

TIBCO ActiveMatrix® BPM SOA Development Tutorials

Software Release 4.2

August 2017

Important Information

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

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

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

TIBCO and Two-Second Advantage are either registered trademarks or trademarks of TIBCO Software Inc. in the United States and/or other countries.

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

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

THIS SOFTWARE MAY BE AVAILABLE ON MULTIPLE OPERATING SYSTEMS. HOWEVER, NOT ALL OPERATING SYSTEM PLATFORMS FOR A SPECIFIC SOFTWARE VERSION ARE RELEASED AT THE SAME TIME. SEE THE README FILE FOR THE AVAILABILITY OF THIS SOFTWARE VERSION ON A SPECIFIC OPERATING SYSTEM PLATFORM.

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

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

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

Copyright © 2010-2017 TIBCO Software Inc. All rights reserved.

TIBCO Software Inc. Confidential Information

Contents

- TIBCO Documentation and Support Services4**
- Basic SOA Design Tutorials6**
- Java Tutorials7**
- Mediation Tutorials10**
- WebApp Tutorials11**
- Policy Tutorials12**
 - Configuring the ResourceInstance Attribute in an External Policy 12
 - Preparing an Application that Injects Resource Instances into Policies 12

TIBCO Documentation and Support Services

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

<https://docs.tibco.com>

Product-Specific Documentation

Documentation for TIBCO products is not bundled with the software. Instead, it is available on the TIBCO Documentation site.

The following documents form the documentation set:

- *Concepts*: Read this manual before reading any other manual in the documentation set. This manual describes terminology and concepts of the platform. The other manuals in the documentation set assume you are familiar with the information in this manual.
- *Development Tutorials*: Read this manual for a step-by-step introduction to the process of creating, packaging, and running composites in TIBCO Business Studio.
- *Composite Development*: Read this manual to learn how to develop and package composites.
- *Java Component Development*: Read this manual to learn how to configure and implement Java components.
- *Mediation Component Development* : Read this manual to learn how to configure and implement Mediation components.
- *Mediation API Reference* : Read this manual to learn how to develop custom Mediation tasks.
- *Spring Component Development* : Read this manual to learn how to configure and implement Spring components.
- *WebApp Component Development* : Read this manual to learn how to configure and implement Web Application components.
- *Administration Tutorial*: Read this manual for a step-by-step introduction to the process of creating and starting the runtime version of the product, starting TIBCO ActiveMatrix servers, and deploying applications to the runtime.
- *Administration*: Read this manual to learn how to manage the runtime and deploy and manage applications.
- *Hawk ActiveMatrix Plug-in User's Guide*: Read this manual to learn about the Hawk plug-in and its optional configurations.
- *Installation and Configuration*: Read this manual to learn how to install and configure the software.
- *Release Notes*: Read this manual for a list of new and changed features, steps for migrating from a previous release, and lists of known issues and closed issues for the release.

The documentation for the following features is installed separately:

- TIBCO ActiveMatrix Implementation Type for C++
- TIBCO ActiveMatrix Binding Type for EJB
- TIBCO ActiveMatrix Binding Type for Adapters
- TIBCO ActiveMatrix Implementation Type for TIBCO Adapters
- TIBCO ActiveMatrix Implementation Type for Microsoft CLR
- TIBCO ActiveMatrix Binding Type for REST

How to Contact TIBCO Support

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

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

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

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

<https://support.tibco.com>

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

How to Join TIBCO Community

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

<https://community.tibco.com>

Basic SOA Design Tutorials

Many TIBCO ActiveMatrix SOA development tutorials are distributed as cheat sheets in TIBCO Business Studio. To access the cheat sheets, start TIBCO Business Studio, select **Help > Cheat Sheets...**, and expand the **TIBCO SOA Development** node. For tutorials that are not cheat sheets, see the specific tutorial category.



Many of the steps in cheat sheets launch modal dialogs. Once a dialog is open, you cannot expand the view containing cheat sheet content. To ensure that you can see all the content, before opening a dialog you can expand the view so that you could see all of the cheat sheet content or you can detach the view so that it can be resized even while a dialog is open.

