

TIBCO ActiveMatrix BusinessWorks™ ActiveAspects Plug-in

Getting Started

*Software Release 1.2
October 2011*

TIBCO provides the two-second advantage™



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 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, The Power of Now, TIBCO ActiveMatrix BusinessWorks, TIBCO Rendezvous, TIBCO Administrator, TIBCO Enterprise Message Service, TIBCO InConcert, TIBCO Policy Manager, and TIBCO Hawk are either registered trademarks or trademarks of TIBCO Software Inc. in the United States and/or other countries.

EJB, Java EE, J2EE, and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. 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 -2011 TIBCO Software Inc. ALL RIGHTS RESERVED.

TIBCO Software Inc. Confidential Information

Contents

Preface	v
Related Documentation	vi
TIBCO ActiveMatrix BusinessWorks ActiveAspects Plug-in Documentation	vi
Other TIBCO Product Documentation	vi
Typographical Conventions	viii
Connecting with TIBCO Resources	x
How to Join TIBCOCommunity	x
How to Access All TIBCO Documentation	x
How to Contact TIBCO Support	x
Chapter 1 About TIBCO ActiveMatrix BusinessWorks ActiveAspects Plug-in	1
Overview	2
Chapter 2 Tutorial	3
Starting TIBCO Designer and Saving Your Project	4
Creating the JMS Queue Sender Process	5
Testing the TIBCO Designer Project	7
Setting Development Environment in Eclipse	8
Product Libraries	8
Advice Development in Eclipse	9
Setting up Eclipse for TIBCO ActiveMatrix BusinessWorks ActiveAspects Plug-in Development	9
Creating a Synchronous Advice Implementation	10
Exporting the Advice Implementation JAR file using Eclipse	11
Creating a New Aspect File	11
Creating an Aspect JAR	12
Using TIBCO Designer with TIBCO ActiveMatrix BusinessWorks ActiveAspects Plug-in	12
Expected Final Output	13
Index	15

Preface

TIBCO ActiveMatrix BusinessWorks ActiveAspects Plug-in (BWAA) extends the capabilities of TIBCO ActiveMatrix BusinessWorks by adding an Aspect Oriented Programming capability. This allows you to modularize and inject crosscutting concerns to enhance your ActiveMatrix BusinessWorks processes at deploy time while keeping the original ActiveMatrix BusinessWorks processes intact. It exposes a JAVA API to build jar files that can alter the execution of a BusinessWorks application.

This preface gives some information on the TIBCO ActiveMatrix BusinessWorks ActiveAspects Plug-in documentation set, related documentation, and on the conventions used in TIBCO manuals.

Topics

- [Related Documentation, page vi](#)
- [Typographical Conventions, page viii](#)
- [Connecting with TIBCO Resources, page x](#)

Related Documentation

This section lists documentation resources you may find useful. The documentation road map shows the relationships between the the books and online references in this product's documentation set.

TIBCO ActiveMatrix BusinessWorks ActiveAspects Plug-in Documentation

The following documents form the TIBCO ActiveMatrix BusinessWorks ActiveAspects Plug-in documentation set:

- *TIBCO ActiveMatrix BusinessWorks ActiveAspects Plug-in Installation and Configuration* Read this manual for information on product installation.
- *TIBCO ActiveMatrix BusinessWorks ActiveAspects Plug-in User's Guide* Read this manual to learn how to develop, build, and deploy aspects in ActiveMatrix BusinessWorks.
- *TIBCO ActiveMatrix BusinessWorks ActiveAspects Plug-in API Reference* This manual gives information about the JAVA API for creating Advice Implementations in TIBCO ActiveMatrix BusinessWorks ActiveAspects Plug-in.
- *TIBCO ActiveMatrix BusinessWorks ActiveAspects Plug-in Getting Started* Read this manual for detailed information about creating and testing a ActiveMatrix BusinessWorks project using TIBCO Designer, setting development environment in Eclipse, advice development in Eclipse, and the expected final output.
- *TIBCO ActiveMatrix BusinessWorks ActiveAspects Plug-in 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

