

TIBCO ActiveMatrix[®] Adapter for PeopleSoft

Examples

*Software Release 6.1
October 2012*

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, The Power of Now, TIBCO ActiveMatrix Adapter for PeopleSoft, TIBCO ActiveMatrix BusinessWorks, TIBCO ActiveMatrix BusinessWorks Service Engine, TIBCO ActiveMatrix Service Bus, TIBCO ActiveMatrix Service Grid, TIBCO Adapter SDK, TIBCO Administrator, TIBCO Business Studio, TIBCO Designer, TIBCO Designer Add-in for TIBCO Business Studio, TIBCO Enterprise Message Service, TIBCO Hawk, TIBCO Rendezvous, and TIBCO Runtime Agent are either registered trademarks or trademarks of TIBCO Software Inc. in the United States and/or other countries.

EJB, 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 © 1998-2012 TIBCO Software Inc. ALL RIGHTS RESERVED.

TIBCO Software Inc. Confidential Information

Contents

Preface	vii
Changes from the Previous Release of This Guide	viii
Related Documentation	ix
TIBCO ActiveMatrix Adapter for PeopleSoft Documentation	ix
Other TIBCO Product Documentation	ix
Third-Party Documentation	x
Typographical Conventions	xi
Terminology and Acronyms	xiv
Connecting with TIBCO Resources	xvi
How to Join TIBCOCommunity	xvi
How to Access TIBCO Documentation	xvi
How to Contact TIBCO Support	xvi
 Chapter 1 Introduction	 1
Examples Overview	2
Component Interfaces	2
Operations	3
Prerequisites of Running the Examples	6
The Adapter Examples Files	6
Loading the PeopleSoft Project for Examples	6
Using TIBCO ActiveMatrix BusinessWorks	8
Importing a Project from a ZIP Archive in TIBCO Designer	8
 Chapter 2 TIBCO ActiveMatrix BusinessWorks: Working with the Customer Component Interface ..	 11
Example Description	12
Setting Up the Example	13
Deploying and Running the Example	14
Expected Results	16
 Chapter 3 TIBCO ActiveMatrix BusinessWorks: Working with the SalesOrder Component Interface ..	 17
Example Description	18
Setting Up the Example	19
Testing the Example	21
Expected Results	22

Chapter 4 TIBCO ActiveMatrix BusinessWorks: Working with the PurchaseOrder Component Interface.	23
Example Description.	24
Setting Up the Example	25
Testing the Example	26
Expected Results.	27
 Chapter 5 TIBCO ActiveMatrix BusinessWorks: Working with the Employee Component Interface .	 29
Example Description.	30
Setting Up the Example	31
Testing the Example	33
Expected Results.	37
 Chapter 6 TIBCO ActiveMatrix BusinessWorks: Working with the ItemMaster Component Interface	 39
Example Description.	40
Setting Up the Example	41
Testing the Example	43
Expected Results.	47
 Chapter 7 TIBCO ActiveMatrix BusinessWorks: Working with the UOM Message	 49
Example Description.	50
Setting Up the Example	51
Preparing PeopleSoft Integration Broker	53
Testing the Example	55
Expected Results.	56
 Chapter 8 TIBCO ActiveMatrix BusinessWorks: Working with the Contact Message	 57
Example Description.	58
Setting Up the Example	59
Preparing PeopleSoft Integration Broker	61
Testing the Example	62
Expected Results.	63
 Chapter 9 TIBCO ActiveMatrix BusinessWorks: Working with the Country Message.	 67
Example Description.	68
Setting Up the Example	69
Preparing PeopleSoft Integration Broker	71

Testing the Example 72

Expected Results..... 73

Index 77

Preface

TIBCO ActiveMatrix Adapter for PeopleSoft is a bi-directional gateway between the PeopleSoft application and applications configured for the TIBCO environment. It is assumed that you are familiar with the basic concepts of the TIBCO environment and PeopleSoft technology.

Topics

- [Changes from the Previous Release of This Guide, page viii](#)
- [Related Documentation, page ix](#)
- [Typographical Conventions, page xi](#)
- [Terminology and Acronyms, page xiv](#)
- [Connecting with TIBCO Resources, page xvi](#)

Changes from the Previous Release of This Guide

This section itemizes the major changes from the previous release of this guide.

Update all the Procedures in the Document

The setup procedures for the examples developed with PeopleTools 8.4x are updated with this release.

Related Documentation

This section lists documentation resources you may find useful.

TIBCO ActiveMatrix Adapter for PeopleSoft Documentation

The following documents form the TIBCO ActiveMatrix Adapter for PeopleSoft documentation set:

- *TIBCO ActiveMatrix Adapter for PeopleSoft Concepts* Read this manual to gain an understanding of the product that you can apply to the various tasks you may undertake.
- *TIBCO ActiveMatrix Adapter for PeopleSoft Installation* Read this manual for instructions on site preparation and installation.
- *TIBCO ActiveMatrix Adapter for PeopleSoft Configuration and Deployment* Read this manual for instructions on how to create, configure, and deploy adapter projects.
- *TIBCO ActiveMatrix Adapter for PeopleSoft Examples* Read this manual to work through the examples provided with the adapter.
- *TIBCO ActiveMatrix Adapter for PeopleSoft 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 ActiveMatrix BusinessWorks™
- TIBCO ActiveMatrix BusinessWorks™ Service Engine
- TIBCO ActiveMatrix® Service Bus
- TIBCO ActiveMatrix® Service Grid
- TIBCO Adapter® SDK
- TIBCO Administrator™
- TIBCO Business Studio™
- TIBCO Designer™
- TIBCO Enterprise Message Service™
- TIBCO Hawk®

- TIBCO Rendezvous[®]
- TIBCO Runtime Agent[™]

Third-Party Documentation

You may also find it useful to read the following documentation:

- "Integration Tools: PeopleSoft Component Interfaces" in *PeopleTools PeopleBook*. Topics include an explanation of Component Interfaces, how to create them, how to test them using PeopleTools, and how to validate that a Component Interface is compatible with the underlying component.
- "PeopleSoft Integration Broker" in *PeopleTools PeopleBook*. Topics include an explanation of Integration Broker, the components involved, and how to use the components in Application Messaging.
- "PeopleSoft Platforms", a link from the PeopleSoft Customer Connection web site (see <http://www.peoplesoft.com>), for the versions of the databases supported by PeopleSoft.
- "PeopleSoft Internet Architecture Adminstrating", a link from the Peoplebook-PeopleTools-Administration Tools web site (see <http://www.peoplesoft.com>), for information on configuring a JOLT Listener port.

Typographical Conventions

The following typographical conventions are used in this manual.

Table 1 General Typographical Conventions

Convention	Use
<i>ENV_NAME</i>	TIBCO products are installed into an installation environment. A product installed into an installation environment does not access components in other installation environments. Incompatible products and multiple instances of the same product must be installed into different installation environments.
<i>TIBCO_HOME</i>	
	<p>An installation environment consists of the following properties:</p> <ul style="list-style-type: none"> • Name Identifies the installation environment. This name is referenced in documentation as <i>ENV_NAME</i>. On Microsoft Windows, the name is appended to the name of Windows services created by the installer and is a component of the path to the product shortcut in the Windows Start > All Programs menu. • Path The folder into which the product is installed. This folder is referenced in documentation as <i>TIBCO_HOME</i>.
<i>TIB_ADPSFT8_HOME</i>	TIBCO ActiveMatrix Adapter for PeopleSoft installs into a directory within a <i>TIBCO_HOME</i> . This directory is referenced in documentation as <i>TIB_ADPSFT8_HOME</i> . The default value of <i>TIB_ADPSFT8_HOME</i> depends on the operating system. For example on Windows systems, the default value is C:\tibco\adapter\adpsft8\6.1.
<i>TIB_EMS_HOME</i>	TIBCO Enterprise Message Service installs into a directory within <i>TIBCO_HOME</i> . This directory is referenced in documentation as <i>TIB_EMS_HOME</i> . The value of <i>TIB_EMS_HOME</i> depends on the operating system. For example on Windows systems, the default value is C:\tibco\ems\6.3.
<i>PS_HOME</i>	<i>PS_HOME</i> is the directory where the PeopleSoft Application server is installed. For example, if the PeopleSoft Application server is installed at C:\PT8.49 on a Windows system, the value of <i>PS_HOME</i> is C:\PT8.49.
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>

Table 1 General Typographical Conventions (Cont'd)




Convention	Use
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]
<i>italic font</i>	<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>
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>
	<p>The note icon indicates information that is of special interest or importance, for example, an additional action required only in certain circumstances.</p>
	<p>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.</p>
	<p>The warning icon indicates the potential for a damaging situation, for example, data loss or corruption if certain steps are taken or not taken.</p>

Table 2 Syntax Typographical Conventions

Convention	Use
[]	<p>An optional item in a command or code syntax.</p> <p>For example:</p> <p>MyCommand [optional_parameter] required_parameter</p>
	<p>A logical OR that separates multiple items of which only one may be chosen.</p> <p>For example, you can select only one of the following parameters:</p> <p>MyCommand param1 param2 param3</p>
{ }	<p>A logical group of items in a command. Other syntax notations may appear within each logical group.</p> <p>For example, the following command requires two parameters, which can be either the pair param1 and param2, or the pair param3 and param4.</p> <p>MyCommand {param1 param2} {param3 param4}</p> <p>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:</p> <p>MyCommand {param1 param2} {param3 param4}</p> <p>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.</p> <p>MyCommand param1 [param2] {param3 param4}</p>

Terminology and Acronyms

The following acronyms are used in this manual:

Table 3 Terminology and Acronyms

Acronym	Meaning
API	Application Programming Interface
QOS	TIBCO Rendezvous quality of service
GUI	Graphical User Interface
RV	Refers to TIBCO Rendezvous reliable message quality of service, as opposed to certified message.
RVCM	Refers to TIBCO Rendezvous certified message quality of service.
RVDQ	Refers to TIBCO Rendezvous distributed queue.
RPC	Remote Procedural Call
TRA	TIBCO Runtime Agent
JMS	Java Message Service
Component Interface (CI)	A Component Interface is a PeopleTools object that you create in PeopleSoft Application Designer. It exposes a PeopleSoft component for synchronous access from another application. External applications need not be concerned with the details of page structures and component definitions in order to access the underlying data and business logic through Component Interfaces.
TIB Work Page	This page is bundled with the TIB_PS8_ADAPTER project. It implements the logic for capturing desired fields for publication at runtime. The page must be attached to the component from which data is captured.
Inbound	Events coming into the adapter. It refers to the inflow of data into the PeopleSoft application. It is applicable to Subscription and Request-Response services.
Outbound	Events going out from the adapter. It refers to data being captured from the PeopleSoft application and being sent out. It is applicable to the Publication Service.
PIA	PeopleSoft Pure Internet Architecture

Table 3 Terminology and Acronyms (Cont'd)

Acronym	Meaning
Integration Broker (IB)	An Integration Broker facilitates exposing PeopleSoft business logic as services and consuming external web services for PeopleSoft applications to invoke. PeopleSoft Integration Broker also supports synchronous and asynchronous messaging and uses a variety of communication protocols, while managing message structure, message content, and transport disparities.
VPD	Vital Product Database

Connecting with TIBCO Resources

How to Join TIBCOCommunity

TIBCOCommunity 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. TIBCOCommunity offers forums, blogs, and access to a variety of resources. To register, go to <http://www.tibcommunity.com>.

How to Access TIBCO Documentation

You can access TIBCO documentation here:

<http://docs.tibco.com>

How to Contact TIBCO Support

For comments or problems with this manual or the software it addresses, 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 **Introduction**

This chapter contains an overview of the examples and the prerequisites needed to run the examples.

Topics

- [Examples Overview, page 2](#)
- [Prerequisites of Running the Examples, page 5](#)

Examples Overview

The examples in this guide demonstrate the integration between the adapter and two TIBCO products, TIBCO ActiveMatrix BusinessWorks and TIBCO IntegrationManager. We have chosen commonly used component interfaces to build an end-to-end enterprise wide integration and demonstrate the adapter capabilities:

- [Component Interfaces, page 2](#)
- [Operations, page 3](#)

Component Interfaces

The following table shows the component interface included in the examples, the corresponding PeopleSoft module and the PeopleTools version used to develop the respective component interface.

Table 4 Examples and PeopleTools Version

Component Interface/ Application Message	Module	PeopleTools Version
Customer	Financials/SCM 8	8.18
SalesOrder	Financials/SCM 8	8.41
PurchaseOrder	Financials/SCM 8	8.18
Employee	HRMS 8.9	8.45
ItemMaster	CRM 8.9	8.45
UOM	CRM 8.9	8.45
Contact	CRM 8.9	8.45
Country	HRMS /CRM 8.9	8.45

The component interfaces developed with PeopleTools 8.18 can be used with PeopleTools versions 8.1x and the component interfaces developed with PeopleTools 8.41/8.45 can be used with PeopleTools version 8.4x subject to the condition that the adapter supports the PeopleTools version



To make the examples work with later versions of PeopleTools, you may need to make changes to the component interfaces.

Operations

The examples provide different operations on each of these component interfaces. A brief description of these operations is given next.

