TIBCO ActiveMatrix BusinessWorks™ Service Engine Plug-in for SmartMapper

User's Guide

Software Release 5.5
June 2011

TIBCO®
The Power of Now®

Important Information

Contents

Figures

Tables

Preface

This user's guide explains how to use TIBCO ActiveMatrix BusinessWorks Service Engine Plug-in for SmartMapper in TIBCO ActiveMatrix Environment.

Topics

Related Documentation

This section lists documentation resources you may find useful.

TIBCO ActiveMatrix BusinessWorks Service Engine Plug-in for SmartMapper Documentation

The following documents form the TIBCO ActiveMatrix BusinessWorks Service Engine Plug-in for SmartMapper documentation set:

- *TIBCO ActiveMatrix BusinessWorks Service Engine Plug-in for SmartMapper Installation Guide* Read this manual for instructions on site preparation and installation.
- *TIBCO ActiveMatrix BusinessWorks Service Engine Plug-in for SmartMapper User's Guide* Read this manual for instruction on using the product.
- *TIBCO ActiveMatrix BusinessWorks Service Engine Plug-in for SmartMapper Release Notes* Read the release notes for a list of new and changed features. This document also contains lists of known issues and closed issues for this release.

Other TIBCO Product Documentation

You may find it useful to read the documentation for the following TIBCO products:

- TIBCO Designer™
- TIBCO Administrator™
- TIBCO ActiveMatrix BusinessWorks™
- TIBCO ActiveMatrix BusinessWorks™ Service Engine
- TIBCO Rendezvous®
- TIBCO Enterprise Message Service™
- TIBCO Hawk®
- TIBCO Runtime Agent™
- TIBCO Business Studio™
- TIBCO ActiveMatrix BusinessWorks™ SmartMapper

Typographical Conventions

The following typographical conventions are used in this manual.

Table 1 General Typographical Conventions

Convention	Use
TIBCO_HOME *ENV_HOME*	Many TIBCO products must be installed within the same home directory. This directory is referenced in documentation as *TIBCO_HOME*. The default value of *TIBCO_HOME* depends on the operating system. For example, on Windows systems, the default value is `C:\tibco`. Other TIBCO products are installed into an *installation environment*. Incompatible products and multiple instances of the same product are installed into different installation environments. An environment home directory is referenced in documentation as *ENV_HOME*. The default value of *ENV_HOME* depends on the operating system. For example, on Windows systems, the default value is `C:\tibco`.
`code font`	Code font identifies commands, code examples, filenames, pathnames, and output displayed in a command window. For example: Use `MyCommand` to start the foo process.
`bold code font`	Bold code font is used in the following ways: • In procedures, to indicate what a user types. For example: Type the username **`admin`**. • In large code samples, to indicate the parts of the sample that are of particular interest. • In command syntax, to indicate the default parameter for a command. For example, if no parameter is specified, `MyCommand` is enabled: `MyCommand [`**`enable`**` | disable]`
italic font	Italic font is used in the following ways: • To indicate a document title. For example: See *TIBCO BusinessWorks Concepts*. • To introduce new terms. For example: A portal page may contain several portlets. *Portlets* are mini-applications that run in a portal. • To indicate a variable in a command or code syntax that you must replace. For example: `MyCommand` *pathname*

Table 1 General Typographical Conventions

Convention	Use
Key combinations	Key name separated by a plus sign indicate keys pressed simultaneously. For example: Ctrl+C. Key names separated by a comma and space indicate keys pressed one after the other. For example: Esc, Ctrl+Q.
📝	The note icon indicates information that is of special interest or importance, for example, an additional action required only in certain circumstances.
💡	The tip icon indicates an idea that could be useful, for example, a way to apply the information provided in the current section to achieve a specific result.
⚠	The warning icon indicates the potential for a damaging situation, for example, data loss or corruption if certain steps are taken or not taken.

Table 2 Syntax Typographical Conventions

Convention	Use
`[ ]`	An optional item in a command or code syntax. For example: `MyCommand [optional_parameter] required_parameter`
`|`	A logical OR that separates multiple items of which only one may be chosen. For example, you can select only one of the following parameters: `MyCommand para1 | param2 | param3`

Table 2 Syntax Typographical Conventions

Convention	Use

`{ }`	A logical group of items in a command. Other syntax notations may appear within each logical group.
	For example, the following command requires two parameters, which can be either `param1` and `param2` or `param3` and `param4`:
	`MyCommand {param1 param2} | {param3 param4}`
	In the next example, the command requires two parameters. The first parameter can be either `param1` or `param2` and the second can be either `param3` or `param4`:
	`MyCommand {param1 | param2} {param3 | param4}`
	In the next example, the command can accept either two or three parameters. The first parameter must be `param1`. You can optionally include `param2` as the second parameter. And the last parameter is either `param3` or `param4`.
	`MyCommand param1 [param2] {param3 | param4}`

Connecting with TIBCO Resources

How to Join TIBCOmmunity

TIBCOmmunity is an online destination for TIBCO customers, partners, and resident experts, a place to share and access the collective experience of the TIBCO community. TIBCOmmunity offers forums, blogs, and access to a variety of resources. To register, go to http://www.tibcommunity.com.

How to Access All TIBCO Documentation

After you join TIBCOmmunity, you can access the documentation for all supported product versions here:

http://docs.tibco.com/TibcoDoc

How to Contact TIBCO Support

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

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

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

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

  https://support.tibco.com

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

Chapter 1 Introduction

This chapter gives an overview of TIBCO ActiveMatrix BusinessWorks Service Engine Plug-in for SmartMapper.

Topics

Overview

TIBCO ActiveMatrix BusinessWorks Service Engine (TIBCO AMX BWSE) is a gateway for TIBCO BusinessWorks to the Service Oriented Architecture (SOA) world. It implements all of the functionality of the BusinessWorks Engine, and it allows you to deploy those components into the TIBCO ActiveMatrix environment via the TIBCO ActiveMatrix Administrator.

TIBCO ActiveMatrix BusinessWorks Service Engine provides a BWSE component that can be associated with a BusinessWorks project to expose BusinessWorks services in the TIBCO ActiveMatrix environment. It also lets TIBCO ActiveMatrix BusinessWorks projects consume services provided by other components in the TIBCO ActiveMatrix environment.

Before starting this menu, it is recommended that you become familiar with:

- TIBCO ActiveMatrix platform terminology and concepts.

  Refer to *TIBCO ActiveMatrix BusinessWorks Service Engine User's Guide* for detailed information.

- TIBCO ActiveMatrix BusinessWorks SmartMapper functions and features.

  Refer to *TIBCO ActiveMatrix BusinessWorks SmartMapper User's Guide* for detailed information.

TIBCO ActiveMatrix Development Tools

The TIBCO ActiveMatrix development tools consist of TIBCO Business Studio Workbench and a set of TIBCO ActiveMatrix plug-ins.

TIBCO Business Studio allows you to create and configure composites, services, references, shared resources, and substitution variables in the Composite Editor. In TIBCO Business Studio, you can associate a BusinessWorks Designer project with a BWSE Component in an ActiveMatrix SOA Project. For information on TIBCO Business Studio, refer to the *Workbench User Guide* in the Workbench online help. To view the online help, select **Help > Help Contents**.

TIBCO Designer Add-in for TIBCO Business Studio allows you to create TIBCO Designer projects in TIBCO Business Studio.

TIBCO ActiveMatrix Deployment Tool

TIBCO ActiveMatrix Administrator is the utility used to create, configure, monitor, and manage objects in the TIBCO ActiveMatrix runtime.

Chapter 2 Working with TIBCO Business Studio

This chapter describes how to create a TIBCO Designer project, create a TIBCO SOA project, enable BusinessWorks service resource for TIBCO ActiveMatrix, and associate a BWSE component with the service resource using TIBCO Business Studio.

Topics

Starting TIBCO Business Studio

To start TIBCO Business Studio, complete the following steps:

1. From the Start menu, select **All Programs > TIBCO > TIBCO Business Studio *version_number* ActiveMatrix SOA Edition > Studio for Designers.**

   or

   From the command line, run *TIBCO_HOME*`\studio\`*version_number*`\eclipse\TIBCOBusinessStudio.exe`.