TIBCO ActiveMatrix BusinessWorks is a pre-requisite for TIBCO ActiveMatrix BusinessWorks ActiveAspects Plug-in and is used with other products. You may find it useful to read the documentation for the following TIBCO products:

TIBCO ActiveMatrix BusinessWorks

- *TIBCO ActiveMatrix BusinessWorks Concepts* Read this manual before reading any other manual in the documentation set. This manual describes terminology and concepts of TIBCO ActiveMatrix BusinessWorks, and the

other manuals in the documentation set assume you are familiar with the information in this manual.

- *TIBCO ActiveMatrix BusinessWorks Getting Started* This manual steps you through a very simple example of designing, deploying, and monitoring a ActiveMatrix BusinessWorks process.
- *ActiveMatrix BusinessWorks Process Design Guide* This manual describes how to create, edit, and test business processes using TIBCO ActiveMatrix BusinessWorks.
- *ActiveMatrix BusinessWorks Palette Reference* This manual describes each of the palettes available in TIBCO ActiveMatrix BusinessWorks.
- *TIBCO ActiveMatrix BusinessWorks Administration* This manual describes how to use TIBCO Administrator to deploy, manage, and monitor TIBCO ActiveMatrix BusinessWorks processes.
- *TIBCO ActiveMatrix BusinessWorks Installation* Read this manual for information on installing one or more components of ActiveMatrix BusinessWorks and setting up a ActiveMatrix BusinessWorks domain.
- *TIBCO ActiveMatrix BusinessWorks Error Codes* This manual describes errors returned by TIBCO ActiveMatrix BusinessWorks.
- *TIBCO ActiveMatrix BusinessWorks 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 Products

- TIBCO Designer™ software: TIBCO Designer is an easy to use graphical user interface for design-time configuration of TIBCO applications. TIBCO Designer includes online help for each palette.
- TIBCO Administrator™ software: TIBCO Administrator is the monitoring and managing interface for new-generation TIBCO products such as TIBCO ActiveMatrix BusinessWorks.
- TIBCO Adapter software
- Third-Party Documentation




Typographical Conventions

The following typographical conventions are used in this manual.

Table 1 General Typographical Conventions

Convention	Use
<i>TIBCO_HOME</i>	<p>Many TIBCO products must be installed within the same home directory. This directory is referenced in documentation as <i>TIBCO_HOME</i>. The value of <i>TIBCO_HOME</i> depends on the operating system. For example, on Windows systems, the default value is C:\tibco\bw\plugins.</p> <p>Incompatible products and multiple instances of the same product can be installed into different installation environments.</p>
<i>BW_HOME</i>	<p>TIBCO ActiveMatrix BusinessWorks ActiveAspects Plug-in installs into the BW directory within <i>TIBCO_HOME</i>. This directory is referenced in documentation as <i>BW_HOME</i>. The value of <i>BW_HOME</i> depends on the operating system. For example on Windows systems, the default value is C:\tibco\bw\5.9.</p>
code font	<p>Code font identifies commands, code examples, filenames, pathnames, and output displayed in a command window. For example:</p> <p>Use MyCommand to start the foo process.</p>
bold code font	<p>Bold code font is used in the following ways:</p> <ul style="list-style-type: none">• In procedures, to indicate what a user types. For example: Type 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	<p>Italic font is used in the following ways:</p> <ul style="list-style-type: none">• To indicate a document title. For example: See <i>TIBCO ActiveMatrix BusinessWorks Concepts</i>.• To introduce new terms For example: A portal page may contain several portlets. <i>Portlets</i> 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 <i>PathName</i>

Table 1 General Typographical Conventions (Cont'd)

Convention	Use
Key combinations	<p>Key name separated by a plus sign indicate keys pressed simultaneously. For example: Ctrl+C.</p> <p>Key names separated by a comma and space indicate keys pressed one after the other. For example: Esc, Ctrl+Q.</p>
	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.