Get

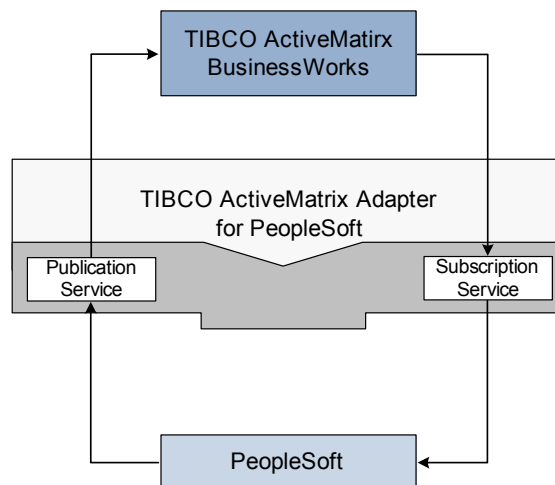
In this operation, TIBCO ActiveMatrix BusinessWorks requests data to be fetched from the PeopleSoft application. The Request-Response Service is used to simulate this. TIBCO ActiveMatrix BusinessWorks reads the request from an XML file and sends the request. The corresponding Request-Response Service is executed on the adapter and the result is returned to TIBCO ActiveMatrix BusinessWorks. TIBCO ActiveMatrix BusinessWorks maps the results according to business requirements and the data is written to an XML file.

Update

Another important requirement in an integration scenario is to update data in an existing record. The Update operation is simulated in two ways.

- The first is a publish-subscribe scenario. Data is published by the adapter's Publication Service; the values that need to be updated are modified using a TIBCO ActiveMatrix BusinessWorks process. The updated message is sent to the adapter's Subscription Service and the service updates the values in PeopleSoft.

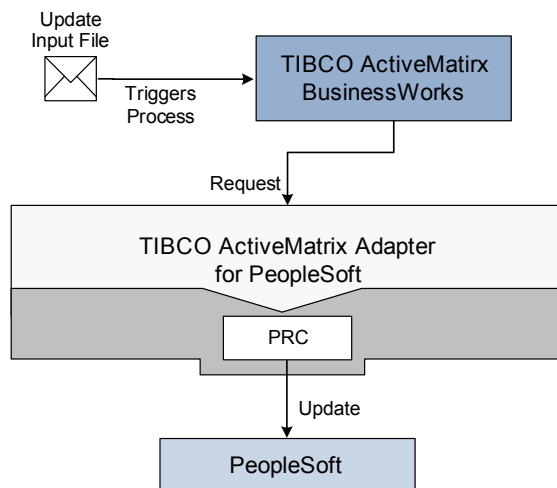
Figure 1 Update Operation Using a Publish-Subscribe Scenario



- The second is a Request-Response scenario. The values are modified in the mapper class in TIBCO ActiveMatrix BusinessWorks and the request is sent to the adapter.

The Request-Response Service of the adapter receives the request and updates the values in PeopleSoft.

Figure 2 Update Operation Using an Request-Response Scenario



The publish-subscribe scenario is used for the ItemMaster component interface. The Request-Response scenario is used for the Employee and SalesOrder Component Interfaces.

Synchronize

In this operation, TIBCO ActiveMatrix Adapter for PeopleSoft publishes a message if there are any changes in the PeopleSoft target application. TIBCO ActiveMatrix BusinessWorks receives this message, does the necessary mapping of the fields and writes them to an XML file.

For publishing an event from PeopleSoft, see *TIBCO ActiveMatrix Adapter for PeopleSoft Configuration and Deployment*.

Prerequisites of Running the Examples

This section explains the prerequisites of running the examples:

- [The Adapter Examples Files, page 5](#)
- [Using TIBCO ActiveMatrix BusinessWorks, page 7](#)
- [Importing a Project from a ZIP Archive in TIBCO Designer, page 7](#)

The Adapter Examples Files

The adapter software includes the following files that are required when you run the examples:

- The PeopleSoft Project file:

The adapter includes two pre-built PeopleSoft projects (TIB_EXAMPLES) that contain the component interfaces to be used in PeopleTools 8.1x and PeopleTools 8.4x.

The TIB_EXAMPLES project to be imported into PeopleTools 8.1x is located in *TIB_ADPSFT8_HOME*\examples\TIB_EXAMPLES_PT81x folder.

The TIB_EXAMPLES project to be imported into PeopleTools 8.4x is located in *TIB_ADPSFT8_HOME*\examples\TIB_EXAMPLES_PT84x folder.

You must set up the projects in your PeopleSoft database before running the examples. See [Loading the PeopleSoft Project for Examples on page 5](#).

- The ZIP archives:

The example ZIP files are included in the adapter installation. After installation, the files are installed in the following directories:

— Component Interface examples:

TIB_ADPSFT8_HOME\examples\BusinessWorks\

— Application message examples:

TIB_ADPSFT8_HOME\examples\IntegrationBroker\

You cannot directly open a ZIP archive in TIBCO Designer and make modifications to the configurations. To work with a ZIP file, see [Importing a Project from a ZIP Archive in TIBCO Designer on page 7](#)

Loading the PeopleSoft Project for Examples

To import the TIB_EXAMPLES project into PeopleTools:

1. Open PeopleTools Application Designer and log into the server where you want to add the project.
2. Verify that there is no existing TIB_EXAMPLES project in the database:
 - a. Select **File > Open**.
 - b. In the Open Definition dialog, select **Project** from the Definition drop-down list and enter **TIB_EXAMPLES** in the Name field of the Selection Criteria area. If there is no existing TIB_EXAMPLES project, skip to [step 3](#).
 - c. Open the existing TIB_EXAMPLES project and delete all the component interfaces.
 - d. Delete the TIB_EXAMPLES project itself.
 - e. If necessary, also delete the cache.
 - f. Exit PeopleTools Application Designer and then log in again.
3. Import the project:
 - a. In PeopleTools 8.1x, select **File > Copy Project from File**.
In PeopleTools 8.4x, select **Tools > Copy Project > From File**.
 - b. In the displayed Copy From File: Select Project dialog, browse to locate the `TIB_ADPSFT8_HOME\examples\TIB_EXAMPLES_PT84x` directory. The TIB_EXAMPLES project is displayed in the list. Select the TIB_EXAMPLES project to open it.
 - c. In the displayed Copy From File dialog, make sure all the definition types are selected, then click **Copy**.
4. Grant permissions to work on the component interfaces:
 - a. Log in to PIA and navigate to **PeopleTools > Security > Permission & Roles > Permission Lists**.
 - b. Choose the appropriate permission list for the account used by the adapter.
 - c. Grant permissions to work on the component interfaces you have just imported. Make sure full access is given to all the component interfaces for the account being used by the adapter.



While running the examples with a particular version of PeopleTools, make sure that the classpath in the .tra file of the example is pointing to the correct version of PSJOA.JAR.

For additional information, see "Loading the Adapter PeopleSoft Project", in the *TIBCO ActiveMatrix Adapter for PeopleSoft Configuration and Deployment* guide.

Using TIBCO ActiveMatrix BusinessWorks

If you are using the adapter with TIBCO ActiveMatrix BusinessWorks, you must install the following TIBCO software products to run the examples:

- TIBCO Rendezvous
- TIBCO Runtime Agent
- TIBCO Enterprise Message Service (EMS)
- TIBCO ActiveMatrix BusinessWorks

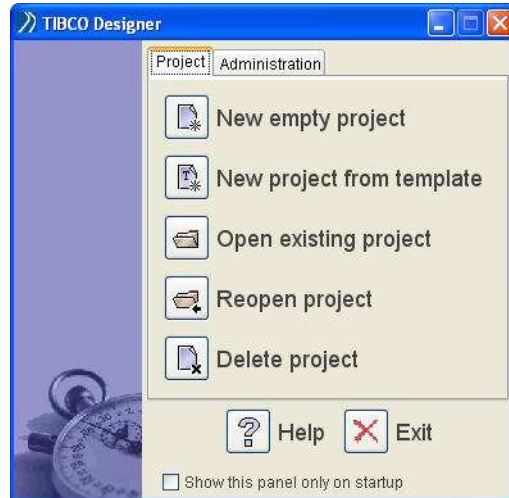


The EMS server must be running and accessible to the machine on which the adapter is installed.

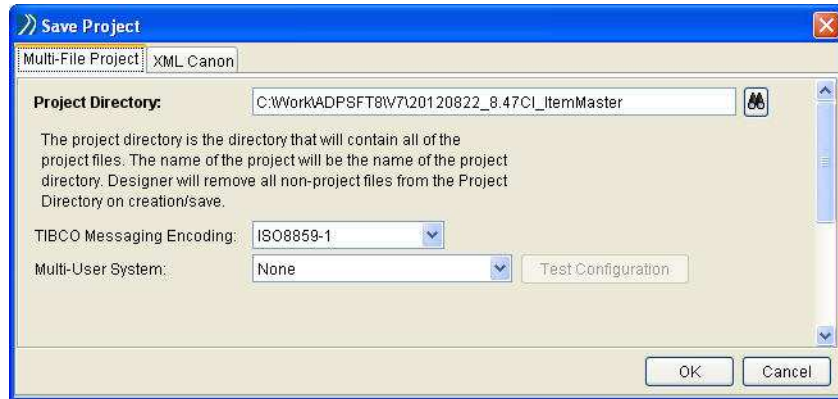
Importing a Project from a ZIP Archive in TIBCO Designer

To import a project from a ZIP archive:

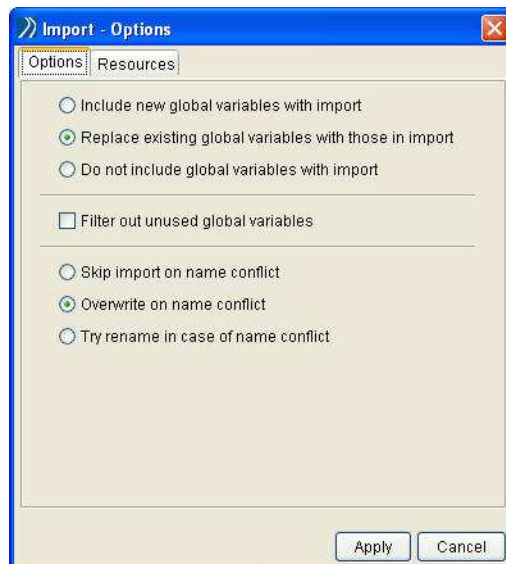
1. Open Designer.
2. In the first dialog that is displayed, click the **Project** tab and click the **New Empty Project** button in the tab. The dialog is shown next.



3. In the Save Project dialog, click the Multi-File Project tab. In the Project Directory field, specify or navigate to the directory where you want to save the project. Click **OK**.



4. In the opened Designer window, select **Project > Import Full Project** from the menu.
5. In the displayed Import Project dialog, click the ZIP Archive dialog. In the File field, navigate to the directory where the ZIP file is located. Click **OK**.
6. In the displayed Import - Options dialog:
 - a. Select the **Replace Existing Global Variables With Those In Import** radio button.
 - b. Select the **Overwrite On Name Conflict** radio button.
 - c. Click **Apply**.



The project is imported in Designer.

Chapter 2

TIBCO ActiveMatrix BusinessWorks: Working with the Customer Component Interface

This example shows how to use several adapter services within a TIBCO ActiveMatrix BusinessWorks process to get, update, and synchronize a Customer record stored in PeopleSoft. The example is deployed and run using the TIBCO Administrator GUI.

Topics

- [Example Description, page 12](#)
- [Setting Up the Example, page 13](#)
- [Deploying and Running the Example, page 14](#)
- [Expected Results, page 16](#)

Example Description

This example shows how the adapter's Publication, Subscription and Request-Response services can be used in a TIBCO ActiveMatrix BusinessWorks process. The example has three processes, `GetProcess`, `UpdateProcess` and `SyncProcess`.

- `GetProcess` uses the adapter's Request-Response service to fetch data from PeopleSoft for key values that you specify in an input XML file.

`GetProcess` queries PeopleSoft based on the given key values and fetches the data. The data is sent back by the adapter to TIBCO ActiveMatrix BusinessWorks and is written into an output XML file.

- `UpdateProcess` uses the adapter's Publication and Subscription services to update PeopleSoft.

In a typical business scenario, an adapter to another application (for example, SAP or Siebel) publishes a message into the TIBCO environment. TIBCO ActiveMatrix BusinessWorks receives the messages and writes them to an input XML file. The content of this XML file will contain the changes to be updated in PeopleSoft. TIBCO ActiveMatrix Adapter for PeopleSoft uses the Subscription service to update the data in PeopleSoft. In Customer example provided, the publisher is also a PeopleSoft adapter.

Alternatively, you may specify key values in the input XML file. `UpdateProcess` queries PeopleSoft based on the key values. The Customer details are updated in PeopleSoft.

- `SyncProcess` uses the adapter's Publication service to send a message out of PeopleSoft whenever Customer Component Interface details are modified or updated in the target PeopleSoft application. Any change to the data in PeopleSoft triggers the `SyncProcess`.

On receiving the modified data, TIBCO ActiveMatrix BusinessWorks logs it to an XML file.

Setting Up the Example

Before starting the example you must import the sample project zip file and save it in a new project.