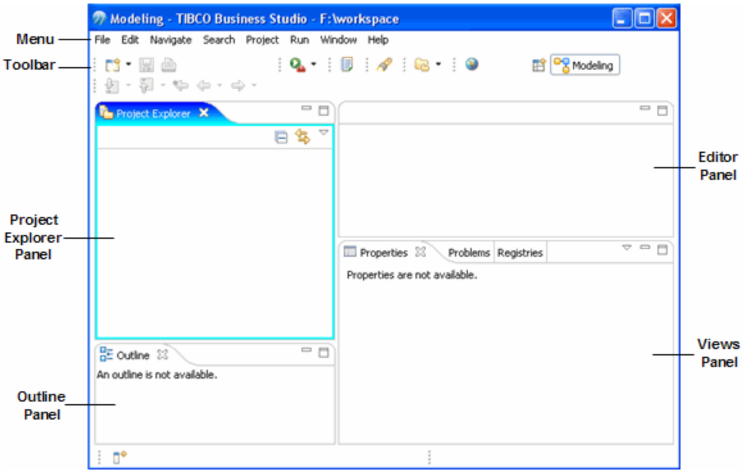
2. The Workspace Launcher dialog appears. Accept the default workspace or browse to create a new workspace, and then click the **OK** button.

   The Workbench window appears. The first time a new workspace is selected, a Welcome screen is displayed in the window.

The Layout of the Workbench Window

The Workbench window contains the following: Menu, Toolbar, Project Explorer Panel, Outline Panel, Editor Panel, and Views Panel. For detailed information, select **Help** > **Help Conten**t.

Figure 1 The Workbench Window



Properties are not available.

An outline is not available.

- **Menu**

  Contains menu items such as File, Edit, Navigate, Search, Project, Run, Window, and Help.

- **Toolbar**

  Contains buttons for the most frequently used commands.

- **Project Explorer Panel**

  Displays a tree containing all the project resources, such as project folders, shared resource definition files, WSDL files, composite files, service assembly files, and so on.

- **Editor Panel**

  Displays editors for the objects currently being edited. You can switch between editors by clicking tabs at the top of the Editor area. The Composite Editor contains a canvas on which you can drop elements and a palette that organizes the elements that you have added to the composite. Other editors allow you to configure shared resources and service assemblies.

- **Outline Panel**

  Provides an overview of the Composite Editor canvas. You can easily navigate from one part of a composite to another.

  The Outline view also displays a content tree structure that contains the composite elements inside the composite. In this view you can delete the contents of the composite. When you select a composite element in the Outline tree, the corresponding artifact in the composite is selected.

- **Views Panel**

  Displayed under the Editor area and contains the following views by default:

  — Properties

    Displays property sheets for editing composites and composite elements. When you select a composite or composite element in the Composite Editor canvas, this view shows the properties of the selected object in a vertical tabbed notebook.

  — Problems

    Displays validation and other errors.

  — Registries

    Lists UDDI registries and the WSDL files returned from searching a registry.

  To open a view, select **Window > Show View**.

Creating a TIBCO Designer Project

After starting TIBCO Business Studio, follow these steps to create a TIBCO Designer project:

1. Select **File > New > Project** from the Menu to open the New Project window.
2. Select **TIBCO Designer > TIBCO Designer Project** in the Select a Wizard page, and then click the **Next** button.

> Before you creat a new TIBCO Designer project, you need to install TIBCO ActiveMatrix BusinessWorks SmartMapper Plug-in or TIBCO ActiveMatrix BusinessWorks SmartMapper Enterprise Server.

— To create a new TIBCO Designer project in the workspace, select the **Create New Project in Workspace** radio button and type the designer project name (for example, `SmartMapperLookup_DESIGNER`) in the Project Name field.

— To create a new TIBCO Designer project using an existing Designer project, select the **Create Project Using Existing Designer Project** radio button and then click the **Browse** button to specify the project location.

3. Click the **Finish** button.

   The project appears in the Project Explorer panel.

Enabling BusinessWorks Service Resource for ActiveMatrix

In order to provide services to other ActiveMatrix components, ensure that the TIBCO ActiveMatrix BusinessWorks project with Service resources is available.

Complete the following steps to check if the Service resource is available:

1. Expand the project in Project Explorer.
2. Search for the BusinessWorks Service resource, for example, `lookupService.serviceagent`.

If the project does not contain a Service resource, complete the following steps to enable a BusinessWorks Service Resource:

> Before creating a Service resource, you will need to create a WSDL file that contains abstract interface description. The interface can include one or more operations. Refer to *TIBCO ActiveMatrix BusinessWorks Palette Reference* for details about how to create a WSDL file.

1. Right-click in the design panel and select **Add Resource** > **Service** > **Service**.
2. In the **Service** tab,
   a. Check the **Enable for AMX** checkbox.

   b. Click the ⊞ button (the Add A New Interface button) to select a resource.

   c. Expand the resource folder, select a WSDL resource file, and click the **OK** button in the Select A Resource dialog.

   d. Double-click the **Operation Implementation** field and use the Browse button to locate the resource that implements each operation.

      Process definitions that implement an operation must have an input, output, and error schemas that match the input, output, and fault messages of the operation.

   e. Select the **Endpoint Bindings** tab, and then click the ⊞ button (the Add New Endpoint button) to add the binding point automatically.

   f. In the Endpoint Bindings tab, specify a name for the endpoint in the Endpoint Name field. And specify the type of the endpoint by double-clicking the Endpoint Type field, and then selecting a type: AMX, SOAP, or Local.

   g. Check the **Expose** Checkbox in the Properties tab to expose the global variables used by resources in a TIBCO BusinessWorks project.

   h. Click the **Apply** button in the Service resource configuration panel.

Creating a TIBCO SOA Project

TIBCO SOA projects are Implementation Types for the ActiveMatrix resources that you can develop with TIBCO Business Studio.

To create a TIBCO SOA project, completing the following steps:

1. Select **File > New > Project** from the Menu to open the New Project dialog.
2. Expand the **TIBCO SOA Platform** folder, select **TIBCO SOA Project** in the Select A Wizard page, and then click the **Next** button.
3. Enter the SOA project name in the Project Name field, for example, `SmartMapperReference`.

   Check the **Use Default Location** checkbox if you want to save the SOA project to your default workspace, and then click the **Next** button.
4. In the Asset Type Selection screen, keep the default selection and click the **Next** button.
5. Select the **Empty SOA Project** type from the Project Types column, then click the **Next** button.

By selecting the Empty SOA Project type, an SOA project with an empty composite is created.

6. Click the **Next** button in the Composite File Name page.
7. Click the **Finish** button in the Set Special Folders page.

   The new SOA project appears in the Project Explorer panel and displays `SmartMapperReference.composite` in the design panel by default.