Connecting with TIBCO Resources

How to Join TIBCOCommunity

TIBCOCommunity is an online destination for TIBCO customers, partners, and resident experts—a place to share and access the collective experience of the TIBCO community. TIBCOCommunity 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 TIBCOCommunity, 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 user name and password. If you do not have a user name, you can request one.

Chapter 1

About TIBCO ActiveMatrix BusinessWorks ActiveAspects Plug-in

TIBCO ActiveMatrix BusinessWorks ActiveAspects Plug-in (BWAA) extends the capabilities of TIBCO ActiveMatrix BusinessWorks by adding an Aspect Oriented Programming capability. This allows you to modularize and inject cross-cutting concerns to enhance your TIBCO ActiveMatrix BusinessWorks processes at deploy time while keeping the original TIBCO ActiveMatrix BusinessWorks process intact. It exposes a JAVA API to build JAR files that can alter the execution of a TIBCO ActiveMatrix BusinessWorks Application.

Topics

- [Overview, page 2](#)

Overview

Before you start working on the TIBCO ActiveMatrix BusinessWorks ActiveAspects Plug-in, read the *Overview* chapter in the *TIBCO ActiveMatrix BusinessWorks ActiveAspects Plug-in User's Guide* introduces the terminology that is used in this manual.

This documentation assumes that you are familiar with TIBCO ActiveMatrix BusinessWorks and TIBCO Designer. Refer to the product documentation for details. Also, read the *TIBCO ActiveMatrix BusinessWorks ActiveAspects Plug-in User's Guide* to get familiar with all features provided by TIBCO ActiveMatrix BusinessWorks ActiveAspects Plug-in.

[Chapter 2](#) provides detailed information about creating and testing a TIBCO ActiveMatrix BusinessWorks project using TIBCO Designer, setting development environment in Eclipse, advice development in Eclipse and finally getting the expected results.

Chapter 2 **Tutorial**

This chapter explains in detail the samples tutorial. This tutorial is a guideline to create a sample for your environment or work with the samples provided with this product.

Topics

- [Starting TIBCO Designer and Saving Your Project, page 4](#)
- [Creating the JMS Queue Sender Process, page 5](#)
- [Testing the TIBCO Designer Project, page 7](#)
- [Setting Development Environment in Eclipse, page 8](#)
- [Advice Development in Eclipse, page 9](#)

Starting TIBCO Designer and Saving Your Project

To start the TIBCO Designer and save your project, follow these steps:

1. From the Start menu, choose **All Programs > TIBCO > TIBCO Designer <version> > Designer <version>**.

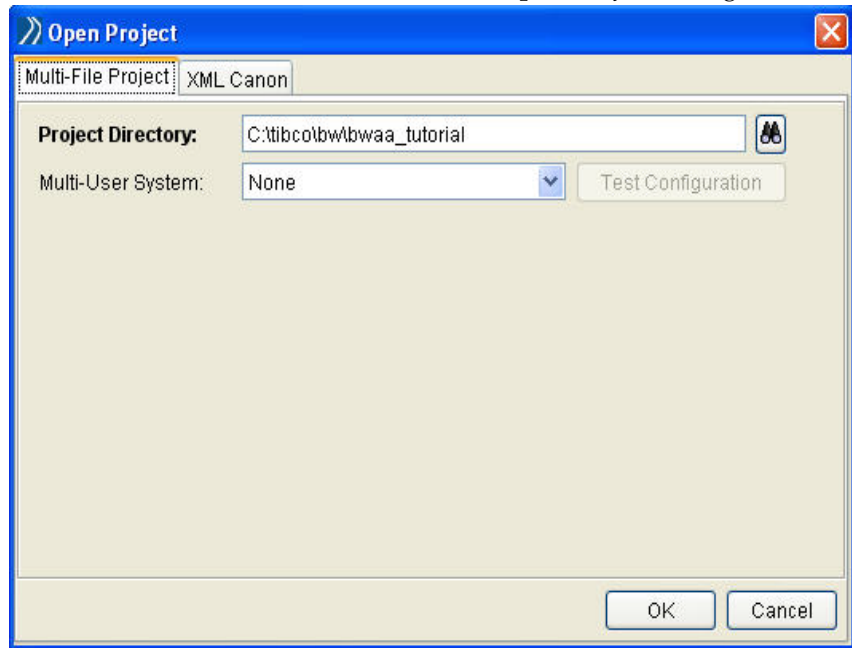
The TIBCO Designer Startup window appears.