In TIBCO Designer:

1. In the initial dialog box click **Open New Project** and specify a name for the project.
2. Click **Project>Import Full Project**.
3. Click the **Zip Archive** tab and browse to locate the `Customer.zip` file. The file is located in the `TIB_ADPSFT8_HOME/examples/BusinessWorks/Customer` folder.
4. Go to the Publication service instance.
5. Click the **Design-time Connection** tab.
6. Go to **TIB_CUSTOMER** and connect to the design-time adapter. Click **Save To Peoplesoft**.
7. Save as a multi-file project by the name **Customer**.
8. Select **Customer** in the project panel.
9. Click **Build Archive** in the **Configuration** tab. This creates an archive file.
10. Exit TIBCO Designer.

Deploying and Running the Example

Perform the following tasks to deploy and run the example.

Task A Deploy the Example

In TIBCO Administrator:

1. Click on the **Installed Software** link on the left panel. Check if TIBCO ActiveMatrix Adapter for PeopleSoft is registered. If the software is not registered, register by clicking the **Add Custom Software** button. Please refer to the TIBCO Administrator documentation for more details on adding custom software.
2. Go to **Application Management**. Click **New Application**. Upload the EAR file created in the previous section and click **OK**. On the following screen, clear the **Quick Configure** check box and then click **Save**.
3. Click **Configuration** in the newly created application. Click on the top level application name in the configuration view.
 - a. Go to the **Advanced** tab and provide the values for:
 - adpsft8.connection.appserver
 - adpsft8.connection.login
 - adpsft8.connection.password
 - adpsft8.connection.port
 - b. Click **Save**. The Configuration screen appears.
 - c. Expand the Enterprise archive link, so that adapter instances, `pubCI.aar`, `sub.aar` and `rpc.aar` are visible.
 - d. Click `pubCI.aar` and then click the **Add to Additional Machines** button.
 - e. Select the machine and click **OK**. In the following screen, click **Save**.
 - f. Repeat steps d and e for `sub.aar` and `rpc.aar`.
 - g. In the Configuration screen click **Deploy**, which will take you to the next screen. The **Start successfully deployed services** check box is selected by default. Click **OK**. The `.tra` files are created in `TIB_ADPSFT8_HOME/bin/domain/domain_name`.
4. Once the deployment is complete, click **Service Instances** under the application. The adapter is listed. Select the adapter and click **Load Selected**. Both the customer creation and customer fetching process is started. The adapter instance is also started.

Task B Run the Example

To run each example:

1. To trigger `GetProcess`, you must enter the key values in the `Get_Input.xml` file located in the `TIB_ADPSFT8_HOME/examples/BusinessWorks/Customer` folder. For example key values for the following parameters can be entered:

```
SETID
TIB_CUSTOMER
```

This fetches the customer details from PeopleSoft based on the details as given in the XML file. After the customer details are fetched from PeopleSoft, the `Get_Output.xml` file is written to the location `TIB_ADPSFT8_HOME/examples/BusinessWorks/Customer`

2. To trigger `UpdateProcess`, you must enter the data in the `input_render.xml` file located in the `TIB_ADPSFT8_HOME/examples/BusinessWorks/Customer` folder. For example:

```
<PROPERTIES>
  <SETID>CRM01</SETID>
  <PRODUCT_ID>10000</PRODUCT_ID>
  <PRODUCT_ID_O>10000</PRODUCT_ID_O>
  <DESCR>CRM01</DESCR>
  <TIB_PROD_PRICE__KIT_COMPS_PRICE>
  <OPRN_CODE>U</OPRN_CODE>
  <PROD_COMPONENT_ID>WE</PROD_COMPONENT_ID>
  <DATE_IN_EFFECT>U</DATE_IN_EFFECT>
  <PROD_KIT_COMP_PRICE>12.78</PROD_KIT_COMP_PRICE>
  <DATE_OBSOLUTE>12</DATE_OBSOLUTE>
  </TIB_PROD_PRICE__KIT_COMPS_PRICE>
  <TIB_PROD_PRICE__PROD_UOM>
  <OPRN_CODE>U</OPRN_CODE>
  <UNIT_OF_MEASURE>VINIBA</UNIT_OF_MEASURE>
  <TIB_PROD_PRICE__PROD_PRICE_BU>
  <OPRN_CODE>U</OPRN_CODE>
  <BUSINESS_UNIT_IN>QUOT</BUSINESS_UNIT_IN>
  <CURRENCY_CD>U</CURRENCY_CD>
  <TIB_PROD_PRICE__PROD_PRICE>
  <OPRN_CODE>U</OPRN_CODE>
  <EFFDT>U</EFFDT>
  <EFF_STATUS>QUOT</EFF_STATUS>
  <LIST_PRICE>1.1</LIST_PRICE>
  <UNIT_CAST>1.1</UNIT_CAST>
  <MSG_SUG_RTL_PRC>1.1</MSG_SUG_RTL_PRC>
  </TIB_PROD_PRICE__PROD_PRICE>
  </TIB_PROD_PRICE__PROD_PRICE_BU>
  </TIB_PROD_PRICE__PROD_UOM>
</PROPERTIES>
```

This creates a customer record in PeopleSoft.

3. To trigger `SyncProcess`, you must modify the data in PeopleSoft. For example you may modify any customer record. This triggers the `SyncProcess` and writes the data to the `Sync_output.xml` file.

Expected Results

The example results can be viewed in the respective XML files written to:
TIB_ADPSFT8_HOME/examples/BusinessWorks/Customer

- GetProcess data is fetched based on the SETID and TIB_CUSTOMER parameters specified in the Get_Input.xml file and is written into the Get_output.xml file. The Get_output.xml file should display the following customer record:

```
<Customers>
<PROPERTIES>
  <SETID>SHARE</SETID>
  <CUST_ID>1000</CUST_ID>
  <NAME1>Alliance & Group12</NAME1>
  <NAME_SHORT>Alliance</NAME_SHORT>
  <ADDRESS_SEQ_NO>1</ADDRESS_SEQ_NO>
  <TIB_CUSTOMER__CUST_ADDR_SEQ>
    <OPRN_CODE>U</OPRN_CODE>
    <SETID>SHARE</SETID>
    <CUST_ID>1000</CUST_ID>
    <ADDRESS_SEQ_NO>1</ADDRESS_SEQ_NO>
    <TIB_CUSTOMER__CUST_ADDRESS>
      <OPRN_CODE>U</OPRN_CODE>
      <EFF_DT>07/19/2002</EFF_DT>
      <EFF_STATUS>A</EFF_STATUS>
      <ADDRESS1>14410 Union Ave</ADDRESS1>
      <ADDRESS2></ADDRESS2>
      <CITY>San Jose</CITY>
      <STATE>CA</STATE>
      <POSTAL>95124</POSTAL>
    </TIB_CUSTOMER__CUST_ADDRESS>
  </TIB_CUSTOMER__CUST_ADDR_SEQ>
</PROPERTIES>
</Customers>
```

- UpdateProcess data is updated based on the SETID and TIB_CUSTOMER parameters specified in the input_render.xml file and is updated in PeopleSoft.
- SyncProcess output is stored in the Sync_output.xml file.

Chapter 3

TIBCO ActiveMatrix BusinessWorks: Working with the SalesOrder Component Interface

This example shows how to use several adapter services within a TIBCO ActiveMatrix BusinessWorks process to get, update, and synchronize a SalesOrder record stored in PeopleSoft.

Topics

- [Example Description, page 18](#)
- [Setting Up the Example, page 19](#)
- [Testing the Example, page 21](#)
- [Expected Results, page 22](#)

Example Description

This example shows how the adapter's Publication, Subscription and Request-Response services can be used in a TIBCO ActiveMatrix BusinessWorks process. The example has three processes, `GetProcess`, `UpdateProcess` and `SyncProcess`.

- `GetProcess` uses the adapter's Request-Response Service to fetch data from PeopleSoft for key values that you specify in an input XML file.

`GetProcess` queries PeopleSoft based on the given key values and fetches the data. The data is sent back by the adapter to TIBCO ActiveMatrix BusinessWorks and is written into an output XML file.

- `UpdateProcess` uses the adapter's Request-Response Service to update PeopleSoft.

In a typical business scenario, an adapter to another application (for example, SAP or Siebel) publishes a message into the TIBCO environment. TIBCO ActiveMatrix BusinessWorks receives the messages and writes them to an input XML file. The content of this XML file will contain the changes to be updated in PeopleSoft. TIBCO ActiveMatrix Adapter for PeopleSoft uses the Request-Response Service to update the data in PeopleSoft and sends a reply to TIBCO ActiveMatrix BusinessWorks.

Alternatively, you may specify key values in the input XML file. `UpdateProcess` queries PeopleSoft based on the key values. The SalesOrder details are updated in PeopleSoft.

- `SyncProcess` uses the adapter's Publication service to send a message out of PeopleSoft whenever SalesOrder component interface details are modified or updated in the target PeopleSoft application. Any change to the data in PeopleSoft triggers the `SyncProcess`.

On receiving the modified data, TIBCO ActiveMatrix BusinessWorks logs it to an XML file.

Setting Up the Example

Start the TIBCO EMS Server and PeopleSoft Design Time Adapter

This example uses JMS as the transport type. You need to start the TIBCO EMS Server and PeopleSoft Design Time Adapter:

1. Start the TIBCO EMS server.
2. Start the `adpsft8DTA_JMS` executable file under `TIB_ADPSFT8_HOME/bin`

Ensure that the TIBCO EMS Server is running and accessible to the machine on which the adapter is installed before running any of the processes.

Load the Project Archive in TIBCO Designer

To load the project archive:

1. Create a new project and import the project from the `SALESORDER.zip` file located in: `TIB_ADPSFT8_HOME/examples/BusinessWorks/SALESORDER`. See [Importing a Project from a ZIP Archive in TIBCO Designer on page 7](#) for details.
2. Substitute the values of the following global variables for design-time connection:
`adpsft8.connection.appserver`
`adpsft8.connection.login`
`adpsft8.connection.password`
`adpsft8.connection.port`
`adpsft8DTAJmsProviderURL`
3. Establish design-time connection:
 - a. In the Project panel, navigate to the **pub** service instance.
 - b. Click the **Connect** button in the Design-time connection tab.
 - c. If the connection is established, the DTA Connection Successful dialog is displayed. Click **OK**.

4. Get schema:
 - a. In the project panel, navigate to the **TIB_SALESORDER** service under the **pub** service instance.
 - b. Click the **Configuration** tab in the Configuration panel.
 - c. Enter **TIB_SALESORDER** in the Component Interface Name field.
 - d. Click **Fetch Interface** next to the field.
 - e. Click **OK** in the displayed Fetch Interface dialog.
 - f. Click **Get Schema**.
 - g. Publication Service only: Click **Save To PeopleSoft**.
5. Follow [step 3](#) and [step 4](#) for the **rpc** and **sub** service instances.
6. Save the project.

Testing the Example

In TIBCO Designer:

1. From the project panel, select the process you want to test, for example, `GetProcess`.
2. Click the **Set Breakpoints** icon.
3. In the window that appears, choose **Select All**, then click **OK**.
4. Click the **Tester** tab to the left of the Project panel. The Test panel replaces the project tree.
5. Click the **Start Testing Viewed Process** button.
6. In the process selection window that appears, the `GetProcess` is selected by default. Click **Load Selected**. The process is now test mode.
7. Click the **File Poller** icon. In the Configuration panel, File Name field, click **Browse** and select the input XML file. Click **Apply**. Any change to the input XML file starts the process.
8. Once the process starter is highlighted (indicating a process has started), click the **Step To Next Activity** icon to step through the process.
9. Step through the process once more.
10. Click the **Stop Testing** icon to return to design mode.

The above steps describe the testing for `GetProcess`. Follow the same steps to run the `SyncProcess` and `UpdateProcess`. The `SyncProcess` does not have an input XML file. Any change to the data in PeopleSoft will invoke the `SyncProcess`.

Expected Results

The example results can be viewed in the respective output XML files configured in [Setting Up the Example on page 19](#).

- **GetProcess:**

The **GetProcess** data is fetched based on the **SETID** and **TIB_SalesOrder** parameters specified in the **Get_Input.xml** file. In case of a success, it retrieves the employee details and generates the **Get_output.xml** file. In case of an error, it prints the error message in the **get_output_err.xml** file. Any change to the input XML file starts the process.

The following **SalesOrder** record is from an example **Get_output.xml** file:

```
<SalesOrder>
<PROPERTIES>
  <BUSINESS_UNIT>US001</BUSINESS_UNIT>
  <ORDER_NO>CEN0001</ORDER_NO>
  <DESCR_0></DESCR_0>
  <ADDRESS1></ADDRESS1>
  <COUNTRY_SHIP_FROM></COUNTRY_SHIP_FROM>
  <SOLD_TO_CUST_ID_0>US011</SOLD_TO_CUST_ID_0>
  <BILL_TO_CUST_ID_1>US011</BILL_TO_CUST_ID_1>
  <ORDENT_HDR_NAV>0000</ORDENT_HDR_NAV>
  <COMMSN_LEVEL>1</COMMSN_LEVEL>
  <COMMSN_METHOD>C</COMMSN_METHOD>
  <EXPORT>N</EXPORT>
  <ORDER_DATE>09/30/2002</ORDER_DATE>
  <ORDER_GRP>STD</ORDER_GRP>
  <ORDER_STATUS>O</ORDER_STATUS>
  <PROD_ID_SRC>S</PROD_ID_SRC>
  <TIB_SALESORDER>
    <OPRN_CODE>U</OPRN_CODE>
    <COPY_LINE_SEL></COPY_LINE_SEL>
    <ORDER_NO></ORDER_NO>
    <ORDER_DATE></ORDER_DATE>
    <DESCR></DESCR>
    <PROD_ID_ENTERED></PROD_ID_ENTERED>
    <QTY_ORDERED>0</QTY_ORDERED>
    <UNIT_OF_MEASURE></UNIT_OF_MEASURE>
  </TIB_SALESORDER>
</PROPERTIES>
</SalesOrder>
```

- **UpdateProcess** data is updated based on the **SETID** and **TIB_SalesOrder** parameters specified in the **Update_input.xml** file and is updated in **PeopleSoft**.
- **SyncProcess** output is stored in the **Sync_output.xml** file.

