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, Two-Second Advantage, TIBCO, The Power of Now, TIBCO ActiveMatrix BusinessWorks, TIBCO ActiveEnterprise, TIBCO Rendezvous, TIBCO Designer, TIBCO Administrator, TIBCO Adapter SDK, TIBCO Runtime Agent, TIBCO ActiveMatrix Adapter, TIBCO ActiveMatrix Adapter for MQSeries, TIBCO Enterprise Message Service, and TIBCO Hawk are either registered trademarks or trademarks of TIBCO Software Inc. in the United States and/or other countries.

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

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

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

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

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

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

Copyright © 1998-2013 TIBCO Software Inc. ALL RIGHTS RESERVED.

TIBCO Software Inc. Confidential Information
# Contents

## Chapter 3  Adapter Instance Options ................................................................. 19
- Overview of an Adapter Instance ........................................................................ 20
  - Configuration Tasks ....................................................................................... 20
- Adapter Instance Fields .................................................................................... 21
  - Configuration Tab .......................................................................................... 21
  - Runtime Connection Tab ................................................................................ 22
  - Logging Tab ................................................................................................... 23
  - Startup Tab .................................................................................................... 26

## Chapter 4  Adapter Services Options ................................................................. 27
- Transport Specific Service Options ..................................................................... 28
  - Transport Type ............................................................................................... 28
  - Quality of Service ........................................................................................... 29
  - Wire Format ..................................................................................................... 29
  - Connection Factory Type ............................................................................... 30
  - Delivery Mode ................................................................................................. 30
- Request-Response Service and Request-Response Invocation Service ............... 32
  - Configuration Tab .......................................................................................... 32
  - Transport Tab .................................................................................................. 33
  - Schema Tab ..................................................................................................... 35
  - SWIFTNet Connection Tab ............................................................................. 36

## Chapter 5  Deploying an Adapter Using TIBCO Administrator ............................. 39
- Creating an EAR File in TIBCO Designer ............................................................ 40
- Deploying a Project ............................................................................................. 41
- Starting or Stopping the Adapter ......................................................................... 42
- Monitoring the Adapter ....................................................................................... 43

## Chapter 6  Advanced Configuration Options .................................................... 45
- Modifying TIBCO Rendezvous Session Parameters ......................................... 46
- Modifying TIBCO JMS Session Parameters ....................................................... 47
- Using the Adapter with a Revision Control System .............................................. 48
- Using Global Variables ....................................................................................... 51
  - Variable Specification ..................................................................................... 51
  - Predefined Global Variables .......................................................................... 53

## Appendix A  Trace Messages ................................................................................. 55
- Overview of Trace Messages ............................................................................... 56
- Trace Message Fields ........................................................................................ 57
# Figures

| Figure 1 | Typical Request-Response Service (Client) Flow | 6 |
| Figure 2 | Build a valid XML Payload by Using SWIFTNet Solution XSDs (Client) | 7 |
| Figure 3 | Typical Request-Response Invocation Service (Server) Flow | 7 |
| Figure 4 | Build a valid XML Payload by Using SWIFTNet Solution XSDs (Server) | 8 |
| Figure 5 | Creating a New Empty Project in TIBCO Designer | 11 |
| Figure 6 | Saving the New Empty Project | 12 |
| Figure 7 | TIBCO Designer Panels | 12 |
| Figure 8 | Creating a New Log Sinks | 25 |
| Figure 9 | TIBCO Rendezvous Session Configuration Screen | 46 |
| Figure 10 | TIBCO Enterprise Message Service Session Configuration Screen | 47 |
| Figure 11 | InterAct Real Time Client | 65 |
| Figure 12 | InterAct Real Time Server | 67 |
## Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table 1</td>
<td>General Typographical Conventions</td>
<td>xiii</td>
</tr>
<tr>
<td>Table 2</td>
<td>Adapter Instance Configuration Tab</td>
<td>21</td>
</tr>
<tr>
<td>Table 3</td>
<td>Adapter Instance Runtime Connection Tab</td>
<td>22</td>
</tr>
<tr>
<td>Table 4</td>
<td>Adapter Instance Logging Tab</td>
<td>23</td>
</tr>
<tr>
<td>Table 5</td>
<td>Adapter Instance Startup Tab</td>
<td>26</td>
</tr>
<tr>
<td>Table 6</td>
<td>Configuration Tab of a Request-Response Service or a Request-Response Invocation Service</td>
<td>32</td>
</tr>
<tr>
<td>Table 7</td>
<td>Transport Tab of a Request-Response Service and a Request-Response Invocation Service</td>
<td>33</td>
</tr>
<tr>
<td>Table 8</td>
<td>A Request-Response Service or a Request-Response Invocation Service</td>
<td>36</td>
</tr>
<tr>
<td>Table 9</td>
<td>Predefined Global Variables</td>
<td>53</td>
</tr>
<tr>
<td>Table 10</td>
<td>Tracing Fields</td>
<td>57</td>
</tr>
<tr>
<td>Table 11</td>
<td>Status Messages</td>
<td>60</td>
</tr>
</tbody>
</table>
Preface