2. Choose **New Empty Project**.

The TIBCO Designer main window appears, with the **Open Project** dialog on top.

In the **Open Project** dialog, make sure that the **Multi-File Project** tab is selected.

3. In the **Project Directory** field, click the **Browse** button to locate the directory in which you want to save the project. Locate the TIBCO ActiveMatrix BusinessWorks installation directory, and create a new subdirectory named `bwaa_tutorial`. Click **OK** to return to the **Open Project** dialog.



4. Click **OK** to create the new project.

You are now ready to create a process definition using TIBCO Designer.

Creating the JMS Queue Sender Process

To create the JMS Queue Sender process, follow these steps:

1. In TIBCO Designer, select the project tree (bwaa_tutorial) in the project panel.
2. Add JMS Shared Connection in the project panel and configure the JMS connection.



If using the default localhost, TIBCO Enterprise Messaging Service must be installed on the same machine. If not, then point to the location where TIBCO EMS is installed and running.

3. In the palette panel, select the **Process** palette.
4. From the palette panel, drag a **Process Definition** resource into the design panel.
5. In the configuration panel, type the name **JMSClient** in the **Name** field to rename the process. Click **Apply**.
6. From the palette panel, drag another **Process Definition** resource into the design panel.
7. In the configuration panel, type the name **JMSServer** in the **Name** field to rename the process. Then click **Apply**.
8. Save your project by choosing **Project > Save** from the menu.

To add activities to the process, follow these steps:

1. Select the **JMSClient** process in the project tree.
The **Start** and **End** activities should be displayed in the design panel.
2. Find the **JMS** palette in the palette panel and select it.
3. Drag a **JMS Queue Sender** activity into the design panel.

4. With the JMS Queue Sender still selected, enter the following values in the configuration panel:
 - a. **Name** the JMS Queue Sender as `Client`.
 - b. Click the **Browse** button to the right of the **JMS Connection** field.
 - c. Select **JMS Connection** in the **Add a Resource** dialog and click **OK**.
 - d. Click **Browse** button to the right of the **Destination Queue** field.
 - e. In the **JMS Destination name browser** dialog, select **sample** and click **Set**.
 - f. In the **Input** panel, provide an input value to the **Body**; for example, `HelloWorld`.
5. Select `JMS Server` process and drag a **JMS Queue Receiver** activity into the design panel.
6. With the **JMS Queue Receiver** still selected, enter the following values in the configuration panel:
 - a. **Name** the JMS Queue Receiver as `JMS Server`.
 - b. Click the **Browse** button to the right of the **JMS Connection** field.
 - c. Select **JMS Connection** in the **Add a Resource** and click **OK**.
 - d. Click **Browse** button to the right of the **Destination Queue** field.
 - e. In the **JMS Destination name browser** dialog, select **sample** and click **Set**.

Testing the TIBCO Designer Project

You can test the processes directly from TIBCO Designer. This allows you to ensure that they work correctly. Follow these steps:

1. Click the **Tester** tab to the left of the project panel. The test panel replaces the project tree.
2. Click the **Start testing viewed process** button. In the process selection window that appears, select the **Client process** and **Server process**. Click **Load and Selected**.

The process is now in **Test** mode.