Chapter 4

TIBCO ActiveMatrix BusinessWorks: Working with the PurchaseOrder Component Interface

This example shows how to use several adapter services within a TIBCO ActiveMatrix BusinessWorks process to get, update, and synchronize a PurchaseOrder record stored in PeopleSoft.

Topics

- [Example Description, page 24](#)
- [Setting Up the Example, page 25](#)
- [Testing the Example, page 26](#)
- [Expected Results, page 27](#)

Example Description

This example shows how the adapter's Publication, Subscription and Request-Response services can be used in a TIBCO ActiveMatrix BusinessWorks process. The example has three processes, `GetProcess`, `UpdateProcess` and `SyncProcess`.

- `GetProcess` uses the adapter's Request-Response service to fetch data from PeopleSoft for key values that you specify in an input XML file.

`GetProcess` queries PeopleSoft based on the given key values and fetches the data. The data is sent back by the adapter to TIBCO ActiveMatrix BusinessWorks and is written into an output XML file.

- `UpdateProcess` uses the adapter's Request-Response service to update PeopleSoft.

In a typical business scenario, an adapter to another application (for example, SAP or Siebel) publishes a message into the TIBCO environment. TIBCO ActiveMatrix BusinessWorks receives the messages and writes them to an input XML file. The content of this XML file will contain the changes to be updated in PeopleSoft. TIBCO ActiveMatrix Adapter for PeopleSoft uses the Request-Response service to update the data in PeopleSoft and sends a reply to TIBCO ActiveMatrix BusinessWorks.

Alternatively, you may specify key values in the input XML file. `UpdateProcess` queries PeopleSoft based on the key values. The PurchaseOrder details are updated in PeopleSoft.

- `SyncProcess` uses the adapter's Publication service to send a message out of PeopleSoft whenever PurchaseOrder Component Interface details are modified or updated in the target PeopleSoft application. Any change to the data in PeopleSoft triggers the `SyncProcess`.

On receiving the modified data, TIBCO ActiveMatrix BusinessWorks logs it to an XML file.

Setting Up the Example

Before starting the example you must import the sample project zip file and save it in a new project.

In TIBCO Designer:

1. In the initial dialog box click **Open New Project** and specify a name for the project.
2. Click **Project>Import Full Project**.
3. Click the **Zip Archive** tab and browse to locate the `PurchaseOrder.zip` file. The file is located in the folder `<adapter_home>/examples/BusinessWorks/PurchaseOrder`
4. Substitute the values of the following global variables for design-time connection:

`adpsft8.connection.appserver`

`adpsft8.connection.login`

`adpsft8.connection.password`

`adpsft8.connection.port`

Or, you can specify these values in the `examples.tra`. In this case, the values given in project file will be overridden.

5. Go to the Publication service instance.
6. Go to **TIB_Purchase_Order** and connect to the design-time adapter. Click **Save To Peoplesoft**.
7. Save as a multi-file project by the name **PurchaseOrder** in a new folder and exit TIBCO Designer.

Start the TIBCO EMS Server:

This example uses JMS as the transport type. Ensure that the TIBCO EMS Server is running and accessible to the machine on which the adapter is installed before running any of the processes.

Testing the Example

In TIBCO Designer:

1. From the project panel, select the process you want to test. For example, `GetProcess`.
2. Click the **Set Breakpoints** icon.
3. In the window that appears, choose **Select All**, then click **OK**.
4. Click the **Tester** tab to the left of the project panel. The test panel replaces the project tree.
5. Click the **Start testing viewed process** button.
6. In the process selection window that appears, the `GetProcess` is selected by default. Click **Load Selected**. The process is now in test mode.
7. Click the **File Poller** icon. In the configuration panel, File Name field, click **Browse** and select the input XML file. Click **Apply**. Any change to the input XML file starts the process.
8. Once the process starter is highlighted (indicating a process has started), click the **Step to next activity** icon to step through the process.
9. Step through the process once more.
10. Click the **Stop Testing** icon to return to design mode.
11. The above steps describe the testing for `GetProcess`. Follow the same steps to run the `SyncProcess` and `UpdateProcess`. The `SyncProcess` does not have an input XML file. Any change to the data in PeopleSoft will invoke the `SyncProcess`.

Expected Results

The example results can be viewed in the respective XML files written to:

TIB_ADPSFT8_HOME/examples/BusinessWorks/PurchaseOrder

- GetProcess data is fetched based on the SETID and TIB_Purchase_Order parameters specified in the Get_Input.xml file and is written into the Get_output.xml file. The Get_output.xml file should display the following PurchaseOrder record:

```
<PurchaseOrders>
<PROPERTIES>
  <BUSINESS_UNIT_2>US001</BUSINESS_UNIT_2>
  <PO_ID_2>0000000001</PO_ID_2>
  <CHNG_ORD_BATCH>0</CHNG_ORD_BATCH>
  <PO_TYPE>GEN</PO_TYPE>
  <HOLD_STATUS>N</HOLD_STATUS>
  <DISP_ACTION>Y</DISP_ACTION>
  <DISP_METHOD>PRN</DISP_METHOD>
  <PO_DT>08/10/2000</PO_DT>
  <TIB_PURCHASE_ORDER_PO_LINE>
    <OPRN_CODE>U</OPRN_CODE>
    <UNIT_OF_MEASURE>EA</UNIT_OF_MEASURE>
    <QTY_TYPE>L</QTY_TYPE>
    <PRICE_DT_TYPE>P</PRICE_DT_TYPE>
    <MFG_ID></MFG_ID>
    <CURRENCY_CD>USD</CURRENCY_CD>
    <CURRENCY_CD_BASE>USD</CURRENCY_CD_BASE>
    <DELETING_LINE></DELETING_LINE>
    <TIB_PURCHASE_ORDER_PO_LINE_MISC>
      <OPRN_CODE>U</OPRN_CODE>
      <MISC_CHARGE_CODE></MISC_CHARGE_CODE>
      <MISC_CHARGE_ORIGN></MISC_CHARGE_ORIGN>
      <CHARGE_AMOUNT>0</CHARGE_AMOUNT>
      <CURRENCY_CD></CURRENCY_CD>
      <LC_COMP_CALC></LC_COMP_CALC>
      <UNIT_PRICE>0</UNIT_PRICE>
      <FLAT_AMOUNT>0</FLAT_AMOUNT>
      <PCT_VALUE>0</PCT_VALUE>
      <UNIT_OF_MEASURE></UNIT_OF_MEASURE>
    </TIB_PURCHASE_ORDER_PO_LINE_MISC>
  </TIB_PURCHASE_ORDER_PO_LINE>
</PROPERTIES>
</PurchaseOrders>
```

- UpdateProcess data is updated based on the SETID and TIB_Purchase_Order parameters specified in the input_render.xml file and is updated in PeopleSoft.
- SyncProcess output is stored in the Sync_output.xml file.

Chapter 5

TIBCO ActiveMatrix BusinessWorks: Working with the Employee Component Interface

This example shows how to use several adapter services within a TIBCO BusinessWorks process to get, update, and synchronize an Employee record stored in PeopleSoft.

Topics

- [Example Description, page 30](#)
- [Setting Up the Example, page 31](#)
- [Testing the Example, page 33](#)
- [Expected Results, page 37](#)

Example Description

This example shows how the adapter's Publication, Subscription and Request-Response services can be used in a TIBCO ActiveMatrix BusinessWorks process. The example has three processes, `GetProcess`, `UpdateProcess` and `SyncProcess`.

- `GetProcess` uses the adapter's Request-Response Service to fetch data from PeopleSoft for key values that you specify in an input XML file.

`GetProcess` queries PeopleSoft based on the given key values and fetches the data. The data is sent back by the adapter to TIBCO ActiveMatrix BusinessWorks and is written into an output XML file.

- `UpdateProcess` uses the adapter's Publication and Subscription services to update PeopleSoft.

In a typical business scenario, an adapter publishes a message to another application (for example, SAP or Siebel) into the TIBCO environment. TIBCO ActiveMatrix BusinessWorks receives the messages and writes them to an input XML file. The content of this XML file will contain the changes to be updated in PeopleSoft. TIBCO ActiveMatrix Adapter for PeopleSoft uses the Subscription service to update the data in PeopleSoft.

In the Employee example provided, the publisher triggers the `UpdateProcess`. The data to be updated is specified in the input XML file. `UpdateProcess` queries PeopleSoft based on the key values specified in the input XML file and the Employee details are updated in PeopleSoft.

- `SyncProcess` uses the adapter's Publication Service to send a message out of PeopleSoft whenever Employee Component Interface details are modified or updated in the target PeopleSoft application. Any change to the data in PeopleSoft triggers the `SyncProcess`.

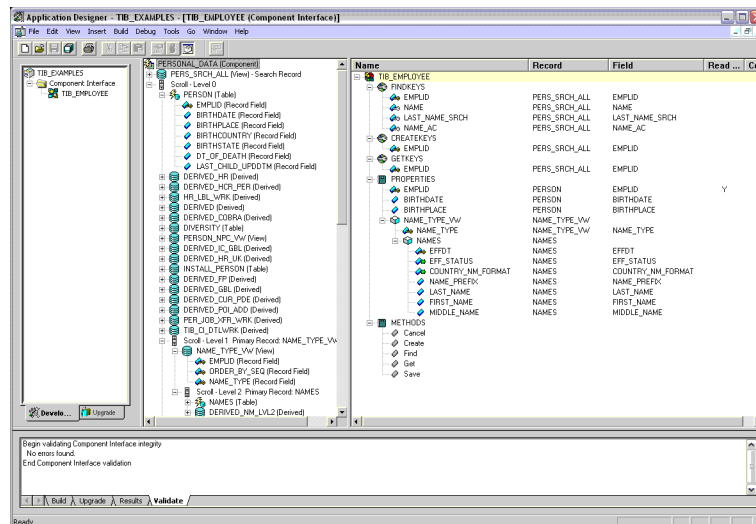
On receiving the modified data, TIBCO ActiveMatrix BusinessWorks logs it to an XML file.

Setting Up the Example

Create TIB_EMPLOYEE Component Interface

To create the component interface:

1. Log in to PeopleSoft Application Designer.
2. Open the TIB_EXAMPLES project.
3. Select **File > New > Component Interface** from the menu.
4. In the Select Source Component For Component Interface dialog, enter PERSONAL_DATA in the Name field and click **Select**.
5. Double-click on the PERSONAL_DATA component.
6. The next dialog box asks "*Do you want to default the properties based on the underlying Component definition, PERSONAL_DATA?*" Click **Yes**.
7. Delete unwanted fields and records from the component interface definition as shown below and save it as TIB_EMPLOYEE.



8. Insert the `TIB_EMPLOYEE` component interface into the project `TIB_EXAMPLES` by pressing F7. Save the project.

After creating the TIB_EMPLOYEE component interface, follow the procedures described in "Preparing PeopleSoft Component Interfaces" in *TIBCO ActiveMatrix Adapter for PeopleSoft Configuration and Deployment* to prepare the component interface correctly.

Start PeopleSoft Design Time Adapter

This example uses Rendezvous as the transport type. You need to start PeopleSoft Design Time Adapter by running the `adpsft8DTA_RV` executable file under `TIB_ADPSFT8_HOME/bin`.

Load the Project Archive in TIBCO Designer

To load the project archive:

1. Create a new project and import the project from the `EMPLOYEE.zip` file located in: `TIB_ADPSFT8_HOME/examples/BusinessWorks/EMPLOYEE`. See [Importing a Project from a ZIP Archive in TIBCO Designer on page 7](#) for details.
2. Substitute the values of the following global variables for design time connection for the pub, rpc, and sub services.
`adpsft8.connection.appserver`
`adpsft8.connection.login`
`adpsft8.connection.password`
`adpsft8.connection.port`
3. Ensure that the global variable `adpsft8DTARvService` is blank.

Testing the Example

Perform the following tasks to test the example.

Task A Test GetProcess

Configure the Process

To test the example for `GetProcess`, you need to log in to TIBCO Designer and configure `GetProcess` first by following these steps:

1. From the Project panel, select **GetProcess**.
2. Configure input:
 - a. Click the **File Poller** icon.
 - b. Click the **Configuration** tab and specify or browse to the location of the input XML file `get_input.xml`.

The default location is:

TIB_ADPSFT8_HOME\examples\businessworks\EMPLOYEE\Get_input.xml.

This XML file conforms to:

TIB_ADPSFT8_HOME\examples\businessworks\EMPLOYEE\Get_input_schema.xsd.