TIBCO ActiveMatrix Adapter for SWIFT is a bidirectional gateway for SWIFTNet InterAct messages between the SWIFT network (via SAG) and TIBCO applications.

This guide describes the architecture and supported features of TIBCO ActiveMatrix Adapter for SWIFTNet. It also describes how to install and configure the adapter.

Topics

- Related Documentation, page xii
- Typographical Conventions, page xiii
- Connecting with TIBCO Resources, page xv
Related Documentation

This section lists documentation resources.

**TIBCO ActiveMatrix Adapter for SWIFTNet Documentation**

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

- *TIBCO ActiveMatrix Adapter for SWIFT Installation*  Read this manual for instructions on site preparation and installation.
- *TIBCO ActiveMatrix Adapter for SWIFTNet User’s Guide*  Read this manual for the concepts relating to the adapter, the applications with which the adapter interacts, configuration and deployment information.
- *TIBCO ActiveMatrix Adapter for SWIFTNet Examples Guide*  Read this manual to work through the examples provided with the adapter.
- *TIBCO ActiveMatrix Adapter for SWIFT Release Notes*  Read this release notes for a list of new and changed features. This document also contains lists of closed issues and known issues for this release.

**Other TIBCO Product Documentation**

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

- TIBCO Administrator™
- TIBCO ActiveMatrix Adapter™
- TIBCO ActiveMatrix Adapter™ for MQSeries
- TIBCO ActiveMatrix BusinessWorks™
- TIBCO® Adapter SDK
- TIBCO Designer™
- TIBCO Enterprise Message Service™
- TIBCO Hawk®
- TIBCO Rendezvous™
- TIBCO Runtime Agent™
Typographical Conventions

The following typographical conventions are used in this manual.

Table 1  General Typographical Conventions

<table>
<thead>
<tr>
<th>Convention</th>
<th>Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>TIBCO_HOME</td>
<td>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.</td>
</tr>
<tr>
<td>ENV_NAME</td>
<td>An installation environment consists of the following properties:</td>
</tr>
<tr>
<td></td>
<td>• Name Identifies the installation environment. This name is referenced in documentation as ENV_NAME. 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 &gt; All Programs menu.</td>
</tr>
<tr>
<td></td>
<td>• Path The directory into which the product is installed. This directory is referenced in documentation as TIBCO_HOME.</td>
</tr>
<tr>
<td>TIB_ADSWIFT_HOME</td>
<td>The product is installed into a directory within a TIBCO_HOME.</td>
</tr>
<tr>
<td>CONFIG_HOME</td>
<td>TIBCO ActiveMatrix Adapter for SWIFTNet - FIN installs the product in the directory which is referenced in documentation as TIB_ADSWIFT_HOME. The default value of TIBCO_ADSWIFT_HOME depends on the operating system. For example, on Windows systems, the default value is C:\tibco\adapter\adswift\version.</td>
</tr>
<tr>
<td>CONFIG_HOME</td>
<td>A TIBCO configuration directory stores configuration data generated by TIBCO products. Configuration data can include sample scripts, session data, configured binaries, logs, and so on. This directory is referenced in documentation as CONFIG_HOME. The default location of the directory is USER_HOME\ENV_NAME\tibco\cfgmgmt\product_name. For example, on Windows, the default location is C:\Documents and Settings\username\Application Data\ENV_NAME\tibco\cfgmgmt\product_name.</td>
</tr>
<tr>
<td>code font</td>
<td>Code font identifies commands, code examples, filenames, pathnames, and output displayed in a command window. For example: Use MyCommand to start the foo process.</td>
</tr>
</tbody>
</table>