Associating the BWSE Component with Service Resource

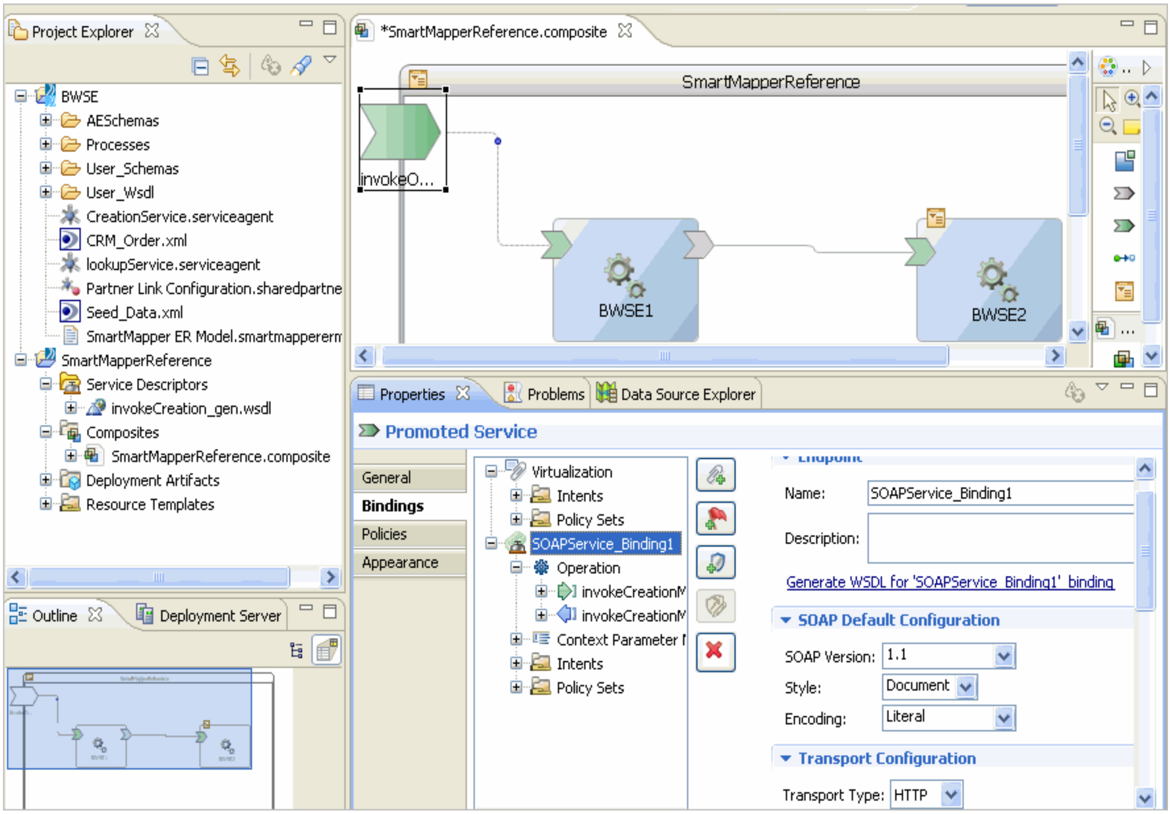
You can associate a BWSE Component with BusinessWorks Service resources by completing the following steps:

1. Click the SOA project in the Project Explorer Panel, for example, `SmartMapperReference`. Then double-click the composite file under the Composites folder to launch the Composite Editor on the right, for example, `SmartMapperReference.composite`.
2. From the Project Explorer panel, expand the TIBCO BusinessWorks project and locate the Service Resource.
3. Drag and drop the service agent into the Composite Editor in the Editor panel, for example, `lookupService.serviceagent`.
4. Select the BWSE component, right-click it, and then select **Promote All**.
5. Click the composite service, for example, the `InvokeOperationPT` composite service in the composite editor.
6. Click **Bindings** in the Promoted Service panel, and then click the button to add a binding.
7. Save the project.

Bindings enable communication between TIBCO ActiveMatrix and its environment. Service bindings enable consumers outside the TIBCO ActiveMatrix environment to consume services provided by TIBCO ActiveMatrix.

In the composite sample screen shown in Figure 2, a SOAP binding type has been selected. Refer to *TIBCO ActiveMatrix BusinessWorks Service Engine User's Guide* for details.

Figure 2 Composite Sample Screen



Name: SOAPService_Binding1

Description:

SOAP Version: 1.1

Style: Document

Encoding: Literal

Transport Type: HTTP

Chapter 3 Working with TIBCO ActiveMatrix Administrator

This chapter describes how to add a new node to the development environment, and how to set up the TIBCO ActiveMatrix BusinessWorks Service Engine Plug-in for SmartMapper application to the newly created node using TIBCO ActiveMatrix Administrator.

Topics

Adding a New Node to the Development Environment

When you create TIBCO ActiveMatrix Administrator, a node named `DevNode` is created by default.

If you want to add a new node, complete the following steps:

1. Select **Infrastucture > Nodes**.
2. Click the + New button to add a node.
3. Enter the node name in the New Node dialog, for example, `testNode`, and then click the **Save** button.
4. Select the newly created node.
5. Click the Install button to install the node into the environment.
6. After the node has been installed, click the ▶ Start button to run the node.

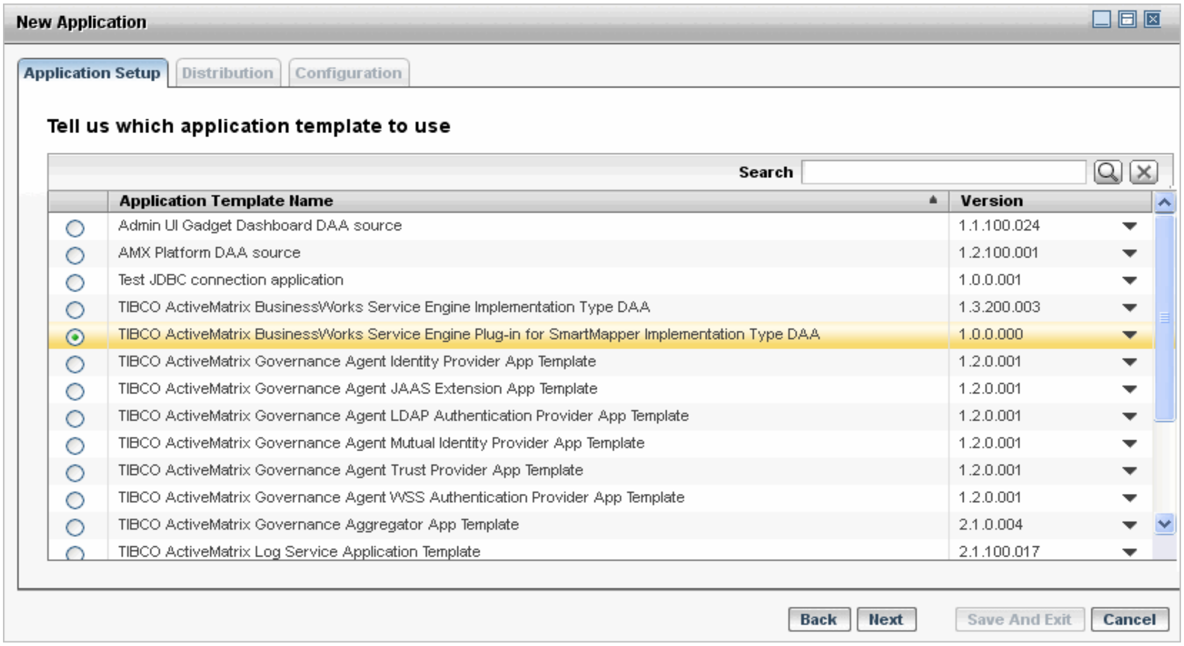
Setting Up the BWSE Plug-in for SmartMapper Application

If you launch TIBCO Configuration Tool after you install the ActiveMatrix platform and TIBCO ActiveMatrix BusinessWorks Service Engine Plug-in for SmartMapper, by default, the TIBCO ActiveMatrix BusinessWorks Service Engine Plug-in for SmartMapper application is automatically installed on the `DevNode`.

However, if you use the newly created node, before deploying SmartMapper applications on this node, you must first complete the following steps to create the TIBCO ActiveMatrix BusinessWorks Service Engine Plug-in for SmartMapper application:

1. Click the **Application** tab in the TIBCO ActiveMatrix Administrator.
2. Click the ✚ New button to add an application.
3. Configure the application in the New Application dialog:
   - Enter a name in the Application Name field, for example, `BWSE_SmartMapper_IT`.
   - Select the **An Existing Application Template** radio button.
   - Click the **Next** button.
4. Select the **TIBCO ActiveMatrix BusinessWorks Service Engine Plug-in for SmartMapper Implementation Type DAA** radio button in the application template list, as shown in Figure 3.