- c. Specify the desired EMPID value (for example, FA0001) in the `Get_input.xml` file.
3. Configure output that is generated in case of a success:
 - a. Click the **Write File** icon.
 - b. Click the **Input** tab.
 - c. In the Activity Input panel, expand **WriteActivityInputTextClass**.
 - d. In the `fileName` field, specify the entire path or location of the `Get_output.xml` file.
4. Configure error output that is generated in case of an error:
 - a. Click the **Write-File-1** icon.
 - b. Click the **Input** tab.
 - c. In the Activity Input panel, expand the **WriteActivityInputTextClass**. In the `fileName` field, specify the entire path or location of the `get_output_err.xml`.
5. Set break points for the process:
 - a. Click the **Set Breakpoints** icon from the menu bar.
 - b. In the displayed dialog, click the **Select All** button, then click **OK**.

6. Save the project.

Run the Test

After you make all the above configurations, follow these steps to test `GetProcess` in TIBCO Designer:

1. Select **Tools > Show Adapter Tester** from the menu and start the `rpc` adapter instance in the displayed dialog.
2. Click the **Tester** tab to the left of the Project panel. The test panel replaces the project tree.
3. Click the **Start Testing Viewed Process** button.
4. In the Select Processes To Load dialog, select `GetProcess` and click **Load Selected**. The process is now in test mode.
5. Once the process starter is highlighted (indicating a process has started), click the **Step To Next Activity** icon to step through the process.
6. Step through the process once more.
7. Click the **Stop Testing** icon to return to design mode.
8. Stop the `rpc` adapter instance in the adapter tester.

Task B Test SyncProcess

Configure the Process

To test the example for `SyncProcess`, you need to log in to TIBCO Designer and configure `SyncProcess` first by following these steps:

1. From the Project panel, select `SyncProcess`.
2. Configure output that is generated in case of a success:
 - a. Click the **Write File** icon.
 - b. Click the **Input** tab.
 - c. In the Activity Input panel, expand `WriteActivityInputTextClass`.
 - d. In the `fileName` field, specify the entire path or location of the `Sync_output.xml` file.
3. Set break points for the process:
 - a. Click the **Set Breakpoints** icon from the menu bar.
 - b. In the displayed dialog, click the **Select All** button, then click **OK**.
4. Save the project.

Run the Test

After you make all the above configurations, follow these steps to test `SyncProcess` in TIBCO Designer:

1. Select **Tools > Show Adapter Tester** from the menu and start the `pub` adapter instance in the displayed dialog.
2. Click the **Tester** tab to the left of the Project panel. The test panel replaces the project tree.
3. Click the **Start Testing Viewed Process** button.
4. In the Select Processes To Load dialog, select `SyncProcess` and click **Load Selected**. The process is now in test mode.
5. Once the process starter is highlighted (indicating a process has started), click the **Step To Next Activity** icon to step through the process.
6. Step through the process once more.
7. Click the **Stop Testing** icon to return to design mode.
8. Stop the `pub` adapter instance in the adapter tester.

Task C Test UpdateProcess

Configure the Process

To test the example for `UpdateProcess`, you need to log in to TIBCO Designer and configure `UpdateProcess` first by following these steps:

1. From the Project panel, select `UpdateProcess`.
2. Configure input:
 - a. Click the **Read File** icon.
 - b. Click the **Input** tab and specify or browse to the location of the input XML file `get_input.xml`.
 - c. In the Activity Input panel, expand **ReadActivityInputClass**.
 - d. In the `fileName` field, specify the entire path or location of the `Update_input.xml` file. This XML file conforms to `TIB_ADPSFT8_HOME\examples\businessworks\EMPLOYEE\Update_input_schema.xsd`.
3. Set break points for the process:
 - a. Click the **Set Breakpoints** icon from the menu bar.
 - b. In the displayed dialog, click the **Select All** button, then click **OK**.
4. Save the project.

Run the Test

After you make all the above configurations, follow these steps to test `UpdateProcess` in TIBCO Designer:

1. Select **Tools > Show Adapter Tester** from the menu and start the **pub** and **sub** adapter instances in the displayed dialog.
2. Click the **Tester** tab to the left of the Project panel. The test panel replaces the project tree.
3. Click the **Start Testing Viewed Process** button.
4. In the Select Processes To Load dialog, select **UpdateProcess** and click **Load Selected**. The process is now in test mode.
5. Once the process starter is highlighted (indicating a process has started), click the **Step To Next Activity** icon to step through the process.
6. Step through the process once more.
7. Click the **Stop Testing** icon to return to design mode.
8. Stop the **pub** and **sub** adapter instance in the adapter tester.

Expected Results

The example results can be viewed in the respective output XML files configured in [Setting Up the Example on page 31](#).

- **GetProcess:**

The `GetProcess` data is fetched based on the `EMPLID` parameter specified in the `Get_Input.xml` file. In case of a success, it retrieves the employee details and generates the `Get_output.xml` file. In case of an error, it prints the error message in the `get_output_err.xml` file. Any change to the input XML file starts the process.

The following `EMPLOYEE` record is from an example `Get_output.xml` file:

```
<?xml version="1.0" encoding=""?>
<EMPLOYEE>
  <EMPLID>FA0001</EMPLID>
  <BIRTHDATE>08/06/1982</BIRTHDATE>
  <BIRTHPLACE>New York City</BIRTHPLACE>
  <TIB_EMPLOYEE__NAME_TYPE_VW>
    <NAME_TYPE>PRI</NAME_TYPE>
    <TIB_EMPLOYEE__NAMES>
      <EFFDT>09/23/1999</EFFDT>
      <EFF_STATUS>A</EFF_STATUS>
      <COUNTRY_NM_FORMAT>001</COUNTRY_NM_FORMAT>
      <NAME_PREFIX>Mr</NAME_PREFIX>
      <LAST_NAME>WOODROW</LAST_NAME>
      <FIRST_NAME>AARON</FIRST_NAME>
      <MIDDLE_NAME/>
    </TIB_EMPLOYEE__NAMES>
  </TIB_EMPLOYEE__NAME_TYPE_VW>
  <TIB_EMPLOYEE__NAME_TYPE_VW>
    <NAME_TYPE>PRF</NAME_TYPE>
    <TIB_EMPLOYEE__NAMES>
      <EFFDT>09/23/1999</EFFDT>
      <EFF_STATUS>A</EFF_STATUS>
      <COUNTRY_NM_FORMAT>001</COUNTRY_NM_FORMAT>
      <NAME_PREFIX/>
      <LAST_NAME/>
      <FIRST_NAME/>
      <MIDDLE_NAME/>
    </TIB_EMPLOYEE__NAMES>
  </TIB_EMPLOYEE__NAME_TYPE_VW>
</EMPLOYEE>
```

- **SyncProcess**

The `SyncProcess` does not have an input XML file. Any change to the employee data in PeopleSoft will invoke the `SyncProcess`. `Sync_output.xml` will have the corresponding message published to the adapter.

The following EMPLOYEE record is from an example Sync_output.xml file:

```
<?xml version="1.0" encoding=""?>
<EMPLOYEE>
  <EMPLID>FA0001</EMPLID>
  <BIRTHDATE>1982-08-06</BIRTHDATE>
  <BIRTHPLACE>N Y City</BIRTHPLACE>
  <TIB_EMPLOYEE__NAME_TYPE_VW>
    <NAME_TYPE>PRI</NAME_TYPE>
    <TIB_EMPLOYEE__NAMES>
      <EFFDT>1999-09-23</EFFDT>
      <EFF_STATUS>A</EFF_STATUS>
      <COUNTRY_NM_FORMAT>001</COUNTRY_NM_FORMAT>
      <NAME_PREFIX/>
      <LAST_NAME>WOODROW</LAST_NAME>
      <FIRST_NAME>AARON</FIRST_NAME>
      <MIDDLE_NAME/>
    </TIB_EMPLOYEE__NAMES>
  </TIB_EMPLOYEE__NAME_TYPE_VW>
  <TIB_EMPLOYEE__NAME_TYPE_VW>
    <NAME_TYPE>PRF</NAME_TYPE>
    <TIB_EMPLOYEE__NAMES>
      <EFFDT>1999-09-23</EFFDT>
      <EFF_STATUS>A</EFF_STATUS>
      <COUNTRY_NM_FORMAT>001</COUNTRY_NM_FORMAT>
      <NAME_PREFIX/>
      <LAST_NAME/>
      <FIRST_NAME/>
      <MIDDLE_NAME/>
    </TIB_EMPLOYEE__NAMES>
  </TIB_EMPLOYEE__NAME_TYPE_VW>
</EMPLOYEE>
```

- UpdateProcess

The adapter tester picks up the EMPID parameter from the Update_input.xml file and updates the corresponding data in PeopleSoft.

Chapter 6

TIBCO ActiveMatrix BusinessWorks: Working with the ItemMaster Component Interface

This example shows how to use several adapter services within a TIBCO ActiveMatrix BusinessWorks process to get, update, and synchronize an ItemMaster record stored in PeopleSoft.

Topics

- [Example Description, page 40](#)
- [Setting Up the Example, page 41](#)
- [Testing the Example, page 43](#)
- [Expected Results, page 47](#)

Example Description

This example shows how the adapter's Publication, Subscription and Request-Response services can be used in a TIBCO ActiveMatrix BusinessWorks process. The example has three processes, `GetProcess`, `UpdateProcess` and `SyncProcess`.

- `GetProcess` uses the adapter's Request-Response service to fetch data from PeopleSoft for key values that you specify in an input XML file.

`GetProcess` queries PeopleSoft based on the given key values and fetches the data. The data is sent back by the adapter to TIBCO ActiveMatrix BusinessWorks and is written into an output XML file.

- `UpdateProcess` uses the adapter's Publication and Subscription services to update PeopleSoft.

In a typical business scenario, an adapter publishes a message to another application (for example, SAP or Siebel) into the TIBCO environment. TIBCO ActiveMatrix BusinessWorks receives the messages and writes them to an input XML file. The content of this XML file will contain the changes to be updated in PeopleSoft. TIBCO ActiveMatrix Adapter for PeopleSoft uses the Subscription service to update the data in PeopleSoft.

In the ItemMaster example provided, the publisher triggers the `UpdateProcess`. The data to be updated is specified in the input XML file. `UpdateProcess` queries PeopleSoft based on the key values specified in the input XML file and the ItemMaster details are updated in PeopleSoft.

- `SyncProcess` uses the adapter's Publication service to send a message out of PeopleSoft whenever ItemMaster Component Interface details are modified or updated in the target PeopleSoft application. Any change to the data in PeopleSoft triggers the `SyncProcess`.

On receiving the modified data, TIBCO ActiveMatrix BusinessWorks logs it to an XML file.

Setting Up the Example

Prepare the TIB_INV_PROD_FAMILIES Component Interface

Follow the procedures described in "Preparing PeopleSoft Component Interfaces" in *TIBCO ActiveMatrix Adapter for PeopleSoft Configuration and Deployment* to prepare the component interface correctly.

Start PeopleSoft Design Time Adapter

This example uses Rendezvous as the transport type. You need to start PeopleSoft Design Time Adapter by running the `adpsft8DTA_RV` executable file under `TIB_ADPSFT8_HOME/bin`.

Load the Project Archive in TIBCO Designer

To load the project archive:

1. Create a new project and import the project from the `ITEMMASTER.zip` file located in: `TIB_ADPSFT8_HOME/examples/BusinessWorks/ITEMMASTER`. See [Importing a Project from a ZIP Archive in TIBCO Designer on page 7](#) for details.
2. Substitute the values of the following global variables for design-time connection:
 - `adpsft8.connection.appserver`
 - `adpsft8.connection.login`
 - `adpsft8.connection.password`
 - `adpsft8.connection.port`

Ensure that the global variable `adpsft8DTARvService` is blank.
3. Establish design-time connection:
 - a. In the Project panel, navigate to the **pub** service instance.
 - b. Click the **Connect** button in the Design-time connection tab.
 - c. If the connection is established, the DTA Connection Successful dialog is displayed. Click **OK**.

4. Get schema:
 - a. In the project panel, navigate to the **TIB_INV_PROD_FAMILIES** service under the **pub** service instance.
 - b. Click the **Configuration** tab in the Configuration panel.
 - c. Enter **TIB_INV_PROD_FAMILIES** in the Component Interface Name field.
 - d. Click **Fetch Interface** next to the field.
 - e. Click **OK** in the displayed Fetch Interface dialog.
 - f. Click **Get Schema**.
 - g. Publication Service only: Click **Save To PeopleSoft**.
5. Follow [step 3](#) and [step 4](#) for the **rpc** and **sub** service instances.
6. Save the project.

Testing the Example

Perform the following task to test the example.

Task A Test GetProcess

Configure the Process

To test the example for GetProcess, you need to log in to TIBCO Designer and configure GetProcess first by following these steps:

1. From the Project panel, select **GetProcess**.
2. Configure input:
 - a. Click the **File Poller** icon.
 - b. Click the **Configuration** tab and specify or browse to the location of the input XML file `Get_input.xml`.

 The default location is:
TIB_ADPSFT8_HOME\examples\businessworks\ITEMMASTER\Get_input.xml.