Use MyCommand to start the foo process.
Table 1  General Typographical Conventions (Cont’d)

<table>
<thead>
<tr>
<th>Convention</th>
<th>Use</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>bold code font</strong></td>
<td>Bold code font is used in the following ways:</td>
</tr>
<tr>
<td></td>
<td>• In procedures, to indicate what a user types. For example: Type <code>admin</code>.</td>
</tr>
<tr>
<td></td>
<td>• In large code samples, to indicate the parts of the sample that are of particular interest.</td>
</tr>
<tr>
<td></td>
<td>• In command syntax, to indicate the default parameter for a command. For example, if no parameter is specified, <code>MyCommand</code> is enabled: `MyCommand [enable</td>
</tr>
<tr>
<td><strong>italic font</strong></td>
<td>Italic font is used in the following ways:</td>
</tr>
<tr>
<td></td>
<td>• To indicate a document title. For example: See <em>TIBCO BusinessWorks Concepts</em>.</td>
</tr>
<tr>
<td></td>
<td>• To introduce new terms For example: A portal page may contain several portlets. <em>Portlets</em> are mini-applications that run in a portal.</td>
</tr>
<tr>
<td></td>
<td>• To indicate a variable in a command or code syntax that you must replace. For example: <code>MyCommand pathname</code></td>
</tr>
<tr>
<td><strong>Key combinations</strong></td>
<td>Key name separated by a plus sign indicate keys pressed simultaneously. For example: Ctrl+C.</td>
</tr>
<tr>
<td></td>
<td>Key names separated by a comma and space indicate keys pressed one after the other. For example: Esc, Ctrl+Q.</td>
</tr>
<tr>
<td>![Note Icon]</td>
<td>The note icon indicates information that is of special interest or importance, for example, an additional action required only in certain circumstances.</td>
</tr>
<tr>
<td>![Tip Icon]</td>
<td>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.</td>
</tr>
<tr>
<td>![Warning Icon]</td>
<td>The warning icon indicates the potential for a damaging situation, for example, data loss or corruption if certain steps are taken or not taken.</td>
</tr>
</tbody>
</table>
Connecting with TIBCO Resources

How to Join TIBCOmmunity

TIBCOmmunity 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. TIBCOmmunity 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 username and password. If you do not have a username, you can request one.
Chapter 1  Concepts of TIBCO ActiveMatrix Adapter for SWIFTNet

This chapter explains an overview of TIBCO ActiveMatrix Adapter for SWIFTNet, and describes the modes for exchanging SWIFTNet InterAct messages.

Topics

- SWIFTNet Overview, page 2
- Adapter Overview, page 4
- Adapter Services, page 6
SWIFTNet Overview

This session gives an overview of SWIFTNet.

SWIFTNet

Secure IP Network (SIPN) or SWIFTNet is the replacement to the previously X.25 protocol-based network used by SWIFT community since 2003. Unlike its predecessor, SWIFTNet supports SWIFTNet InterAct, SWIFTNet FileAct, and SWIFTNet Browse (XML-based communication protocols) messages in addition to the legacy SWIFT FIN (presently referred as SWIFTNet - FIN).

This guide discusses SWIFTNet InterAct messages only. For more information of SWIFTNet - FIN, see TIBCO ActiveMatrix Adapter for SWIFTNet - FIN User’s Guide.

SWIFT Alliance Gateway

SWIFT Alliance Gateway (SAG) is an interface for accessing SWIFTNet. It is installed on SWIFTNet Link. In addition, it provides several different connectivity and usability features for SWIFTNet users. SAG is designed to concentrate traffic from multiple SWIFT Alliance WebStations. It provides a graphical user interface for the administration of the SAG and related SWIFTNet security administration functions. It can serve as a message concentrator, receiving messages from other applications for passage through SWIFTNet.