Figure 3 Select a Template for the Application



Tell us which application template to use

	Application Template Name	Version
○	Admin UI Gadget Dashboard DAA source	1.1.100.024
○	AMX Platform DAA source	1.2.100.001
○	Test JDBC connection application	1.0.0.001
○	TIBCO ActiveMatrix BusinessWorks Service Engine Implementation Type DAA	1.3.200.003
●	TIBCO ActiveMatrix BusinessWorks Service Engine Plug-in for SmartMapper Implementation Type DAA	1.0.0.000
○	TIBCO ActiveMatrix Governance Agent Identity Provider App Template	1.2.0.001
○	TIBCO ActiveMatrix Governance Agent JAAS Extension App Template	1.2.0.001
○	TIBCO ActiveMatrix Governance Agent LDAP Authentication Provider App Template	1.2.0.001
○	TIBCO ActiveMatrix Governance Agent Mutual Identity Provider App Template	1.2.0.001
○	TIBCO ActiveMatrix Governance Agent Trust Provider App Template	1.2.0.001
○	TIBCO ActiveMatrix Governance Agent WSS Authentication Provider App Template	1.2.0.001
○	TIBCO ActiveMatrix Governance Aggregator App Template	2.1.0.004
○	TIBCO ActiveMatrix Log Service Application Template	2.1.100.017

a. Click the **Next** button.

b. Review the setup information of the newly created application. Click the **Save and Exit** button.

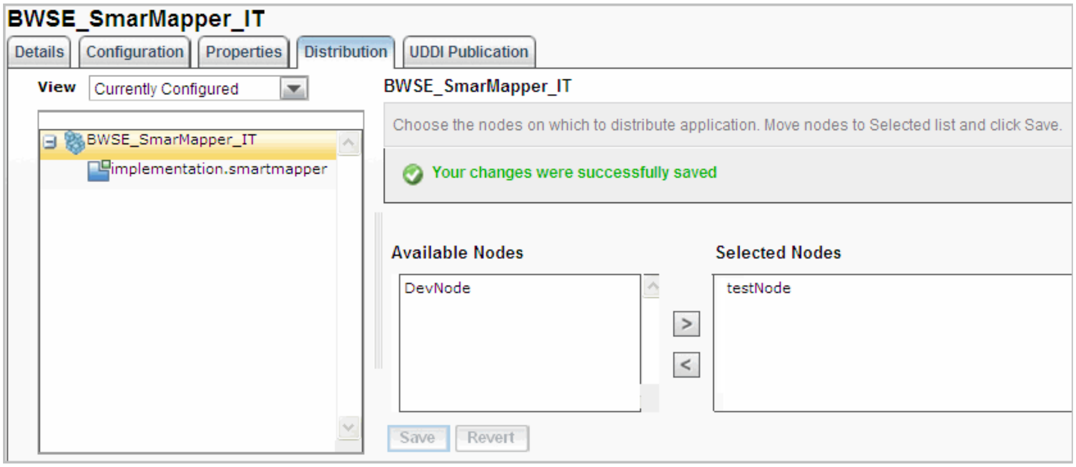
c. Select the newly created application from the applications list. Click the **Distribution** tab of that application.

d. Select the application in the left-hand panel, then select the node on which you want to distribute it. The node named `DevNode`, shown in the Selected Nodes list, is the default nodes.

If you want to use another node, for example, the one named `testNode`, then move it from the Available Nodes list to the Selected Nodes list, and move the default `DevNode` from the Selected Nodes list to the Available Nodes list, as shown in Figure 4.

e. Click the **Save** button.

Figure 4   Selecting Node



View: Currently Configured

BWSE_SmarMapper_IT

Choose the nodes on which to distribute application. Move nodes to Selected list and click Save.

Your changes were successfully saved

Available Nodes: DevNode

Selected Nodes: testNode

f. Click the Deploy button to deploy the application.

   You can select Deploy with Start, Deploy without Start, or More Deploy Options from the Deploy list.

If using the newly created node, you also need to deploy the TIBCO ActiveMatrix BusinessWorks Service Engine application on it.

Chapter 4 Creating a Distributed Application Archive

This chapter describes how to create Distributed Application Archive (DAA).

Topics

Overview

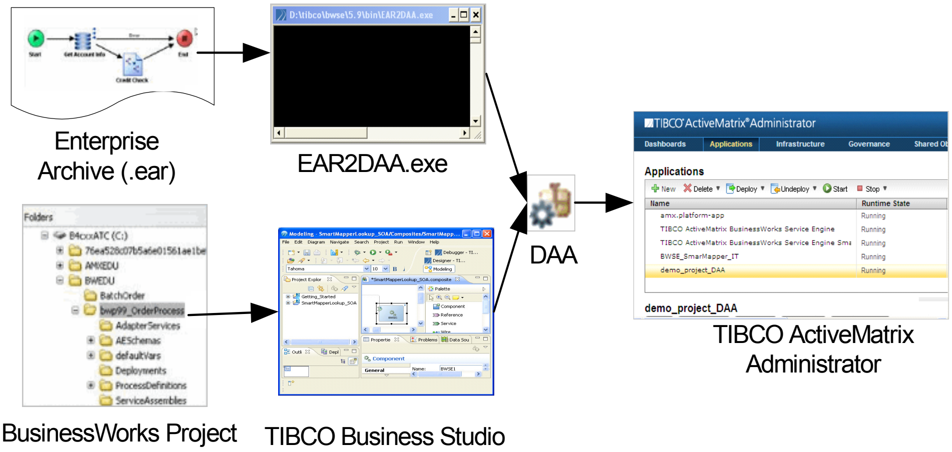
Distributed Application Archive (DAA) is a deployable archive of the composite. It is a file imported into TIBCO ActiveMatrix Administrator to deploy a project to a running node.

Once a component is configured, you can create a DAA for deployment. The DAA file gathers all the information required by the component at runtime.

TIBCO ActiveMatrix BusinessWorks Service Engine provides two ways of creating a DAA:

- Using Business Studio to Create a DAA, page 19
- Using the EAR2DAA Utility to Convert an EAR File to a DAA File, page 20

Figure 5 Two Methods of Creating a DAA



Using Business Studio to Create a DAA

To create a DAA using Business Studio, complete the following steps:

Before creating a DAA, it is a good practice to validate the project using the ☑ button (the Validate Project button) in the Designer perspective.

1. Expand the **Composite** folder in the Project Explorer panel, and select a composite, right-click it and select **Create DAA.**

   Or

   Right-click the composite in the Editor panel, select **Create DAA**.
2. Select the **Deployment Artifacts** folder in the Select Archive Location.
3. Enter a new filename or accept the default name.
4. Click the **Next** button.
5. Select the **Do not use a distribution file** radio button in the Select Distribution page, and then click the **Next** button.
6. Check the **Save DAA Spec** checkbox in the DAA Specification page, and then click the **Finish** button.

Using the EAR2DAA Utility to Convert an EAR File to a DAA File

This section shows how to use the EAR2DAA utility to convert an EAR file to a DAA file.

The EAR2DAA utility allows you to create a DAA archive without creating an ActiveMatrix SOA project. The utility is available at the following location only if the product UI features are installed on the machine:
TIBCO_HOME`\bwse\`*version*`\bin`

Run the utility on the command line with the following syntax:

```
EAR2DAA.exe -earPath EarPath [ -daaLocation DAALocation -componentName
ComponentName -componentVersion ComponentVersion -compositeName
CompositeName -compositeVersion CompositeVersion -verbose true]
```

where *EarPath* is the absolute path of the EAR file.

The EAR2DAA utility accepts the optional arguments specified in the following table.

Table 3 EAR2DAA Optional Arguments

Argument	Default Value
daaLocation	*TIBCO_HOME*`/bwse/`*`version`*`/bin`
componentName	*EarName*
componentVersion	1.0.0
compositeName	*EarName*`_Composite`
compositeVersion	1.0.0
verbose	`false`

Chapter 5 Tutorial

This chapter demonstrates how to configure and deploy a TIBCO BusinessWorks project in the TIBCO ActiveMatrix environment.

Topics

Overview

The example of the TIBCO ActiveMatrix BusinessWorks Service Engine Plug-in for SmartMapper is in the *TIBCO_HOME*`\bwse\plugins\smartmapper\samples` directory. There are two projects:

- `BWSE` is a TIBCO BusinessWorks project. It has two processes, `CreateData` and `Lookup and Remove Data`.