 This XML file conforms to:
TIB_ADPSFT8_HOME\examples\businessworks\ITEMMASTER\Get_input_schema.xsd.
 - c. Specify the desired SETID and INV_PROD_FAM_CD values (for example, CRM01 and COMPACTOR) in the `Get_input.xml` file.
3. Configure output that is generated in case of a success:
 - a. Click the **Write File** icon.
 - b. Click the **Input** tab.
 - c. In the Activity Input panel, expand **WriteActivityInputTextClass**.
 - d. In the `fileName` field, specify the entire path or location of the `Get_output.xml` file.
4. Configure error output that is generated in case of an error:
 - a. Click the **Write-File-1** icon.
 - b. Click the **Input** tab.
 - c. In the Activity Input panel, expand the **WriteActivityInputTextClass**. In the `fileName` field, specify the entire path or location of the `get_output_err.xml`.

5. Set break points for the process:
 - a. Click the **Set Breakpoints** icon from the menu bar.
 - b. In the displayed dialog, click the **Select All** button, then click **OK**.
6. Save the project.

Run the Test

After you make all the above configurations, follow these steps to test `GetProcess` in TIBCO Designer:

1. Select **Tools > Show Adapter Tester** from the menu and start the `rpc` adapter instance in the displayed dialog.
2. Click the **Tester** tab to the left of the Project panel. The test panel replaces the project tree.
3. Click the **Start Testing Viewed Process** button.
4. In the Select Processes To Load dialog, select `GetProcess` and click **Load Selected**. The process is now in test mode.
5. Once the process starter is highlighted (indicating a process has started), click the **Step To Next Activity** icon to step through the process.
6. Step through the process once more.
7. Click the **Stop Testing** icon to return to design mode.
8. Stop the `rpc` adapter instance in the adapter tester.

Task B Test SyncProcess

Configure the Process

To test the example for `SyncProcess`, you need to log in to TIBCO Designer and configure `SyncProcess` first by following these steps:

1. From the Project panel, select `SyncProcess`.
2. Configure output that is generated in case of a success:
 - a. Click the **Write File** icon.
 - b. Click the **Input** tab.
 - c. In the Activity Input panel, expand **WriteActivityInputTextClass**.
 - d. In the `fileName` field, specify the entire path or location of the `Sync_output.xml` file.

3. Set break points for the process:
 - a. Click the **Set Breakpoints** icon from the menu bar.
 - b. In the displayed dialog, click the **Select All** button, then click **OK**.
4. Save the project.

Run the Test

After you make all the above configurations, follow these steps to test `SyncProcess` in TIBCO Designer:

1. Select **Tools > Show Adapter Tester** from the menu and start the `pub` adapter instance in the displayed dialog.
2. Click the **Tester** tab to the left of the Project panel. The test panel replaces the project tree.
3. Click the **Start Testing Viewed Process** button.
4. In the Select Processes To Load dialog, select `SyncProcess` and click **Load Selected**. The process is now in test mode.
5. Once the process starter is highlighted (indicating a process has started), click the **Step To Next Activity** icon to step through the process.
6. Step through the process once more.
7. Click the **Stop Testing** icon to return to design mode.
8. Stop the `pub` adapter instance in the adapter tester.

Task C Test `UpdateProcess`

Configure the Process

To test the example for `UpdateProcess`, you need to log in to TIBCO Designer and configure `UpdateProcess` first by following these steps:

1. From the Project panel, select `UpdateProcess`.

2. Configure input:
 - a. Click the **Read File** icon.
 - b. Click the **Input** tab and specify or browse to the location of the input XML file `get_input.xml`.
 - c. In the Activity Input panel, expand **ReadActivityInputClass**.
 - d. In the `fileName` field, specify the entire path or location of the `Update_input.xml` file. This XML file conforms to `TIB_ADPSFT8_HOME\examples\businessworks\ITEMMASTER\Update_input_schema.xsd`.
3. Set break points for the process:
 - a. Click the **Set Breakpoints** icon from the menu bar.
 - b. In the displayed dialog, click the **Select All** button, then click **OK**.
4. Save the project.

Run the Test

After you make all the above configurations, follow these steps to test `UpdateProcess` in TIBCO Designer:

1. Select **Tools > Show Adapter Tester** from the menu and start the **pub** and **sub** adapter instances in the displayed dialog.
2. Click the **Tester** tab to the left of the Project panel. The test panel replaces the project tree.
3. Click the **Start Testing Viewed Process** button.
4. In the Select Processes To Load dialog, select **UpdateProcess** and click **Load Selected**. The process is now in test mode.
5. Once the process starter is highlighted (indicating a process has started), click the **Step To Next Activity** icon to step through the process.
6. Step through the process once more.
7. Click the **Stop Testing** icon to return to design mode.
8. Stop the **pub** and **sub** adapter instance in the adapter tester.

Expected Results

The example results can be viewed in the respective output XML files configured in [Setting Up the Example on page 41](#).

- GetProcess:

The GetProcess data is fetched based on the SETID and INV_PROD_FAM_CD parameters specified in the Get_input.xml file. In case of a success, it retrieves the employee details and generates the Get_output.xml file. In case of an error, it prints the error message in the get_output_err.xml file. Any change to the input XML file starts the process.

The following TIB_INV_PROD_FAMILIES record is from an example Get_output.xml file:

```
<?xml version="1.0" encoding="UTF-8"?>
<MapperOutput>
  <PROPERTIES>
    <SETID>CRM01</SETID>
    <INV_PROD_FAM_CD>COMPACTOR</INV_PROD_FAM_CD>
    <TIB_INV_PROD_FAMILIES_caret_INV_ITEM_FAM>
      <item>
        <OPRN_CODE>U</OPRN_CODE>
        <EFFDT>01/01/2000</EFFDT>
        <EFF_STATUS>A</EFF_STATUS>
        <DESCR>With Compactor 11</DESCR>
        <DESCRSHORT>Compactor</DESCRSHORT>
      </item>
    </TIB_INV_PROD_FAMILIES_caret_INV_ITEM_FAM>
  </PROPERTIES>
</MapperOutput>
```

- UpdateProcess

The adapter tester picks up the SETID and INV_PROD_FAM_CD parameters from the Update_input.xml file and updates the corresponding data in PeopleSoft.

- SyncProcess

The SyncProcess does not have an input XML file. Any change to the employee data in PeopleSoft will invoke the SyncProcess. Sync_output.xml will have the corresponding message published to the adapter.

The following TIB_INV_PROD_FAMILIES record is from an example Sync_output.xml file:

```
<?xml version="1.0" encoding="UTF-8"?>
<ns0:TIB_INV_PROD_FAMILIES_PROPERTIES
  xmlns:ns0="http://www.tibco.com/xmlns/ae2xsd/2002/05/ae/
  PeopleSoft/pub/businessObjects">
  <SETID>CRM01</SETID>
  <INV_PROD_FAM_CD>COMPACTOR</INV_PROD_FAM_CD>
```

```
<TIB__INV__PROD__FAMILIES_caret_INV__ITEM__FAM>
  <item>
    <OPRN__CODE>U</OPRN__CODE>
    <EFFDT>2000-01-01</EFFDT>
    <EFF__STATUS>A</EFF__STATUS>
    <DESCR>With Compactor 12</DESCR>
    <DESCRSHORT>Compactor</DESCRSHORT>
  </item>
</TIB__INV__PROD__FAMILIES_caret_INV__ITEM__FAM>
</ns0:TIB__INV__PROD__FAMILIES__PROPERTIES>
```

Chapter 7

TIBCO ActiveMatrix BusinessWorks: Working with the UOM Message

This example shows how to use the Message Publication Service within a TIBCO ActiveMatrix BusinessWorks process to trace changes in the PeopleSoft environment and publish them to the TIBCO environment. The message is transferred to the PeopleSoft environment using Integration Broker.

Topics

- [Example Description, page 50](#)
- [Setting Up the Example, page 51](#)
- [Preparing PeopleSoft Integration Broker, page 53](#)
- [Testing the Example, page 55](#)
- [Expected Results, page 56](#)

Example Description

This example shows how the adapter's Message Publication Service can be used in a TIBCO ActiveMatrix BusinessWorks process. The example has one process, `ProcessMessagePublication`, that publishes the `UOM_SYNC` message whenever `UnitOfMeasure` details are modified or updated in the PeopleSoft application.

Setting Up the Example

Start the TIBCO EMS Server and PeopleSoft Design Time Adapter

This example uses JMS as the transport type. You need to start the TIBCO EMS Server and PeopleSoft Design Time Adapter:

1. Start the TIBCO EMS server.
2. Start the `adpsft8DTA_JMS` executable file under `TIB_ADPSFT8_HOME/bin`

Ensure that the TIBCO EMS Server is running and accessible to the machine on which the adapter is installed before running any of the processes.

Load the Project Archive in TIBCO Designer

To load the project archive:

1. Create a new project and import the project from the `UOM.zip` file located in: `TIB_ADPSFT8_HOME/examples/IntegrationBroker/UOM`. See [Importing a Project from a ZIP Archive in TIBCO Designer on page 7](#) for details.
2. Substitute the values of the following global variables for design-time connection:
`adpsft8.connection.appserver`
`adpsft8.connection.login`
`adpsft8.connection.password`
`adpsft8.connection.port`
`adpsft8DTAJmsProviderURL`
3. Establish design-time connection:
 - a. In the Project panel, navigate to the **MessagePublisher** service instance.
 - b. Click the **Connect** button in the Design-time connection tab.
 - c. If the connection is established, the DTA Connection Successful dialog is displayed. Click **OK**.

4. Get schema:
 - a. In the project panel, navigate to the **UOM_SYNC_VERSION_1** service under the **MessagePublisher** service instance.
 - b. Click the **Configuration** tab in the Configuration panel.
 - c. Enter **UOM_SYNC.VERSION_1** in the Message Name field.
 - d. Click **Fetch Message** next to the field.
 - e. Click **OK** in the displayed Fetch Message dialog.
 - f. Click **Get Schema**.
5. Save the project.
6. Exit TIBCO Designer.

Preparing PeopleSoft Integration Broker

After setting up the example, you need to create an integration scenario between the adapter and PeopleSoft Integration Broker. For detailed procedures, see "Preparing PeopleSoft Application Messages" in *TIBCO ActiveMatrix Adapter for PeopleSoft Configuration and Deployment*.

In particular, follow these steps when you make the configuration:

1. When you modify the `integrationGateway.Properties` file, set the queue for the test as:

```
ig.jms.Queue#=TEST_PSFT_PUB_UOM_SYNC
```

Remember to create a queue with the same name in the TIBCO EMS server.

2. When you define a Gateway and Node, specify the following connector properties:

— **JMSQueue:** TEST_PSFT_PUB_UOM_SYNC

— **serviceType:** PUBLISHER

3. When you define a Service Operation, select **Asynchronous - One Way** as the Operation Type of the Service Operation.
4. In PeopleTools Application Designer, update the PeopleCode of the `SavePostChange` method with the following code:

```
Local Message &msg;  
Local Message &msg1;  
Local Rowset &rowSet;  
Local XmlDocument &xmlDoc;  
Local string &xmlString;  
Local XmlNode &rootNode;  
Local XmlNode &actionNode;  
&rowSet = GetLevel0();  
&msg = CreateMessage(Message.UOM_SYNC);  
&msg1 = CreateMessage(Message.UOM_SYNC);  
&msg.CopyRowset(&rowSet);  
&xmlString = &msg.GenXMLString();  
&xmlDoc = CreateXmlDoc(&xmlString);  
&rootNode = &xmlDoc.DocumentElement;  
&actionNode = &rootNode.FindNode("MsgData/Transaction/PSCAMA/AUDIT_ACTN");  
&actionNode.NodeValue = %Mode;  
&xmlString = &xmlDoc.GenXmlString();  
&msg1.LoadXMLString(&xmlString);  
&msg1.Publish();
```

Testing the Example

Configure the Process

To test the example, you need to log in to TIBCO Designer and configure the `ProcessMessagePublication` process first by following these steps:

1. From the Project panel, select **ProcessMessagePublication**.
2. Configure output that is generated in case of a success:
 - a. Click the **Write File** icon.
 - b. Click the **Input** tab.
 - c. In the Activity Input panel, expand **WriteActivityInputTextClass**.
 - d. In the `fileName` field, specify the entire path or location of the `pub_output.xml` file.
3. Set break points for the process:
 - a. Click the **Set Breakpoints** icon from the menu bar.
 - b. In the displayed dialog, click the **Select All** button, then click **OK**.
4. Save the project.

Run the Test

After you make all the above configurations, follow these steps to test `SyncProcess` in TIBCO Designer:

1. Select **Tools > Show Adapter Tester** from the menu and start the **MessagePublisher** adapter instance in the displayed dialog.
2. Click the **Tester** tab to the left of the Project panel. The test panel replaces the project tree.
3. Click the **Start Testing Viewed Process** button.
4. In the Select Processes To Load dialog, select **ProcessMessagePublication** and click **Load Selected**. The process is now in test mode.
5. Once the process starter is highlighted (indicating a process has started), click the **Step To Next Activity** icon to step through the process.
6. Step through the process once more.
7. Click the **Stop Testing** icon to return to design mode.
8. Stop the **MessagePublisher** adapter instance in the adapter tester.

Expected Results

The example results can be viewed in the respective output XML files configured in [Setting Up the Example on page 51](#).

ProcessMessagePublisher looks for changes on the UOM_SYNC_VERSION1 message. Based on the message, the process generates the pub_output.xml file that is published to the TIBCO environment.