SWIFT Alliance Gateway Remote API

SWIFT Alliance Gateway Remote API is a software package that establishes a communication link with the Remote API Host Adapter (RAHA) component of SWIFT Alliance Gateway (SAG), either from a SWIFTNet application existing on a remote computer, or from a SWIFTNet application existing on the computer where SAG is installed.

Using Remote API, developed applications can be run directly on top of the SWIFTNet Link, and software can use SWIFT Alliance Gateway transparently as a concentrator for their SWIFTNet traffic, thereby implementing the single window concept. Remote API offers two sets of APIs: SWIFTNet Link specific and SAG specific. Message flow, from a Remote API instance to SAG, is managed by the SWIFTNet Adapter, a sub-component of SAG’s Application Interface.
SWIFTNet Messaging Services

SWIFTNet InterAct

SWIFTNet InterAct provides secure and reliable exchange of financial messages. It offers a broad range of message transmission modes:

- **Store-and-Forward Messaging**

  Store-and-forward messaging frees you from worrying about whether or not your correspondents are online at the time you send the messages, if the recipients are not online, then these messages will be stored and delivered as soon as the recipient is ready to receive them. As a result, it provides an ideal way to send individual instructions, confirmations, and reports to large numbers of correspondents, some of whom may be in different time zones.

- **Real-Time Messaging**

  Real-time messaging offers a low-cost alternative (compared to the store-and-forward mode) for messages, which are destined for correspondents that are online at the time of transmission. As a result, it is suited for sending individual instructions, confirmations, and reports to a few large recipients, or for messages to market infrastructures.
Chapter 1  Concepts of TIBCO ActiveMatrix Adapter for SWIFTNet

**Adapter Overview**

The adapter connects the SWIFT Alliance Gateway (SAG) and your back office applications. This connection enables the two-way transfer of SWIFTNet InterAct messages between applications on both sides. The adapter thus makes possible the single window concept that SWIFT envisions. According to SWIFT, their secure network is the platform for the financial industry to access or offer services securely, reliably, efficiently, and cost effectively.

The adapter supports InterAct messaging service in the store-and-forward mode and the real-time mode. The adapter provides client as well as server message flows in both real-time and store-and-forward modes.

For a general introduction to adapters and the services they provide, read *TIBCO Adapter Concepts*. The book is part of the adapter documentation set.

**Components**

The adapter has two main components: adapter palette and the runtime adapter. The adapter palette comprises an adapter-specific GUI that seamlessly integrates with TIBCO Designer. It allows you to configure the adapter, and stores the configuration in persistent storage. You will require TIBCO Designer for this, which is installed as part of the TIBCO Runtime Agent (TRA) installation. TRA must be installed before installing the adapter.

**Features**

The adapter contains the following features:

- **An Easy-To-Use GUI**
  TIBCO Designer is used to configure and maintain the adapter. You can quickly specify operational parameters and change them as needed.

- **Support for TIBCO ActiveMatrix BusinessWorks**
  TIBCO ActiveMatrix BusinessWorks is a scalable, extensible, and easy to use integration platform that allows you to develop, deploy, and run integration projects. A TIBCO ActiveMatrix BusinessWorks project integrates enterprise applications and automates business processes.

- **Support for TIBCO Messaging Transports**
  The adapter supports TIBCO Rendezvous and JMS as messaging transports.
The TIBCO Rendezvous transport supports the following qualities of service:

Reliable—This ensures that each multicast or broadcast message is received as long as the physical network and packet recipients are working, and the loss of a message is detected.

Certified Delivery—Assures every certified message reaches its intended recipient in the order sent. When delivery is not possible, both sending and listening programs receive explicit information about each undelivered message.

Distributed Queue—This is designed to deliver a message to one-of-many service listeners (workers). It contains features of both certified messaging and fault tolerance. Senders of distributed queue messages are ordinary certified message senders.

The JMS transport supports the following connection types (The TIBCO Enterprise Message Service software must be installed when using the JMS transport.)

Queue—Where a service sends to a queue or receives from a queue. This message protocol is known as point-to-point because one and only one receiver consumes messages sent to a queue. A queue retains all messages sent until such time the messages are consumed or expired. Each message has only one receiver, though multiple receivers can connect to the queue. JMS messages use the XML message wire format only.