Setting Development Environment in Eclipse

This section explains the process about how an Eclipse JAVA developer can develop an ActiveMatrix BusinessWorks aspect and subsequently remotely debug it in Eclipse after deploying the aspect in a TIBCO ActiveMatrix BusinessWorks engine instance.

Product Libraries

The installation of TIBCO ActiveMatrix BusinessWorks ActiveAspects Plug-in contains the following jars:

1. `poa-core.jar` and `poa-bwaa.jar` - This is the core runtime library of this product.
2. `poa-api.jar` - Contains the public API of this product.
3. `gxml.jar`
`gxmlBridges.jar`
`gxmlProcessors.jar`
4. `poa-samplesImpl.jar` and `poa-bwaa-samplesImpl-src.jar` - Contains all the advice implementations for the sample advices.
5. `bwaa-palette.jar` - Contains information about the BWAA palette.
6. `bwconfig-api.jar` - Contains all the interfaces to get activity configuration data.
7. `bwconfig-impl.jar` - Contains all the Provider implementations to get read-only access to activity configuration data.

Advice Development in Eclipse

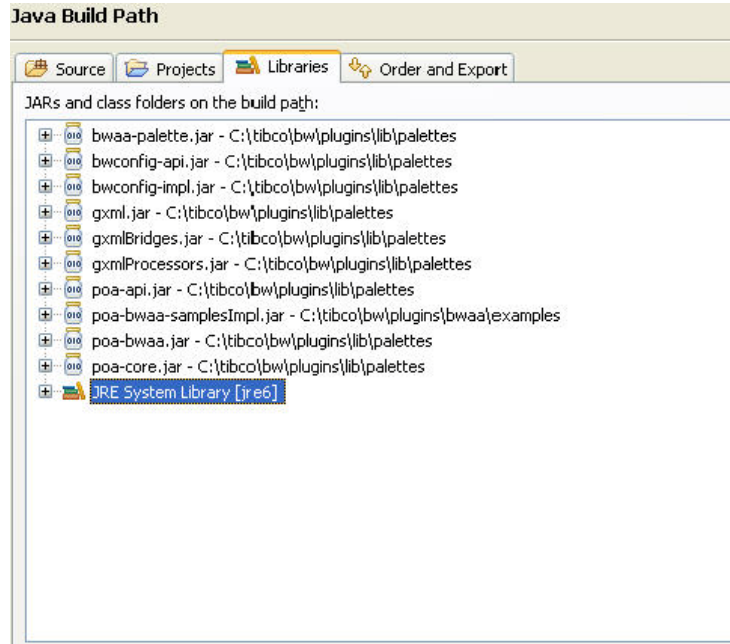
Setting up Eclipse for TIBCO ActiveMatrix BusinessWorks ActiveAspects Plug-in Development

To setup Eclipse for the TIBCO ActiveMatrix BusinessWorks ActiveAspects Plug-in development do the following:

1. Create a JAVA Eclipse project.
2. Click **Add External JARs** to add all the required JARs contained in the installation to the build path and click **Finish**.
 - poa-core.jar and poa-bwaa.jar
 - poa-api.jar
 - gxml.jar
 - gxmlBridges.jar
 - gxmlProcessors.jar
 - poa-samplesImpl.jar and poa-bwaa-samplesImpl-src.jar
 - bwaa-palette.jar
 - bwconfig-api.jar
 - bwconfig-impl.jar
3. To attach the source to the samplesImpl.jar, right-click on the JAR in the Referenced Libraries in the Eclipse Project Explorer and select **properties**. In the **Java Source Attachment**, select **External File** and browse to **samplesImpl-src.jar**.

This will attach the source code to the JAVA classes.

Figure 1 Jars and Class Folders on the Build Path



4. To attach the source to the `samplesImpl.jar`, right-click on the JAR in the **Referenced Libraries** in the Eclipse Project Explorer and select **properties**. In the JAVA Source Attachment, select **External File** and browse to **samplesImpl-src.jar**.