To access the basic SOA design tutorials, expand the **Basic** folder.

- **Create an SOA Project with an Empty Composite**

Describes how to create an SOA project.

- **Import Projects from an Archive**

Describes how to import projects from an archive file.

- **Import a WSDL File into an SOA Project**

Describes how to import a WSDL file into an SOA project.

Java Tutorials

Several Java tutorials that are included with TIBCO ActiveMatrix explain how to use and modify different features.



Many of the steps in cheat sheets launch modal dialogs. Once a dialog is open, you cannot expand the view containing cheat sheet content. To ensure that you can see all the content, before opening a dialog you can expand the view so that you could see all of the cheat sheet content or you can detach the view so that it can be resized even while a dialog is open.

To access the Java tutorials, expand the **Java** folder.

- **How to Create and Package a Hello World Application**

The Hello World Application exposes one service that returns an acknowledgment string when a user submits a name to a Hello World SOAP client. The Hello World SOAP client sends requests to Hello World SOAP service, which forwards them to the Hello World Java component. When the Hello World component receives the request, it constructs the response string and returns it to Hello World SOAP service, which then forwards the response to the Hello World SOAP client.



- **How to Run the Hello World Application**

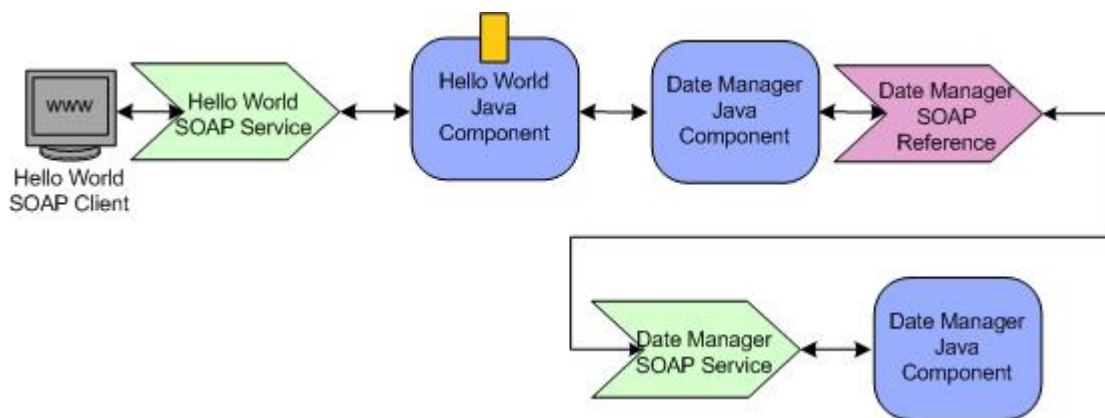
Describes how to use rapid application development features to run the Hello World application on a local node and the Web Services Explorer to generate SOAP requests to invoke the service exposed by the application.

- **How to Enhance the Hello World Application**

Describes how to enhance the Hello World application as follows:

- Add an implementation property to configure the Hello World component
- Add a partner Date Manager component
- Add a composite reference and HTTP client shared resource to access an external Date Manager service

The enhanced application consists of two components—Hello World and Date Manager—that work together to return a greeting when a user submits a name to a Hello World SOAP client. The SOAP client sends requests to the Hello World service SOAP binding, which forwards them to a Hello World component. When a Hello World component receives the request, it sends a request for the current time to Date Manager component. The Date Manager component in turn invokes a Date Manager reference, which uses an HTTP client shared resource to invoke the Date Manager application. The Date Manager application returns a string containing the date and time to the Date Manager component via the reference. A property contains the salutation portion of the response returned by the Hello World component.



- **How to Run the Enhanced Hello World Application**

Describes how to use rapid application development features to run the Hello World and Date Manager applications on a local node and the Web Services Explorer to generate SOAP requests to invoke the service exposed by the Hello World application.

- **How to Deploy the Enhanced Hello World Application**

Describes how to use Administrator CLI features to deploy the Hello World and Date Manager applications on a remote node.