Topic—Where a service publishes to a topic or subscribes to a topic. This type of message protocol is also known as broadcast messaging because messages are sent over the network and received by all interested subscription services, similar to how radio or television signals are broadcast and received.

The JMS mode of transport supports two types of delivery modes, Persistent and Non-Persistent.

- **Support for SWIFTNet InterAct Message Types**

The adapter supports all SWIFTNet InterAct services in the real-time and store-and-forward modes.
Adapter Services

TIBCO ActiveMatrix Adapter for SWIFTNet provides the following services:

- Request-Response Service, page 6
- Request-Response Invocation Service, page 7

Request-Response Service

The adapter gets a request from the TIBCO-enabled SWIFTNet application and sends the request to SWIFTNet via SWIFT Alliance Gateway (SAG). When a response comes from SWIFTNet, the adapter sends the response to the TIBCO-enabled SWIFTNet application. See Figure 1.

Figure 1  Typical Request-Response Service (Client) Flow

The adapter supports the Request-Response scenarios with a Remote Procedure Call server. When the adapter receives a request, it takes the raw requested data, converts it into a formatted SWIFTNet InterAct message schema, then sends it to the SWIFTNet via SAG.

The TIBCO application sending the SWIFTNet message is responsible for building a valid XML payload using SWIFTNet solution XSDs supplied by SWIFT (for example, Exceptions and Investigation, Cash Reporting, and so on). See Figure 2.
Request-Response Invocation Service

The adapter gets a request from SWIFTNet, which is sent by the SWIFTNet central store-and-forward system or some other systems similar to it on SWIFTNet. Adapter sends this request to the appropriate TIBCO-enabled SWIFTNet application that is configured to receive this message. When this application replies to the adapter, the adapter sends this response to SWIFTNet via SAG. See Figure 3 and Figure 4.

Figure 3   Typical Request-Response Invocation Service (Server) Flow
Figure 4  Build a valid XML Payload by Using SWIFTNet Solution XSDs (Server)

Note: AE or XML Message contains "XML Payload" that conforms to the SWIFTNet solution that is being implemented.
Chapter 2  Getting Started

This chapter presents examples that demonstrate key adapter features. You can work through these examples to get a hands-on understanding of how the adapter works.

Topics

- Prerequisites for the Configuration Exercise, page 10
- Creating a Project, page 11
- Configuring an Adapter Instance, page 13
- Configuring the Services, page 14
- Exporting the Project, page 15
- Deploying and Testing the Adapter, page 16
Prerequisites for the Configuration Exercise

Before starting the exercise, ensure that all required software has been installed and is operating correctly. For a list of required software, see *TIBCO ActiveMatrix Adapter for SWIFT Installation*.

This manual assumes that you know how to drag icons in TIBCO Designer and how to save projects. If you are not familiar with these topics, read the documentation for TIBCO Designer. It can be accessed from Help > Designer Help menu in TIBCO Designer.

TIBCO Designer is also used to configure TIBCO Adapter for SWIFTNet instances. See Chapter 3, Adapter Instance Options, on page 19 for details.

Scenario Overview

The exercises in this chapter are designed to help you become familiar with basic adapter configuration.

Running through these exercises will familiarize you with the different SWIFTNet operations that a service supports.

Connections to SWIFT Alliance Gateway (SAG)

You must ensure that the adapter to the SAG connections are in place.

Permissions to Access Repository Server

If your site is using TIBCO Administrator to set access control to the repository server, you must have the account name and password available that is used by the adapter to log in to the server.

The repository server must be running on your subnet before starting the exercise.
Creating a Project

The TIBCO Designer is used to configure adapter instances. When starting the Designer, you must create or select a project. A project contains the configuration files that define options used by a runtime adapter.

To create a project:

1. Start TIBCO Designer by executing the following commands, depending on your operating system.
   - On Microsoft Windows, select Start > All Programs > TIBCO > TIBCO Designer version_number > Designer.
   - On UNIX, from a command window, run TIBCO_HOME/designer/version_number/bin/designer.