  The example uses JDBC-based storage. Before starting this example, make sure that the database is installed and initialized. Refer to *TIBCO ActiveMatrix BusinessWorks Service Engine Plug-in for SmartMapper Installation*, Chapter 3 Post-Installation for more information about how to add a Third-Party driver.

- `SmartMapperReference` is an SOA project.

Input

The input of this example project are `Seed_Data.xml` and `CRM_Order.xml`,which are located in the *TIBCO_HOME*`\bwse\plugins\smartmapper\samples\BWSE` directory.

Procedure

The detailed procedure about how to deploy the example project is specified in Procedure on page 24.

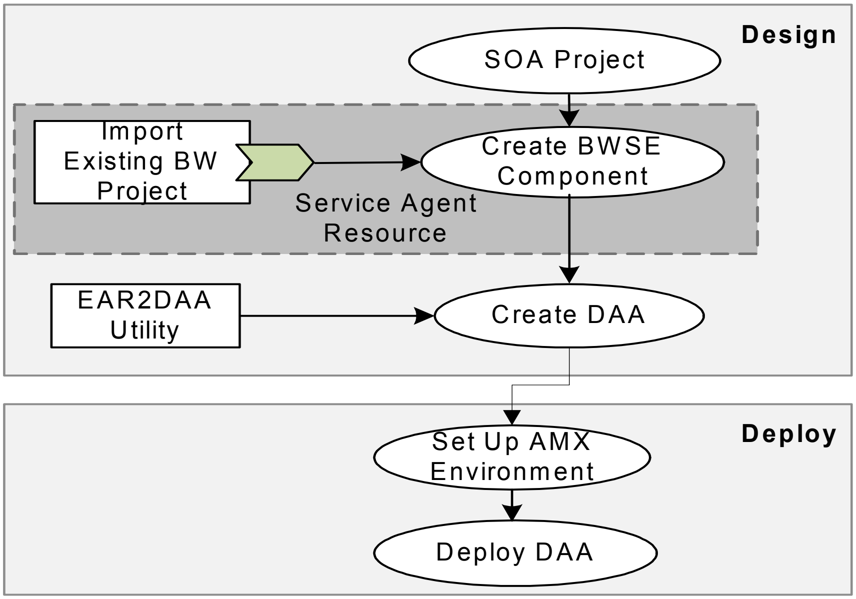
Output

After deploying the example project successfully, you can use the SOAP client, such as, SOAPUI, for testing and running the example. You can also use TIBCO Business Studio to check the output.

Refer to Running the Example on page 33 for more information.

Figure 6 shows the general implementation process.

Figure 6 Project Implementation



Legend

BWSE	TIBCO BusinessWorks Service Engine	AMX	TIBCO ActiveMatrix
BW	TIBCO BusinessWorks	DAA	Distributed Application Archive

Procedure

The following sections provide details about how the details how to develop and deploy a project:

1. Importing the TIBCO Designer Project, page 24
2. Importing the TIBCO SOA Project, page 25
3. Reviewing the Composite Configuration, page 25
4. Reviewing the JDBC-Based Storage Settings, page 27
5. Creating a Distributed Application Archive (DAA), page 28
6. Setting Up the Environment for TIBCO ActiveMatrix, page 29
7. Adding an HTTP Connector Resource Instance, page 29
8. Deploying the DAA, page 31

Importing the TIBCO Designer Project

The example project is a predefined designer project. This section shows how to import the predefined `BWSE` Designer project to TIBCO Business Studio and change the configuration as necessary.

Complete the following steps to import an existing Designer project:

1. Start TIBCO Business Studio.
2. Select **File > New > Project** from the Menu to open the New Project window.
3. Select **TIBCO Designer > TIBCO Designer Project** in the Select A Wizard page, and then click the **Next** button.

Before you create a new TIBCO Designer project, you need to install TIBCO ActiveMatrix BusinessWorks SmartMapper Plug-in or TIBCO ActiveMatrix BusinessWorks SmartMapper Enterprise Server.

4. Select the **Create Project Using Existing Designer Project** radio button, and then click the **Browse** button to specify the project location.

   The directory for the `BWSE` Designer project is *TIBCO_HOME*`\bwse\plugins\smartmapper\samples`.

5. Click the **Finish** button.

   The `BWSE` Designer project appears in the Project Explorer panel.

Importing the TIBCO SOA Project

Complete the following steps to import the existing SmartMapperReference SOA project:

1. Select **File > Import** from the Menu in TIBCO Business Studio.
2. Select **General > Existing Projects Into Workspace**, and then click the **Next** button in the Import dialog.
3. Select the root directory of the SOA project.

   The directory for the SmartMapperReference SOA project is *TIBCO_HOME*\bwse\plugins\smartmapper\samples.
4. Check the **Copy Projects Into Workspace** checkbox, and then click the **Finish** button.

   The SmartMapperReference SOA project appears in the Project Explorer panel.

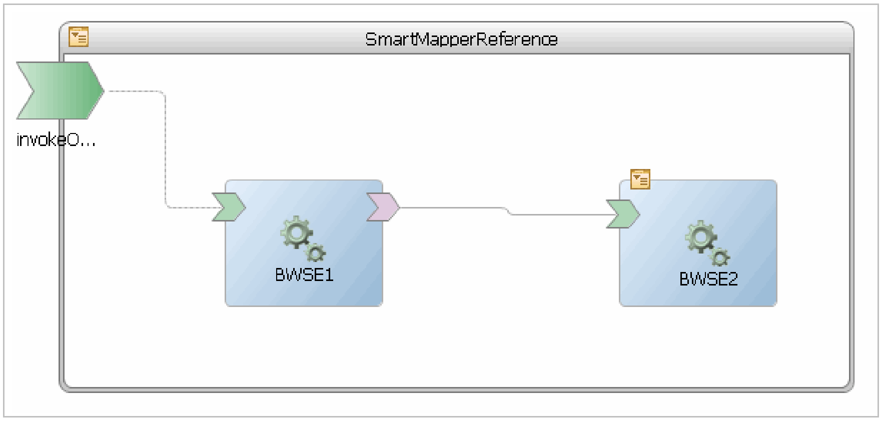
To create a new TIBCO SOA project, refer to Creating a TIBCO SOA Project on page 8 for details.

Reviewing the Composite Configuration

Complete the following steps to review the composite configuration:

1. Expand the **Composites** folder of the SmartMapperReference project in the Project Explorer panel.
2. Double-click the predefined SmartMapperReference.composite composite to launch the Composite Editor.

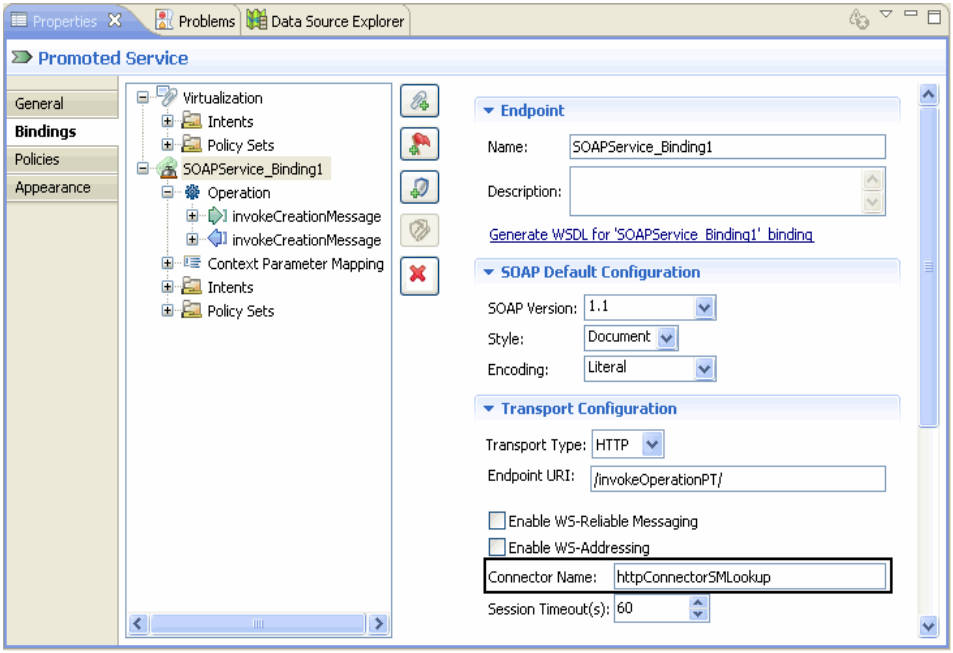
Figure 7 Composite



3. Select the component on the canvas and view its configuration from the Properties tab.

4. Select the promoted service named `invokeOperationTP` and view the binding configuration from the Properties tab.

Figure 8 Promoted Service Binding



Name: SOAPService_Binding1

Description:

SOAP Version: 1.1

Style: Document

Encoding: Literal

Transport Type: HTTP

Endpoint URI: /invokeOperationPT/

□ Enable WS-Reliable Messaging

□ Enable WS-Addressing

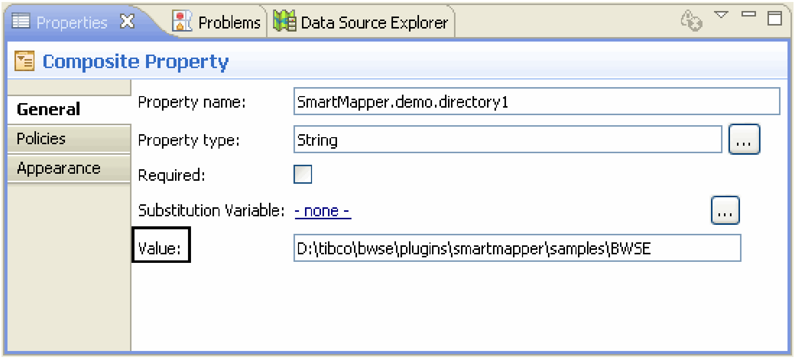
Connector Name: httpConnectorSMLookup

Session Timeout(s): 60

5. Select the component or composite property to view the property configuration from the Properties view.

Make sure that the path in the Value field points to the `BWSE` BusinessWorks project.

Figure 9 Composite Property



Composite Property

Property name: SmartMapper.demo.directory1

Property type: String

Required: □

Substitution Variable: - none -

Value: D:\tibco\bwse\plugins\smartmapper\samples\BWSE

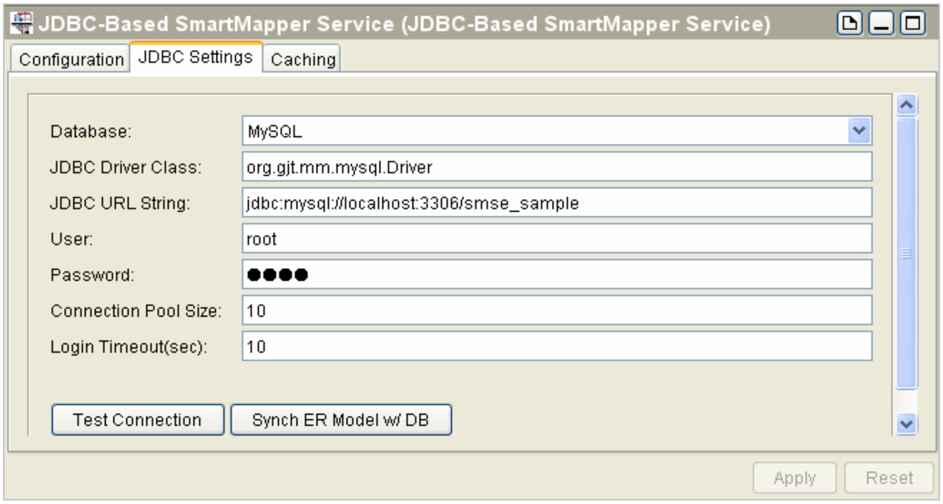
6. Save the configuration if any changes have been made.

Reviewing the JDBC-Based Storage Settings

Before creating the DAA file, make sure that the information of JDBC-Based SmartMapper Service in the TIBCO BusinessWorks project (`BWSE` project) is correct.

Figure 10 shows a sample screen of the JDBC Settings panel.

Figure 10 JDBC Settings



JDBC-Based SmartMapper Service (JDBC-Based SmartMapper Service)

Database: MySQL

JDBC Driver Class: org.gjt.mm.mysql.Driver

JDBC URL String: jdbc:mysql://localhost:3306/smse_sample

User: root

Password: ●●●●

Connection Pool Size: 10

Login Timeout(sec): 10

The following steps show how to review and update the JDBC settings:

1. Double-click the **SmartMapper ER Model.smartmappermodel** file in the Project Explorer panel.

2. Double-click the **Storage** folder.
3. Select **JDBC-Based SmartMapper Service**, then click the **JDBC Settings** tab.
4. Configure the information in the JDBC Settings tab, such as, JDBC URL String, username, and password.
5. Click the **Synch ER Model W/ DB** button to synchronize the Entity Relationship (ER) Model.

   Refer to *TIBCO ActiveMatrix BusinessWorks SmartMapper User's Guide* (Chapter 7 SmartMapper ER Model Resources) for details about JDBC-Based SmartMapper Service.
6. Test the connection:

   a. Add the MYSQL JAR into the environment.

   — Copy and paste the JAR you want to use in the *TIBCO_HOME*`\bw\plugin\lib` directory.

   — Close Business Studio.

   — Run the **generateManifest.exe** generate manifest tool. It is available under *TIBCO_HOME*`\dabs\`*version_number*`\tools\generateManifest`.

   — Start TIBCO Business Studio. Open the command window, change to the *TIBCO_HOME*`\studio\`*version_number*`\eclipse` directory, then type **`TIBCOBusinessStudio - clean`**.

   — Click the **Test Connection** button.

Creating a Distributed Application Archive (DAA)

Projects developed during design time need to be packaged into a DAA file before you can deploy them in the TIBCO ActiveMatrix environment.

To create a DAA using Business Studio, complete the following steps:

1. Expand the **Composite** folder in the Project Explorer panel, right-click the composite, and select **Create DAA.**
2. Select the **Deployment Artifacts** folder in the Select Archive Location page.

   Enter a new filename or accept the default name in the File Name field. Then click the **Next** button.
3. Select the **Do Not Use A Distribution File** radio button in the Select Distribution page, then click the **Next** button.
4. Check the **Save DAA Spec** checkbox in the DAA Specification page, and then click the **Finish** button.

The `SmartMapperReference` DAA file is created successfully and stored in the SmartMapperReference\Deployment Artifacts folder in your workspace.

Setting Up the Environment for TIBCO ActiveMatrix

Complete the following steps to set up the environment for TIBCO ActiveMatrix:

1. Start TIBCO Enterprise Message Service (EMS) Server.

   Right-click the **My Computer** icon on your desktop, select **Mange > Sevices and Applications > Services**, and then select and start the **TIBCO EMS Server** service.

2. Start the predefined TIBCO host.

   Run the executable named `tibcohost.exe` located in the configuration directory of TIBCO ActiveMatrix, for example: *TIBCO_configuration_Folder*`\tibcohost\`*TibcohostInstance_name*`\host\bin`.

If you shut down the EMS server and the TIBCO host (for example, if you reboot your computer) before using TIBCO ActiveMatrix Administrator, you need to restart TIBCO EMS Server and the TIBCO host.

3. Start TIBCO ActiveMatrix Administrator.

   If TIBCO ActiveMatrix Administrator has been created:

   a. Launch a web browser and enter the Administrator URL, for example, `http://`*host_name*`:8120/amxadministrator/loginForm.jsp`, in the address bar to start the TIBCO ActiveMatrix Administrator Server.

   b. Enter the username and password to log in. The default username is `root`, and the password is `t`.

   If TIBCO ActiveMatrix Administrator has not been created, see *TIBCO ActiveMatrix BusinessWorks Service Engine Plug-in for SmartMapper Installation* for instructions about how to create it.

