	<p>Creates Samples-Domain and deploys all EAR files found under <code>\${BW_HOME}/samples/core/admin/ears/samples</code></p> <p><code>-h</code> or <code>-help</code> - Displays this usage message.</p> <p><code>&lt;mode&gt;</code> : [<code>-sapp</code>] or [<code>-mapp</code>]</p> <p><code>-sapp</code> : Single Application per AppSpace deployment mode. Each AppSpace supports only one application deployment.</p> <p><code>-mapp</code> : Multiple Application per AppSpace deployment mode. Each AppSpace supports one or more application deployments.</p> <div style="border-left: 1px solid black; padding-left: 10px; margin-top: 10px;">  ActiveMatrix BusinessWorks™ supports both <code>-sapp</code> and <code>-mapp</code> modes. The default is <code>-mapp</code> mode. This script dynamically creates a <b>bwadmin</b> command file in <code>cmd/Samples-Domain-&lt;mode&gt;.cmd</code> and executes it. </div>	<code>\$BW_HOME/samples/core/admin</code>

Script	Description	Script Location
<b>showprocs.sh</b>	<p>Shows process ID and complete binary path of all processes required in ActiveMatrix BusinessWorks™:</p> <ul style="list-style-type: none"> <li>• tibemsd</li> <li>• tea</li> <li>• bwagent</li> <li>• bwappnode</li> <li>• bwadmin</li> </ul>	<i>\$BW_HOME/scripts/admin</i>
<b>tea.sh</b>	<p>This script starts TIBCO Enterprise Administrator in the background and waits until it is completely initialized, or the maxWait time (&lt;n&gt; * 2 sec) expires.</p> <p>-h or -help - Prints this usage message.</p> <p>[-maxWait &lt;n&gt;] - Max number of wait time (2 sec increment) for TIBCO Enterprise Administrator Server startup success.</p> <p>The default value for &lt;n&gt; is 30, which means 30 * 2 sec = 60 seconds.</p>	<i>\$BW_HOME/scripts/admin</i>
<b>teaclean.sh</b>	<p>This utility script cleans TIBCO Enterprise Administrator Server's configuration data store.</p> <p>The end effect of this clean up is similar to a fresh installation of TIBCO Enterprise Administrator.</p> <p>-h or -help - Prints this usage message</p> <p>-force or -forceClean - Proceeds with wiping ActiveMatrix BusinessWorks™ Domain Data and internal data store without prompting user reconfirmation.</p> <p>By default, the script prompts user confirmation.</p>	<i>\$BW_HOME/scripts/admin</i>

Script	Description	Script Location
<b>tibemsd64.sh</b>	<p>This script starts <code>tibemsd64</code> in the background and waits until it is completely initialized, or the <code>maxWait</code> time (<code>&lt;n&gt; * 2 sec</code>) expires.</p> <p><code>-h</code> or <code>-help</code> - Prints this usage message</p> <p><code>[-maxWait &lt;n&gt;]</code> - Max number of wait time (2 seconds increment) for <code>tibemsd64</code> start up success.</p> <p>The default value for <code>&lt;n&gt;</code> is 30, which means <code>30 * 2 sec = 60 seconds</code>.</p> <p>This script is only supported on UNIX based systems.</p> <p>For Windows, use Windows Systems Services to start or stop <code>tibemsd64</code>.</p>	<code>\$BW_HOME/scripts/admin</code>



Each `runAcme.sh`, `runBookStore.sh`, `runSamples.sh`, `deploy.sh`, and `AppManage.sh` generates `bwadmin` commands before execution.

The generated `bwadmin` command files are found under `cmd` subdirectory.

## Legal and Third-Party Notices

---

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

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

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

TIBCO, the TIBCO logo, the TIBCO O logo, TIBCO ActiveMatrix BusinessWorks, TIBCO Rendezvous, TIBCO Enterprise Message Service, TIBCO Business Studio for BusinessWorks, TIBCO Enterprise Administrator, TIBCO ActiveSpaces, TIBCO Runtime Agent, and TIBCO Designer are either registered trademarks or trademarks of TIBCO Software Inc. in the United States and/or other countries.

Java and all Java based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

All other product and company names and marks mentioned in this document are the property of their respective owners and are mentioned for identification purposes only.

This software may be available on multiple operating systems. However, not all operating system platforms for a specific software version are released at the same time. Please see the readme.txt file for the availability of this software version on a specific operating system platform.

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

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

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

This and other products of TIBCO Software Inc. may be covered by registered patents. Please refer to TIBCO's Virtual Patent Marking document (<https://www.tibco.com/patents>) for details.

Copyright © 2001-2020. TIBCO Software Inc. All Rights Reserved.