- **How to Create and Package the Hello World Fault Application**

Describes how to create a version of the Hello World application that returns a SOAP fault message if the user does not submit a name.

- **How to Run the Hello World Fault Application**

Describes how to use rapid application development features to run the Hello World Fault application on a local node and the Web Services Explorer to invoke the Hello World service to return the SOAP fault.

- **How to Create and Package Applications that Use a Shared Data Binding Library**

Describes how to create and package a shared data binding library and how to configure and package two applications to reference the library.

- **How to Deploy Applications that Use a Shared Library**

Describes how to use Administrator CLI features to deploy a DAA containing a shared library and the applications that reference the library.

- **How to Create and Package a Phonebook Application**

The Phonebook application exposes a Phonebook service with two operations: addPhone and getPhone. The application consists of one Java component that uses a JDBC resource to access the database that stores the phonebook data. The JDBC resource references a database driver packaged in a separately deployed custom feature.

- **How to Run the Phonebook Application**

Describes how to use rapid application development features to run the Phonebook application on a local node and the Web Services Explorer to generate SOAP requests to invoke the operations in the Phonebook service.

- **How to Run the Hello World JMS Application**

(tutorial only) Describes how to use rapid application development features to run the Hello World JMS application on a local node and a JMS test client to invoke the Hello World service.

Sample Project Location

Cheat Sheet	Sample Project Archives
Hello World <ul style="list-style-type: none"> • Create and Package • Run 	<i>TIBCO_HOME/amx/version/samples/java/helloworld1.zip</i>
Enhanced Hello World <ul style="list-style-type: none"> • Create and Package • Run • Deploy 	<i>TIBCO_HOME/amx/version/samples/java/helloworld2.zip</i>
Hello World Fault <ul style="list-style-type: none"> • Create and Package • Run 	<i>TIBCO_HOME/amx/version/samples/java/helloworld3.zip</i>
Hello World Shared <ul style="list-style-type: none"> • Create and Package • Deploy 	<i>TIBCO_HOME/amx/version/samples/java/helloworld4.zip</i>
Phonebook <ul style="list-style-type: none"> • Create and Package • Run 	<i>TIBCO_HOME/amx/version/samples/java/phonebook.zip</i>
Hello World JMS	<i>TIBCO_HOME/amx/version/samples/java/helloworldjms.zip</i> The .zip file also contains a Readme file.

Mediation Tutorials

The mediation tutorials explain how to create, package, and modify mediation flows.



Many of the steps in cheat sheets launch modal dialogs. Once a dialog is open, you cannot expand the view containing cheat sheet content. To ensure that you can see all the content, before opening a dialog you can expand the view so that you could see all of the cheat sheet content or you can detach the view so that it can be resized even while a dialog is open.

To access the mediation tutorials, expand the **Mediation** folder.

- **How to Create and Package a Hello World Mediation Flow**

Develop a simple Hello World application that returns an acknowledgment when a user submits a name to a SOAP client.

- **How to Create and Package a Data Enrichment Mediation Flow**

Use the Invoke Operation mediation task to enrich data in a mediation flow. The Invoke Operation task enables you to enhance message input, so that the message gathers additional data as it follows the mediation flow to the target operation.

- **How to Create and Package a Content-Based Routing and Data Transformation Mediation Flow**

Create a mediation flow that uses route tasks to send requests to service implementations in a sample Global Distribution System (GDS) travel service.

Sample Project Location

Cheat Sheet	Sample Project Archives
Hello World Mediation • Create and Package	<i>TIBCO_HOME/amx_it_mediation/version/samples/HelloWorld</i>
Content-Based Routing and Data Transformation • Create and Package	<i>TIBCO_HOME/amx_it_mediation/version/samples/ContentBasedRouting</i> <i>TIBCO_HOME/amx_it_mediation/version/samples/TargetService</i>
Data Enrichment Mediation Flow • Create and Package	<i>TIBCO_HOME/amx_it_mediation/version/samples/DataEnrichment</i>

WebApp Tutorials

WebApp tutorials explain how to create and run web applications.