Adding an HTTP Connector Resource Instance

The following steps describe how to add an HTTP Connector resource instance using TIBCO ActiveMatrix Administrator:

1. Open TIBCO ActiveMatrix Administrator.
2. Select **Shared Objects > Resource Templates**.
3. Click the **New** button in the Resource Templates page. The Add Resource Template dialog appears.

4. In the Add Resource Template dialog:

   — Enter **`httpConnectorSMLookup`** in the Name field.

   — Select **HTTP Connector** from the Type drop-down list.

The name of the HTTP Connector resource must be the same as the binding connector name of the `invokeOperationTP` promoted service, as shown in Figure 8.

Also make sure that the values in the Machine Name and the Port fields are consitent with the Host and the Port fields when creating the concrete WSDL to run the example project, see Running the Example on page 33.

   — Click the **Save** button.

5. Select the newly created HTTP Connector resource, then click the **New Resource Instances** button in the Resource Templates page.

   The New Resource Instances dialog appears.

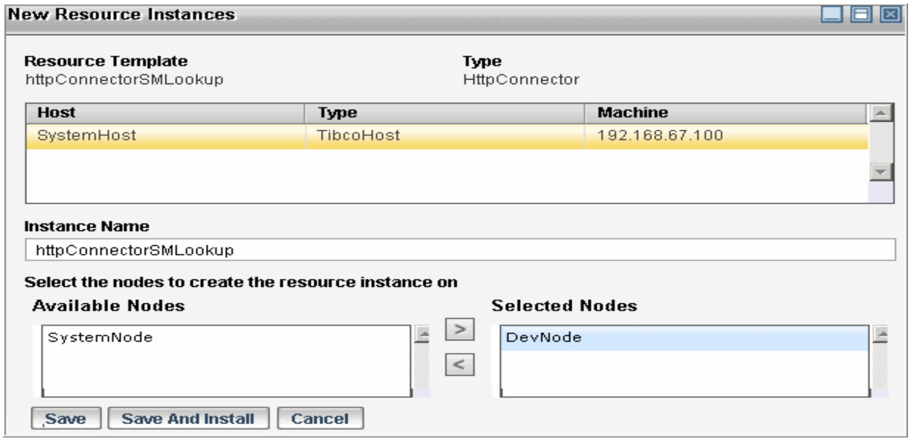
6. In the New Resource Instances dialog:

   — Select the **SystemHost** host (the default host).

   — Select the **DevNode** node in the Available Nodes column, then click the > button to move the DevNode node to the Selected Nodes column.

   — Click the **Save And Install** button.

Figure 11 New Resource Instances



New Resource Instances

Resource Template
httpConnectorSMLookup

Type
HttpConnector

Host	Type	Machine
SystemHost	TibcoHost	192.168.67.100

Instance Name
httpConnectorSMLookup

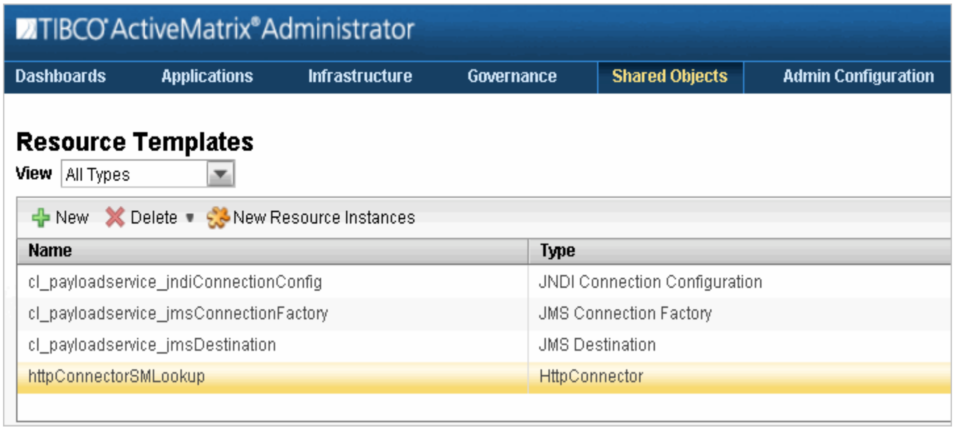
Select the nodes to create the resource instance on

Available Nodes
SystemNode

Selected Nodes
DevNode

The HTTP Connector resource instance is created successfully, as shown in Figure 12.

Figure 12 HTTP Connector Resource Instance



Resource Templates

View: All Types

Name	Type
cl_payloadservice_jndiConnectionConfig	JNDI Connection Configuration
cl_payloadservice_jmsConnectionFactory	JMS Connection Factory
cl_payloadservice_jmsDestination	JMS Destination
httpConnectorSMLookup	HttpConnector

Deploying the DAA

The following steps describe how to deploy the DAA using TIBCO ActiveMatrix Administrator:

1. Open TIBCO ActiveMatrix Administrator.
2. Select the **Applications** tab and click the **New** button.
3. Configure the application in the New Application dialog.
   a. Enter a name in the Application Name field, for example, `SmartMapperReference`.
   b. Select the **Upload A DAA File** radio button.
   c. Click the **Next** button.
4. Click the **Browse** button and select the DAA file created previously in Creating a Distributed Application Archive (DAA) on page 28.
5. Click the **Next** button.
6. Review the application setup information and click the **Done With Application Setup** button.

You can click the **Save and Exit** button instead, and then configure and deploy the application in a different way, as shown in step 4 of Setting Up the BWSE Plug-in for SmartMapper Application on page 13.

Refer to the TIBCO ActiveMatrix Administration documentation for more details. You can access the online help by clicking the **Help** button on the top-right corner of the Administrator page.

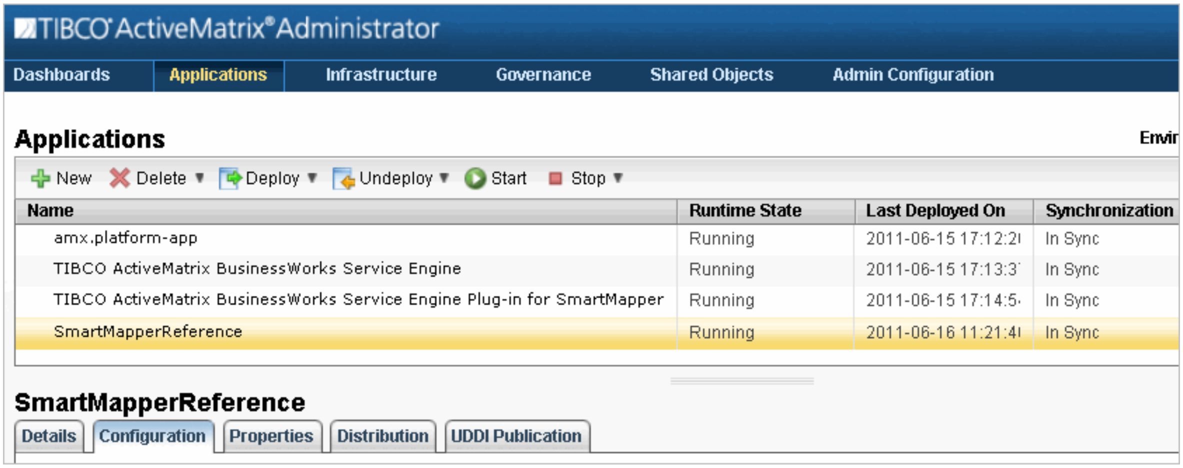
7. Accept the defaults and click the **Next** button.
8. Select the host on which you want to distribute the application.
9. Click the **Next** button until you see the **Customize Your Node Selection** page.
10. Select the **DevNode** node from the drop-down list in the Node Name column, then click the **Save And Exit** button.
11. Add the Third-Party driver to the `DevNode` node.

This exmple uses the MySQL database. Refer to *TIBCO ActiveMatrix BusinessWorks Service Engine Plug-in for SmartMapper Installation* for more information about how to add a Third-Party driver to the node manually.

12. Click the **Deploy** button.

    The application is deployed successfully and appears in the Applications table's Runtime State column as `Running`, as shown below in Figure 13.