The following UOM record is from an example put_output.xml file:

```
<?xml version="1.0" encoding="UTF-8" ?>
-<ns0:uoms xmlns:ns0="http://xmlns.example.com/unique/default/namespace/ 1158039121783">
  -<ns0:uom>
    <ns0:unit_of_measure>AVC </ns0:unit_of_measure>
    <ns0:Desc>Average Cost Dollars </ns0:Desc>
    <ns0:DescShort>AvgCost </ns0:DescShort>
  </ns0:uom>
</ns0:uoms>
```

Chapter 8

TIBCO ActiveMatrix BusinessWorks: Working with the Contact Message

This example shows how to use the Message Subscription Service within a TIBCO ActiveMatrix BusinessWorks process to get a request from the TIBCO environment, process it, and send it to PeopleSoft. The message is transferred to the PeopleSoft environment using Integration Broker.

Topics

- [Example Description, page 58](#)
- [Setting Up the Example, page 59](#)
- [Preparing PeopleSoft Integration Broker, page 61](#)
- [Testing the Example, page 62](#)
- [Expected Results, page 63](#)

Example Description

This example shows how the adapter's Message Subscription Service can be used in a TIBCO ActiveMatrix BusinessWorks process. The example has one process, `ProcessMessageSubscriber`, that uses the adapter's Message Subscription service to subscribe for PeopleSoft messages from the TIBCO environment and send them to the PeopleSoft environment.

`ProcessMessageSubscriber` subscribes for PeopleSoft messages from the TIBCO environment, maps them to PeopleSoft schemas, and inserts data into the PeopleSoft database.

Setting Up the Example

Start the TIBCO EMS Server and PeopleSoft Design Time Adapter

This example uses JMS as the transport type. You need to start the TIBCO EMS Server and PeopleSoft Design Time Adapter:

1. Start the TIBCO EMS server.
2. Start the `adpsft8DTA_JMS` executable file under `TIB_ADPSFT8_HOME/bin`

Ensure that the TIBCO EMS Server is running and accessible to the machine on which the adapter is installed before running any of the processes.

Load the Project Archive in TIBCO Designer

To load the project archive:

1. Create a new project and import the project from the `Contact.zip` file located in: `TIB_ADPSFT8_HOME/examples/IntegrationBroker/Contact`. See [Importing a Project from a ZIP Archive in TIBCO Designer on page 7](#) for details.
2. Substitute the values of the following global variables for design-time connection:
 - `adpsft8.connection.appserver`
 - `adpsft8.connection.login`
 - `adpsft8.connection.password`
 - `adpsft8.connection.port`
 - `adpsft8DTAJmsProviderURL`
3. Establish design-time connection:
 - a. In the Project panel, navigate to the **MessageSubscriber** service instance.
 - b. Click the **Connect** button in the Design-time connection tab.
 - c. If the connection is established, the DTA Connection Successful dialog is displayed. Click **OK**.

4. Get schema:
 - a. In the project panel, navigate to the **CONTACT_SYNC_VERSION_1** service under the **MessageSubscriber** service instance.
 - b. Click the **Configuration** tab in the Configuration panel.
 - c. Enter **CONTACT_SYNC.VERSION_1** in the Message Name field.
 - d. Click **Fetch Message** next to the field.
 - e. Click **OK** in the displayed Fetch Message dialog.
 - f. Click **Get Schema**.
5. Save the project.
6. Exit TIBCO Designer.

Preparing PeopleSoft Integration Broker

After setting up the example, you need to create an integration scenario between the adapter and PeopleSoft Integration Broker. For detailed procedures, see "Preparing PeopleSoft Application Messages" in *TIBCO ActiveMatrix Adapter for PeopleSoft Configuration and Deployment*.

In particular, follow these steps when you make the configuration:

1. When you modify the `integrationGateway.Properties` file, set the queue for the test as:

`ig.jms.Queue#=TEST_PSFT_SUB_CONTACT`

 Remember to create a queue with the same name in the TIBCO EMS server.
2. When you define a Gateway and Node, specify the following connector properties:

 — **JMSQueue:** `TEST_PSFT_SUB_CONTACT`
3. When you define a Service Operation, select **Asynchronous - One Way** as the Operation Type of the Service Operation.
4. In PeopleTools Application Designer, update the PeopleCode with the following code:

```

Local Message &MSG;
Local string &XML_STRING;
Local File &SUCCESS_FILE;
Local XmlDocument &xmlDoc;
&MSG = GetMessage();
/* get an xmldoc object loaded with the content data. */
&XML_STRING = &MSG.GenXMLString();
&SUCCESS_FILE = GetFile("c:\templasync_sub_contact.txt", "w", "a", %FilePath_Absolute);
&SUCCESS_FILE.WriteLine(&XML_STRING);
&SUCCESS_FILE.Close();

```

Testing the Example

Configure the Process

To test `ProcessMessageSubscriber`, you need to log in to TIBCO Designer and configure the process first by following these steps:

1. From the Project panel, select `ProcessMessageSubscriber`.
2. Configure input:
 - a. Click the **File Poller** icon.
 - b. Click the **Configuration** tab and specify or browse to the location of the input XML file `Sub_Contact_InputData.xml`.

The default location is: `TIB_ADPSFT8_HOME\examples\IntegrationBroker\Contact\`.

3. Set break points for the process:
 - a. Click the **Set Breakpoints** icon from the menu bar.
 - b. In the displayed dialog, click the **Select All** button, then click **OK**.
4. Save the project.

Run the Test

After you make all the above configurations, follow these steps to test `GetProcess` in TIBCO Designer:

1. Select **Tools > Show Adapter Tester** from the menu and start the **MessageSubscriber** adapter instance in the displayed dialog.
2. Click the **Tester** tab to the left of the Project panel. The test panel replaces the project tree.
3. Click the **Start Testing Viewed Process** button.
4. In the Select Processes To Load dialog, select `ProcessMessageSubscriber` and click **Load Selected**. The process is now in test mode.
5. Once the process starter is highlighted (indicating a process has started), click the **Step To Next Activity** icon to step through the process.
6. Step through the process once more.
7. Click the **Stop Testing** icon to return to design mode.
8. Stop the **MessageSubscriber** adapter instance in the adapter tester.

Expected Results

The example results can be viewed in the respective output XML files configured in [Setting Up the Example on page 59](#).

ProcessMessageSubscriber data is subscribes for changes on CONTACT_SYNC_VERSION1 message. Based on the Contact message, the process generates an xml file that is published to the PeopleSoft environment. The PeopleSoft environment processes the message and generates output xml file.

The following CONTACT record is from an example output XML file:

```
<?xml version="1.0" ?>
- <CONTACT_SYNC>
- <FieldTypes>
- <CONTACT class="R">
  <SETID type="CHAR" />
  <CONTACT_ID type="CHAR" />
  <EFF_STATUS type="CHAR" />
  <CONTACT_FLAG type="CHAR" />
  <NAME1 type="CHAR" />
  <TITLE type="CHAR" />
  <LANGUAGE_CD type="CHAR" />
  <EMAILID type="CHAR" />
  <COMM_METHOD type="CHAR" />
  <SALUTATION_CD type="CHAR" />
  <SALUTATION type="CHAR" />
  <CREATE_CUSTOMER type="CHAR" />
  <LAST_MAINT_OPRID type="CHAR" />
  <DATE_LAST_MAINT type="DATE" />
  <AUTHORIZATION_ID type="CHAR" />
  <PERSON_ID type="CHAR" />
</CONTACT>
- <CONTACT_CUST class="R">
  <SETID type="CHAR" />
  <CONTACT_ID type="CHAR" />
  <CUSTOMER_SETID type="CHAR" />
  <CUST_ID type="CHAR" />
  <EFF_STATUS type="CHAR" />
  <CNTCT_SEQ_NUM type="NUMBER" />
  <ADDRESS_SEQ_NUM type="NUMBER" />
  <BILL_TO_FLG type="CHAR" />
  <SOLD_TO_FLG type="CHAR" />
  <SHIP_TO_FLG type="CHAR" />
  <LAST_MAINT_OPRID type="CHAR" />
  <DATE_LAST_MAINT type="DATE" />
</CONTACT_CUST>
- <CONTACT_PHN class="R">
  <SETID type="CHAR" />
  <CONTACT_ID type="CHAR" />
  <PHONE_TYPE type="CHAR" />
```

```

    <COUNTRY_CODE type="CHAR" />
    <PHONE type="CHAR" />
    <EXTENSION type="CHAR" />
    <LAST_MAINT_OPRID type="CHAR" />
    <DATE_LAST_MAINT type="DATE" />
  </CONTACT_PHN>
- <CONTACT_PAGER class="R">
  <SETID type="CHAR" />
  <CONTACT_ID type="CHAR" />
  <PHONE_TYPE type="CHAR" />
  <COUNTRY_CODE type="CHAR" />
  <PHONE type="CHAR" />
  <EXTENSION type="CHAR" />
  <PASSWORD type="CHAR" />
  <LAST_MAINT_OPRID type="CHAR" />
  <DATE_LAST_MAINT type="DATE" />
</CONTACT_PAGER>
- <CONTACT_CARD class="R">
  <SETID type="CHAR" />
  <CONTACT_ID type="CHAR" />
  <CUSTOMER_SETID type="CHAR" />
  <CUST_ID type="CHAR" />
  <CR_CARD_NBR type="CHAR" />
  <CR_CARD_TYPE type="CHAR" />
  <PRIMARY_CARD type="CHAR" />
  <CR_CARD_FNAME type="CHAR" />
  <CR_CARD_LNAME type="CHAR" />
  <CR_CARD_EXPMO type="CHAR" />
  <CR_CARD_EXPYR type="CHAR" />
  <CR_CARD_DIGITS type="CHAR" />
  <ADDR_SEQ_NUM type="NUMBER" />
  <LAST_MAINT_OPRID type="CHAR" />
  <DATE_LAST_MAINT type="DATE" />
</CONTACT_CARD>
- <PSCAMA class="R">
  <LANGUAGE_CD type="CHAR" />
  <AUDIT_ACTN type="CHAR" />
  <BASE_LANGUAGE_CD type="CHAR" />
  <MSG_SEQ_FLG type="CHAR" />
  <PROCESS_INSTANCE type="NUMBER" />
  <PUBLISH_RULE_ID type="CHAR" />
  <MSGNODENAME type="CHAR" />
</PSCAMA>
</FieldTypes>
- <MsgData>
- <Transaction>
- <CONTACT class="R">
  <SETID>site id </SETID>
  <CONTACT_ID>cx </CONTACT_ID>
  <EFF_STATUS>cx </EFF_STATUS>
  <CONTACT_FLAG>cx </CONTACT_FLAG>
  <NAME1>cx </NAME1>
  <TITLE>cx </TITLE>
  <LANGUAGE_CD>cx </LANGUAGE_CD>
  <EMAILID>cx </EMAILID>
  <COMM_METHOD>cx </COMM_METHOD>
  <SALUTATION_CD>cx </SALUTATION_CD>

```



```

<SALUTATION>cx </SALUTATION>
<CREATE_CUSTOMER>cx </CREATE_CUSTOMER>
<LAST_MAINT_OPRID />
<DATE_LAST_MAINT />
<AUTHORIZATION_ID>sa</AUTHORIZATION_ID>
<PERSON_ID>ds</PERSON_ID>
</CONTACT>
- <CONTACT_CUST class="R">
  <SETID />
  <CONTACT_ID />
  <CUSTOMER_SETID />
  <CUST_ID />
  <EFF_STATUS />
  <CNTCT_SEQ_NUM />
  <ADDRESS_SEQ_NUM />
  <BILL_TO_FLG />
  <SOLD_TO_FLG />
  <SHIP_TO_FLG />
  <LAST_MAINT_OPRID />
  <DATE_LAST_MAINT />
</CONTACT_CUST>
- <CONTACT_PHN class="R">
  <SETID />
  <CONTACT_ID />
  <PHONE_TYPE />
  <COUNTRY_CODE />
  <PHONE />
  <EXTENSION />
  <LAST_MAINT_OPRID />
  <DATE_LAST_MAINT />
</CONTACT_PHN>
- <CONTACT_PAGER class="R">
  <SETID />
  <CONTACT_ID />
  <PHONE_TYPE />
  <COUNTRY_CODE />
  <PHONE />
  <EXTENSION />
  <PASSWORD />
  <LAST_MAINT_OPRID />
  <DATE_LAST_MAINT />
</CONTACT_PAGER>
- <CONTACT_CARD class="R">
  <SETID />
  <CONTACT_ID />
  <CUSTOMER_SETID />
  <CUST_ID />
  <CR_CARD_NBR />
  <CR_CARD_TYPE />
  <PRIMARY_CARD />
  <CR_CARD_FNAME />
  <CR_CARD_LNAME />
  <CR_CARD_EXPMO />
  <CR_CARD_EXPYR />
  <CR_CARD_DIGITS />
  <ADDR_SEQ_NUM />
  <LAST_MAINT_OPRID />

```

```
<DATE_LAST_MAINT />
</CONTACT_CARD>
- <PSCAMA class="R">
  <LANGUAGE_CD>ENG</LANGUAGE_CD>
  <AUDIT_ACTN />
  <BASE_LANGUAGE_CD>ENG</BASE_LANGUAGE_CD>
  <MSG_SEQ_FLG />
  <PROCESS_INSTANCE>0</PROCESS_INSTANCE>
  <PUBLISH_RULE_ID />
  <MSGNODENAME />
</PSCAMA>
</Transaction>
</MsgData>
</CONTACT_SYNC>
```

Chapter 9

TIBCO ActiveMatrix BusinessWorks: Working with the Country Message

This example shows how to use the Request-Response Invocation Service within a TIBCO ActiveMatrix BusinessWorks process to get a request from PeopleSoft, process it, and respond back with a message. The message is transferred to the PeopleSoft environment using Integration Broker.