This will attach the source code to the JAVA classes.

Creating a Synchronous Advice Implementation

To create a synchronous advice implementation, you have to extend the `SyncAdvice` class.

1. Create a new class in the JAVA Eclipse Project.
2. Specify the following fields:
 - Package - `com.bwaa.tutorial`
 - Name - `JMSDocumentLogger`
3. Extend the `SyncAdvice` (for Sync Advice Implementations) class with the required generics.
4. Annotate the class with `@AdviceImpl`.

Figure 2 Creating a Synchronous Advice Implementation

```

JMSDocumentLogger.java
package com.bwaa.tutorial;

import java.io.StringWriter;

import com.tibco.bw.poa.runtime.AspectException;
import com.tibco.bw.poa.runtime.AspectProcessContext;
import com.tibco.bw.poa.runtime.SyncAdvice;
import com.tibco.bw.poa.runtime.annotation.AdviceImpl;
import com.tibco.bw.poa.samples.GxmlUtils;

@AdviceImpl
public class JMSDocumentLogger<I, U, N extends I, A extends I, S, T, X> extends SyncAdvice<I, U, N, A, S, T, X> {
    @Override
    public N execute(N input, AspectProcessContext<I, U, N, A, S, T, X> context) throws AspectException {
        System.out.println("-----[ADVICE EXECUTING: Document Logger]-----");
        if (input != null)
        {
            System.out.println("The document received by the advice is :");
            StringWriter sw = new StringWriter();
            GxmlUtils.serializeNode(input, getAdviceContext().getGxProcessingContext(), sw);
            System.out.println(sw.toString());
        }
        else
        {
            System.out.println("The document received by the advice is null.");
        }
        System.out.println("-----[ADVICE EXECUTION DONE: Document Logger]-----");

        return input;
    }
}

```

Exporting the Advice Implementation JAR file using Eclipse

To export the advice implementation JAR file, do the following:

1. Right-click on the project (in this case `com.bwaa.tutorial`) and select **Java > JAR File**.

Make sure that **Export generated classfiles and resources** check box is selected.

2. Click **Browse** in the JAR file field and select the export destination. For example,
`C:\tibco_bwaa\bw\plugins\bwaa\lib\bwaaTutorialAdviceImpl.jar`.



The export destination must be in the classpath of BWAA. The recommended export location is `$TIBCO_HOME\bw\plugins\bwaa\lib`.

3. Click **Finish**.



To run the scenario, you must copy the `aspect.jar` (in this case, `JMSLoggerAspect.jar`) file in the `bw\plugins\bwaa\aspects` folder.

Creating a New Aspect File

Create a new aspect file similar to the following XML snippet.

It may change as per your requirement.

```
<?xml version = "1.0" encoding = "UTF-8"?>
<aspect order="1" targetNamespace = "http://example.org/Eight"
xmlns = "http://schemas.tibco.com/bw/poa" xmlns:xsi =
"http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation =
"http://schemas.tibco.com/bw/poa BWAspect.xsd">
<pointcut name = "JMSActivities">
<query queryLanguage =
"http://schemas.tibco.com/bw/poa/pointCutSelectionLanguage">activi
ty (name = "JMS*")</query>
</pointcut>

<advice name = "JMSLoggerAdvice" pointcut = "JMSActivities">
<documentation>Document Logger for JMS activities</documentation>
<activity where = "After">
<implementation.java className =
"com.bwaa.tutorial.JMSDocumentLogger"/>
</activity>
</advice>
</aspect>
```

Now validate this XML against the bwaspect.xsd file.

Creating an Aspect JAR

Create an aspect JAR with the following structure:

Figure 3 Aspect JAR Structure

Name	Type	Path
Aspect.bwaspect	BWASPECT File	aspects\
Bwpoatest.amf	AMF File	meta-inf\
Manifest.mf	MF File	meta-inf\



To run the scenario, you must copy the aspect.jar (in this case, JMSLoggerAspect.jar) file in the bw\plugins\bwaa\aspects folder.