Figure 13 Deploy and Run the Service



Name	Runtime State	Last Deployed On	Synchronization
amx.platform-app	Running	2011-06-15 17:12:2	In Sync
TIBCO ActiveMatrix BusinessWorks Service Engine	Running	2011-06-15 17:13:3	In Sync
TIBCO ActiveMatrix BusinessWorks Service Engine Plug-in for SmartMapper	Running	2011-06-15 17:14:5	In Sync
SmartMapperReference	Running	2011-06-16 11:21:4	In Sync

Running the Example

The following steps show how to run the example using TIBCO Business Studio after the example is deployed successfully:

1. Generate WSDL

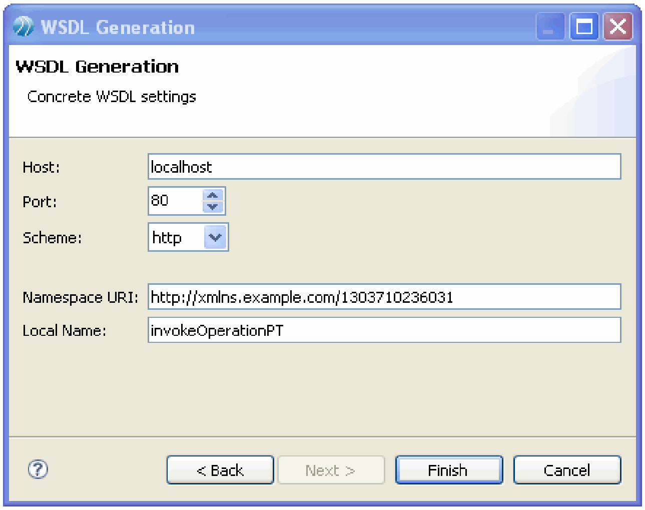
   a. Select **InvokeOperationPT** in the SmartMapperReference.composite Editor panel, as shown in Figure 7.

   b. Click the **Bindings** tab, then select the **SOAPService_Binding1** binding, as shown in Figure 8.

   c. Click the **Generate WSDL For 'SOAPService_Binding1' Binding** link. The WSDL Generation dialog appears.

   d. Accept the defaults, then click the **Next** button.

   e. Provide the host and port in the Host field and Port field, then click the **Finish** button.



WSDL Generation

WSDL Generation

Concrete WSDL settings

Host: localhost

Port: 80

Scheme: http

Namespace URI: http://xmlns.example.com/1303710236031

Local Name: invokeOperationPT

The `invokeCreation_gen.wsdl` is created successfully.

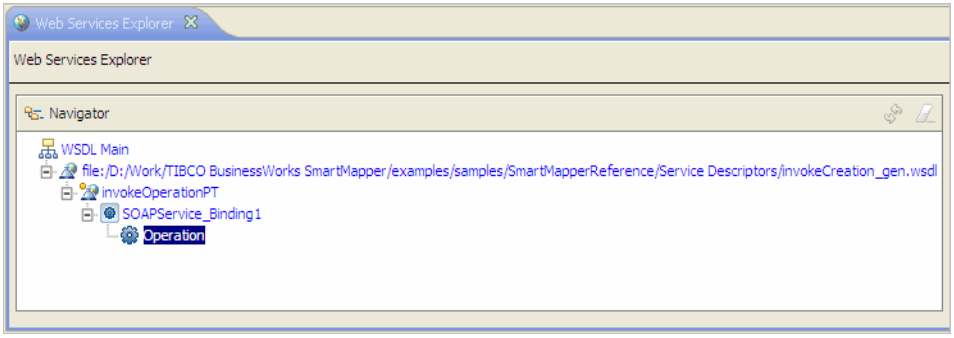
Make sure that the host and the port are consistent with the values provided in the Machine Name and the Port fields when creating the HTTP Connector Resource. See Adding an HTTP Connector Resource Instance on page 29 for more information.

2. Right-click **InvokeCreation_gen.wsdl** under the SmartMapperReference project in the Project Explorer, then select **Web Services > Test With Web Services Explorer**.

The Web Services Explorer appears in the Editor Panel.

3. Select **SOAPService_Binding1 > Operation** in the Navigator panel, as shown in Figure 14.

Figure 14 Web Service Navigator



Web Services Explorer

Web Services Explorer

Navigator

WSDL Main

file:/D:/Work/TIBCO BusinessWorks SmartMapper/examples/samples/SmartMapperReference/Service Descriptors/invokeCreation_gen.wsdl

invokeOperationPT

SOAPService_Binding1

Operation

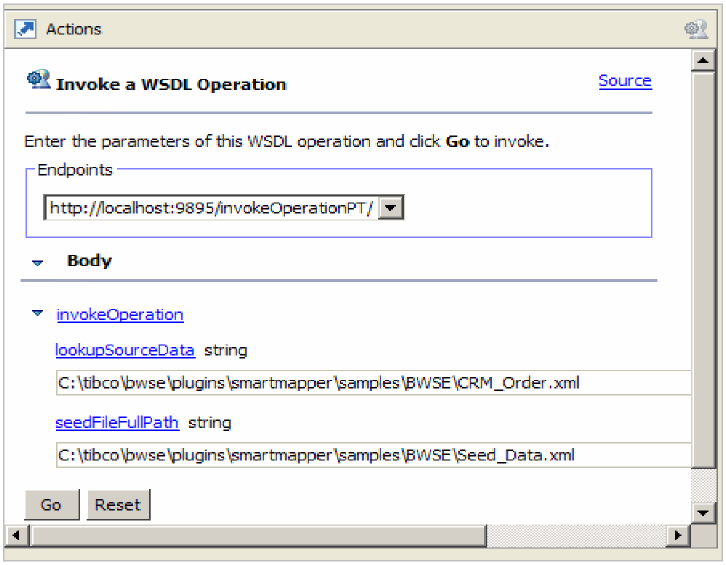
4. Enter the input data in the LookupSourceData and SeedFileFullPath field, as shown in Figure 15.

— LookupSourceData: `CRM_Order.xml`

— SeedFileFullPath: `Seed_Data.xml`

You can find these two files in *TIBCO_HOME*`\bwse\plugins\smartmapper\samples\BWSE` directory.

Figure 15 Invoke a WSDL Operation



Actions

Invoke a WSDL Operation Source

Enter the parameters of this WSDL operation and click **Go** to invoke.

Endpoints

http://localhost:9895/invokeOperationPT/

Body

invokeOperation

lookupSourceData string

C:\tibco\bwse\plugins\smartmapper\samples\BWSE\CRM_Order.xml

seedFileFullPath string

C:\tibco\bwse\plugins\smartmapper\samples\BWSE\Seed_Data.xml

Go Reset

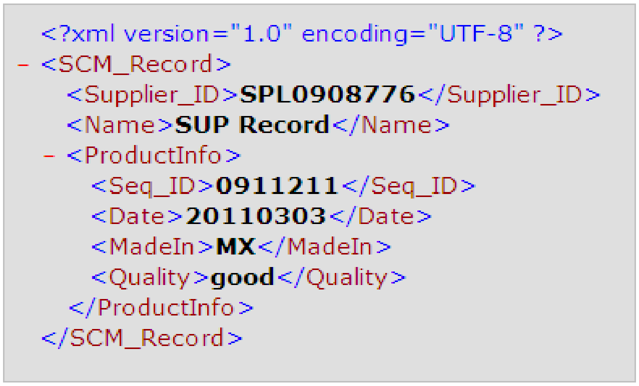
5. Click the **Go** button.

After running the example successfully, an output file named `SCM_Record.xml` is created.

This output file is located in the direcory that you defined in the Value field of the Composite Property's panel, as shown in Figure 9.

Figure 16 shows the content of this output file.

Figure 16 SCR_Record XML



```
  <?xml version="1.0" encoding="UTF-8" ?>
- <SCM_Record>
    <Supplier_ID>SPL0908776</Supplier_ID>
    <Name>SUP Record</Name>
  - <ProductInfo>
      <Seq_ID>0911211</Seq_ID>
      <Date>20110303</Date>
      <MadeIn>MX</MadeIn>
      <Quality>good</Quality>
    </ProductInfo>
  </SCM_Record>
```

Index