Many of the steps in cheat sheets launch modal dialogs. Once a dialog is open, you cannot expand the view containing cheat sheet content. To ensure that you can see all the content, before opening a dialog you can expand the view so that you could see all of the cheat sheet content or you can detach the view so that it can be resized even while a dialog is open.

To access the web application tutorials, expand the **WebApp** folder.

- **How to Create a Hello World Web Application**

The Hello World web application exposes one service—Hello World service—that returns an acknowledgment string when a user submits a name to the web application URL. In this cheat sheet you package an existing web application.

- **How to Run the Hello World Web Application**

Describes how to use rapid application development features to run the Hello World web application on a local node and invoke the web application from a web browser.

- **How to Create a Hello World2 Web Application**

- The Hello World web application exposes one service—Hello World service—that returns an acknowledgment string when a user submits a name to the web application URL. In this cheat sheet you create a new web application.

Sample Project Location

Cheat Sheet	Sample Project Archives
Hello World Web Application <ul style="list-style-type: none"> • Create and Package • Run 	<code>TIBCO_HOME/amx/version/samples/webapp/helloworld.zip</code>
Hello World2 Web Application <ul style="list-style-type: none"> • Create and Package 	<code>TIBCO_HOME/amx/version/samples/webapp/helloworld2.zip</code>

Policy Tutorials

Policy tutorials illustrate how to use TIBCO ActiveMatrix policies.

Several external policies require you to configure an attribute named `resourceInstance`. The value of that attribute is not an actual resource instance, but an application that references the required resource instance; the purpose of the application is to inject the resource instance into policies that require it.

When configuring these policies, you must first ensure that the appropriate application is deployed and running on the nodes where you will deploy the policy. To prepare the resource instance and application, see [Preparing an Application that Injects Resource Instances into Policies](#) on page 12.

To configure the policy to use the application, see [Configuring the ResourceInstance Attribute in an External Policy](#) on page 12.

The location of the sample policies is `TIBCO_HOME/amx/version/samples/policy/samples.zip`.

Configuring the ResourceInstance Attribute in an External Policy

Procedure

- Several external policies require you to configure an attribute named `resourceInstance`. When configuring such policies (by copying from the samples), replace the sample value of the `resourceInstance` attribute with the actual name of the application that references (and injects) the appropriate resource instance.

Preparing an Application that Injects Resource Instances into Policies

Procedure

- Create resource templates of the required types:

Policy	Required Resource Templates
Signing or Decryption	Identity Provider
	Keystore Provider
Signature Verification or Encryption	Trust Provider
	Keystore Provider
Both of the above	Mutual Identity Provider
	Keystore Provider
WssProvider or WssConsumer	WS-Security ASP
LDAP for authentication	LDAP Authentication Provider

- Create a resource instance from each template.

3. Create governance agent applications that reference the resource instances (one application per resource instance):

Resource Instance Type	Application Template	Application Template DAA Name
LDAP Authentication Provider	TIBCO ActiveMatrix Governance Agent LDAP Authentication Provider App Template	<code>com.tibco.governance.sharedresource.ldap.asp.daa</code>
Mutual Identity Provider	TIBCO ActiveMatrix Governance Agent Mutual Identity Provider App Template	<code>com.tibco.governance.sharedresource.mutual.isp.daa</code>
Identity Provider	TIBCO ActiveMatrix Governance Agent Identity Provider App Template	<code>com.tibco.governance.sharedresource.subject.isp.daa</code>
Trust Provider	TIBCO ActiveMatrix Governance Agent Trust Provider App Template	<code>com.tibco.governance.sharedresource.trust.isp.daa</code>
WS-Security ASP	TIBCO ActiveMatrix Governance Agent WSS Authentication Provider App Template	<code>com.tibco.governance.sharedresource.wss.asp.daa</code>

4. Distribute the applications to the appropriate nodes and run them. Each application that injects a resource instance must run on every node where a policy requires that resource instance.
5. Each application type has a single property that you must set. In the Properties tab of the application, specify the resource instance name as the value of that property. For example, the Trust Provider application has a property named `TrustIsConnectionFactory`. Set its value to the name of the resource instance that this application injects into policies.