Using TIBCO Designer with TIBCO ActiveMatrix BusinessWorks ActiveAspects Plug-in

The following steps provide the detailed information about using TIBCO Designer with this example.

1. Open `$TIBCO-HOME\Designer\5.7\bin\designer.tra` and append the aspect lib folder to the property `tibco.env.CUSTOM_CP_EXT`. The default aspect lib folder is `$TIBCO-HOME\bw\plugins\bwaa\lib`.
2. Create a properties file, for example, `C:\test\props.cfg`, and add the following properties to it:
 - `java.property.aspectPath C:/tibco/bw/plugins/bwaa/aspects`
 - `ServiceAgent.poa.serviceagent.Class=com.tibco.bw5.poa.core.runtime.DefaultBw5AspectServiceAgentImpl`
 - `Jmx.Enabled=true`

(Replace the value of the `aspectPath` property with the appropriate value based on your environment)
3. Start TIBCO Designer and open the project that you want to debug.
4. Go to the **Tester** tab, click the **Start** button to start debugging your project and then click the **Advanced** button to launch the advanced configuration. In the **Test Engine User Args** field, introduce the following:
 - `-p C:\test\props.cfg`

(Replace the path with the appropriate one based on where you saved the properties file that was created at step #2)
5. Start debugging your project.
6. At this point, the engine should also execute the aspects that are available in `aspectPath`. Ensure that all advice implementations as well as their dependent JARs are available in the classpath.

Expected Final Output

The final output should be as follows:

```
-----[ADVICE EXECUTING: Document Logger]-----
The document received by the advice is :
<?xml version="1.0" encoding="UTF-8"?>
<jms1:aEmptyOutputClass
xmlns:jms1="http://www.tibco.com/namespaces/tnt/plugins/jms">
<jms1:MessageID>ID:EMS-SERVER-HGAWANDE.E684E896BACAD:12</jms1:MessageID>
</jms1:aEmptyOutputClass>
-----[ADVICE EXECUTION DONE: Document Logger]-----
-----[ADVICE EXECUTING: Document Logger]-----
The document received by the advice is :
```

```

<?xml version="1.0" encoding="UTF-8"?>
<jms1:ActivityOutput
xmlns:jms1="http://www.tibco.com/namespaces/tnt/plugins/jms">
<JMSHeaders>
    <jms1:JMSDestination>sample</jms1:JMSDestination>
    <jms1:JMSDeliveryMode>PERSISTENT</jms1:JMSDeliveryMode>

<jms1:JMSMessageID>ID:EMS-SERVER-HGAWANDE.E684E896BACAD:12</jms1:J
MSMessageID>
    <jms1:JMSTimestamp>1317806706130</jms1:JMSTimestamp>
    <jms1:JMSEExpiration>0</jms1:JMSEExpiration>
    <jms1:JMSRedelivered>false</jms1:JMSRedelivered>
    <jms1:JMSPriority>4</jms1:JMSPriority>
</JMSHeaders>
<JMSProperties/>
<Body>HelloWorld</Body>
</jms1:ActivityOutput>
-----[ADVICE EXECUTION DONE: Document Logger]-----

```


Index

C

Creating a .bwaspect File [11](#)
Creating a New Aspect File [11](#)
Creating a Synchronous Advice Implementation [10](#)
customer support [x](#)

E

Expected Final Output [13](#)
Exporting the Advice Implementation JAR file using
Eclipse [11](#)
Exporting the JAR file using Eclipse [12](#)

O

overview [2](#)

S

Setting up Eclipse for TIBCO ActiveMatrix Business-
Works ActiveAspects Plug-in Development [9](#)
support, contacting [x](#)

T

technical support [x](#)
TIBCO_HOME [viii](#)
Tutorial [3](#)

U

Using TIBCO Designer with TIBCO ActiveMatrix
BusinessWorks ActiveAspects Plug-in [12](#)