Topics

- [Example Description, page 68](#)
- [Setting Up the Example, page 69](#)
- [Preparing PeopleSoft Integration Broker, page 71](#)
- [Testing the Example, page 72](#)
- [Expected Results, page 73](#)

Example Description

This example shows how the adapter's Request-Response Invocation Service can be used in a TIBCO ActiveMatrix BusinessWorks process. The example has one process, `ProcessMessageRPCClient`, that uses the adapter's Request-Response Invocation Service to fetch data from PeopleSoft for key values that you specify in an input XML file.

`ProcessMessageRPCClient` queries PeopleSoft based on the given key values and fetches the data. The data is sent back by the adapter to TIBCO ActiveMatrix BusinessWorks and is written into an output XML file.

In a typical business scenario, an adapter to another application (for example, SAP or Siebel) publishes a message into the TIBCO environment. TIBCO ActiveMatrix BusinessWorks receives the messages and writes them to an input XML file. The content of this XML file will contain the changes to be updated in PeopleSoft. TIBCO ActiveMatrix Adapter for PeopleSoft uses the Request-Response Invocation service to update the data in PeopleSoft and sends a reply to TIBCO ActiveMatrix BusinessWorks.

In this case, you may specify key values in the input XML file. `ProcessMessageRPCClient` queries PeopleSoft based on the key values. The COUNTRY details are updated in PeopleSoft. On receiving the modified data, TIBCO ActiveMatrix BusinessWorks logs it to an XML file.

Setting Up the Example

Start the TIBCO EMS Server and PeopleSoft Design Time Adapter

This example uses JMS as the transport type. You need to start the TIBCO EMS Server and PeopleSoft Design Time Adapter:

1. Start the TIBCO EMS server.
2. Start the `adpsft8DTA_JMS` executable file under `TIB_ADPSFT8_HOME/bin`

Ensure that the TIBCO EMS Server is running and accessible to the machine on which the adapter is installed before running any of the processes.

Load the Project Archive in TIBCO Designer

To load the project archive:

1. Create a new project and import the project from the `Country.zip` file located in: `TIB_ADPSFT8_HOME/examples/IntegrationBroker/Country`. See [Importing a Project from a ZIP Archive in TIBCO Designer on page 7](#) for details.
2. Substitute the values of the following global variables for design-time connection:
`adpsft8.connection.appserver`
`adpsft8.connection.login`
`adpsft8.connection.password`
`adpsft8.connection.port`
`adpsft8DTAJmsProviderURL`
3. Establish design-time connection:
 - a. In the Project panel, navigate to the **MessageRPCClient** service instance.
 - b. Click the **Connect** button in the Design-time connection tab.
 - c. If the connection is established, the DTA Connection Successful dialog is displayed. Click **OK**.

4. Get schema:
 - a. In the project panel, navigate to the **COUNTRY_SYNC_VERSION_1** service under the MessageRPCClient service instance.
 - b. Click the **Configuration** tab in the Configuration panel.
 - c. Enter **COUNTRY_SYNC.VERSION_1** in the Message Name field.
 - d. Click **Fetch Message** next to the field.
 - e. Click **OK** in the displayed Fetch Message dialog.
 - f. Click **Get Schema**.
5. Save the project.
6. Exit TIBCO Designer.

Preparing PeopleSoft Integration Broker

After setting up the example, you need to create an integration scenario between the adapter and PeopleSoft Integration Broker. For detailed procedures, see "Preparing PeopleSoft Application Messages" in *TIBCO ActiveMatrix Adapter for PeopleSoft Configuration and Deployment*.

In particular, follow these steps when you make the configuration:

1. When you modify the `integrationGateway.Properties` file, set the queue for the test as:

```
ig.jms.Queue#=TEST.CLIENT
```

Remember to create a queue with the same name in the TIBCO EMS server.

2. When you define a Gateway and Node, specify the following connector properties:

— **JMSQueue:** TEST.CLIENT

— **serviceType:** RPCCLIENT

3. When you define a Service Operation, select **Synchronous** as the Operation Type of the Service Operation.

```
Local string &XML_STRING;
Local Message &request_MSG, &response_MSG;
Local Rowset &request_RS;
Local Rowset &response_RS;
Local Rowset &MSG_RS;
Local Record &REC;
&request_RS = GetLevel0();
&request_MSG = CreateMessage(Message.COUNTRY_SYNC);
&request_MSG.CopyRowset(&request_RS);
/* publish request and wait for the response = &XML_STRING;*/
&response_MSG = &request_MSG.SyncRequest();
If (&response_MSG.ResponseStatus = 0) Then
    &MSG_RS = &response_MSG.GetRowset();
    &REC = &response_MSG.GetRowset().GetRow(1).GetRecord(Record.COUNTRY);
    &XML_STRING = &response_MSG.GenXMLString();
End-If
```

Testing the Example

Configure the Process

To test `ProcessMessageRPCClient`, set break points for the process:

1. Click the **Set Breakpoints** icon from the menu bar.
2. In the displayed dialog, click the **Select All** button, then click **OK**.
3. Save the project.

Run the Test

After you make all the above configurations, follow these steps to test `GetProcess` in TIBCO Designer:

1. Select **Tools > Show Adapter Tester** from the menu and start the **MessageRPCClient** adapter instance in the displayed dialog.
2. Click the **Tester** tab to the left of the Project panel. The test panel replaces the project tree.
3. Click the **Start Testing Viewed Process** button.
4. In the Select Processes To Load dialog, select **ProcessMessageRPCClient** and click **Load Selected**. The process is now in test mode.
5. Once the process starter is highlighted (indicating a process has started), click the **Step To Next Activity** icon to step through the process.
6. Update PeopleSoft: Log in to navigate to **Setup CRM > Common Definitions > Location > Country TBL**. Open an existing Country definition, update and save.
7. Step through the process once more.
8. Click the **Stop Testing** icon to return to design mode.
9. Stop the **MessageRPCClient** adapter instance in the adapter tester.

Expected Results

The example results can be viewed in the respective output XML files configured in [Setting Up the Example on page 69](#).

ProcessMessageRPCClient data is fetched based on the COUNTRY_SYNC_VERSION1 message. The adapter receives a request whenever the COUNTRY_SYNC_VERSION1 message is modified. The adapter processes the request and sends a response back to PeopleSoft in an xml file.

The following COUNTRY record is from an example output XML file:

```
<?xml version="1.0" ?>
- <COUNTRY_SYNC>
- <FieldTypes>
- <COUNTRY_TBL class="R">
  <ADDR1_AVAIL type="CHAR" />
  <ADDR1_LBL type="CHAR" />
  <ADDR2_AVAIL type="CHAR" />
  <ADDR2_LBL type="CHAR" />
  <ADDR3_AVAIL type="CHAR" />
  <ADDR3_LBL type="CHAR" />
  <ADDR4_AVAIL type="CHAR" />
  <ADDR4_LBL type="CHAR" />
  <ADDR_FIELD1_AVAIL type="CHAR" />
  <ADDR_FIELD1_LBL type="CHAR" />
  <ADDR_FIELD2_AVAIL type="CHAR" />
  <ADDR_FIELD2_LBL type="CHAR" />
  <ADDR_FIELD3_AVAIL type="CHAR" />
  <ADDR_FIELD3_LBL type="CHAR" />
  <CITY_AVAIL type="CHAR" />
  <CITY_LBL type="CHAR" />
  <COUNTRY type="CHAR" />
  <COUNTRY_2CHAR type="CHAR" />
  <COUNTY_AVAIL type="CHAR" />
  <COUNTY_LBL type="CHAR" />
  <DESCR type="CHAR" />
  <DESCRSHORT type="CHAR" />
  <EU_MEMBER_STATE type="CHAR" />
  <GBSYS_CFGPATH_UK type="CHAR" />
  <GBSYS_NRPATH_UK type="CHAR" />
  <GEO_CODE_AVAIL type="CHAR" />
  <GEO_CODE_LBL type="CHAR" />
  <HOUSE_TYPE_AVAIL type="CHAR" />
  <HOUSE_TYPE_LBL type="CHAR" />
  <IN_CITY_LIM_AVAIL type="CHAR" />
  <IN_CITY_LIM_LBL type="CHAR" />
  <NUM1_AVAIL type="CHAR" />
  <NUM1_LBL type="CHAR" />
  <NUM2_AVAIL type="CHAR" />
  <NUM2_LBL type="CHAR" />
  <POSTAL_AVAIL type="CHAR" />
  <POSTAL_LBL type="CHAR" />
  <POST_SRCH_AVAIL type="CHAR" />
```

```

    <STATE_AVAIL type="CHAR" />
    <STATE_LBL type="CHAR" />
    <SYNCDTM type="CHAR" />
    <SYNCID type="NUMBER" />
  </COUNTRY_TBL>
- <PSCAMA class="R">
  <LANGUAGE_CD type="CHAR" />
  <AUDIT_ACTN type="CHAR" />
  <BASE_LANGUAGE_CD type="CHAR" />
  <MSG_SEQ_FLG type="CHAR" />
  <PROCESS_INSTANCE type="NUMBER" />
  <PUBLISH_RULE_ID type="CHAR" />
  <MSGNODENAME type="CHAR" />
</PSCAMA>
</FieldTypes>
- <MsgData>
- <Transaction>
- <COUNTRY_TBL class="R">
  <ADDR1_AVAIL />
  <ADDR1_LBL />
  <ADDR2_AVAIL />
  <ADDR2_LBL />
  <ADDR3_AVAIL />
  <ADDR3_LBL />
  <ADDR4_AVAIL />
  <ADDR4_LBL />
  <ADDR_FIELD1_AVAIL />
  <ADDR_FIELD1_LBL />
  <ADDR_FIELD2_AVAIL />
  <ADDR_FIELD2_LBL />
  <ADDR_FIELD3_AVAIL />
  <ADDR_FIELD3_LBL />
  <CITY_AVAIL />
  <CITY_LBL />
  <COUNTRY>India</COUNTRY>
  <COUNTRY_2CHAR />
  <COUNTY_AVAIL />
  <COUNTY_LBL />
  <DESCR>India</DESCR>
  <DESCRSHORT>IND</DESCRSHORT>
  <EU_MEMBER_STATE />
  <GBSYS_CFGPATH_UK />
  <GBSYS_NRPATH_UK />
  <GEO_CODE_AVAIL />
  <GEO_CODE_LBL />
  <HOUSE_TYPE_AVAIL />
  <HOUSE_TYPE_LBL />
  <IN_CITY_LIM_AVAIL />
  <IN_CITY_LIM_LBL />
  <NUM1_AVAIL />
  <NUM1_LBL />
  <NUM2_AVAIL />
  <NUM2_LBL />
  <POSTAL_AVAIL />
  <POSTAL_LBL />
  <POST_SRCH_AVAIL />
  <STATE_AVAIL />

```

```
<STATE_LBL />
<SYNCDTTM />
<SYNCID />
</COUNTRY_TBL>
- <PSCAMA class="R">
  <LANGUAGE_CD>ENG</LANGUAGE_CD>
  <AUDIT_ACTN />
  <BASE_LANGUAGE_CD>ENG</BASE_LANGUAGE_CD>
  <MSG_SEQ_FLG />
  <PROCESS_INSTANCE>0</PROCESS_INSTANCE>
  <PUBLISH_RULE_ID />
  <MSGNODENAME />
  </PSCAMA>
</Transaction>
</MsgData>
</COUNTRY_SYNC>
```

Index

A

API [xiv](#)

C

changes from the previous release [viii](#)
commands

TIB_PS8_ADAPTER [7](#)

conventions used in this manual [x](#)

customer support [xiv](#), [xvi](#)

D

Deploy and Run the Example [14](#)

documents, related [x](#)

E

Example Description [12](#), [18](#), [24](#), [30](#), [40](#), [50](#), [58](#), [68](#)

Expected Results [16](#), [22](#), [27](#), [37](#), [47](#), [56](#), [63](#), [73](#)

G

Get [3](#)

O

Overview [2](#)

P

Prerequisites [6](#)

Q

QOS [xiv](#)

R

related documents [x](#)

RV [xiv](#), [xiv](#)

RVCN [xiv](#)

RVDQ [xiv](#)

S

Setup the Example [13](#), [19](#), [25](#), [39](#), [41](#), [51](#), [59](#), [69](#)

support, contacting [xiv](#), [xvi](#)

Sync [5](#)

T

technical support [xiv](#), [xvi](#)

Test the Example [21](#), [26](#), [33](#), [43](#), [55](#), [62](#), [72](#)

TIB_PS8_ADAPTER PeopleSoft project [6](#)

TIBCO_HOME [xi](#)

U

- Update [3](#)
- Update Operation Using a Publish-Subscribe Scenario [4](#)
- Update Operation Using an Request-Response Scenario [4](#)
- Using TIBCO ActiveMatrix BusinessWorks [8](#)