2. Click the New Empty Project button in the TIBCO Designer dialog, and the Save Project dialog appears.

   Figure 5  Creating a New Empty Project in TIBCO Designer

3. In the Save Project dialog, click the Browse button for the Project Directory field. Create a directory to store the project: C:\tibco\projects\adswiftnet, then click the OK button. The project will be saved in the adswiftnet directory.
**Figure 6**  Saving the New Empty Project

![Image of TIBCO ActiveMatrix Adapter for SWIFTNet User's Guide](image1.png)

**Figure 7**  TIBCO Designer GUI with the adswiftnet project defined.

**Figure 7**  TIBCO Designer Panels

![Image of TIBCO ActiveMatrix Adapter for SWIFTNet User's Guide](image2.png)
Configuring an Adapter Instance

After creating a project in TIBCO Designer, you need to create an adapter instance and configure it to access the SWIFTNet application server. See Chapter 3, Adapter Instance Options, on page 19 for details.
Configuring the Services

After configuring an adapter instance, select one or multiple adapter services for the instance. The following sections describe the services and fields that are available to the adapter.

- Configuring a Request-Response Service
- Configuring the Request-Response Invocation Service

Configuring a Request-Response Service

In the Project panel, expand the `SWIFTNetAdapterConfiguration` icon, then click the `Adapter Services` directory, and drag the `Request-Response Service` icon from the Palette panel to the Design panel.

For more information in the fields that can be configured, see Request-Response Service and Request-Response Invocation Service on page 32.

Configuring the Request-Response Invocation Service

In the Project panel, expand the `SWIFTNetAdapterConfiguration` icon, then click the `Adapter Services` directory, and drag the `Request-Response Invocation Service` icon from the Palette panel to the Design panel.

For more information on the fields that can be configured, see Request-Response Service and Request-Response Invocation Service on page 32.
Exporting the Project

You must export the project to a local repository:

1. Select Project > Export Full Project from the Menu. The Export Project dialog appears.

2. Type adswiftnet in the Project Name field. Click the Browse button in the Dir Name field. Create a directory to store the project C:\tibco\adapter\adswift\version_num\examples\projects.

3. Click the OK button, then click the Yes button in the pop-up Create Project? dialog.

4. Select Project > Exit from the Menu.
Deploying and Testing the Adapter

After configuring the adapter instances, you must create a properties file for each adapter instance.

**Task A Deploying the Adapter**

To create the properties files:

1. Open a command window and change directory to the adapter bin directory:
   ```
   cd TIB_ADSWIFT_HOME\bin
   ```

2. Copy the `adraha.tra` file to a new text file named `server.tra`.
   ```
   copy adraha.tra server.tra
   ```

3. Using a text editor to open the `server.tra` file and change the following properties:
   - Change `tibco.repourl repo url` to:
     ```
     tibco.repourl TIB_ADSWIFT_HOME/examples/projects/adswiftnet.dat
     ```
   - Change `tibco.configurl config url` to:
     ```
     TIBCO.configurl RequestResponseInvocationService
     ```
   - Change: `application.args adswift -system:propFile TIB_ADSWIFT_HOME/bin/adraha.tra` to:
     ```
     application.args adswift -system:propFile TIB_ADSWIFT_HOME/bin/server.tra
     ```

4. Save and close the `server.tra` file, then copy the file to a new text file named `client.tra`.
   ```
   copy server.tra client.tra
   ```

5. Using a text editor to open the `client.tra` file, and change the following properties:
   - Change `tibco.configurl config url` to:
     ```
     TIBCO.configurl RequestResponseService
     ```
   - Change: `application.args adswift -system:propFile TIB_ADSWIFT_HOME/bin/adraha.tra` to:
     ```
     application.args adswift -system:propFile TIB_ADSWIFT_HOME/bin/server.tra
     ```

6. Save and close the `client.tra` file, then exit the text editor.
Task B  Start the Adapter

1. Open two command windows.
2. In a command window, start the server from the adapter bin directory:
   
   adraha --run --propFile server.tra

3. In the second command window, start the client from the adapter bin directory:
   
   adraha --run --propFile client.tra

Task C  Stop the Adapter

CTRL+C must NEVER be used to shut down the adapter as it could lead to a potential loss or duplication of messages.

To stop the adapter, in a command window, send a message on the ExitSubscriber subject for the server and client. For example:

   tibrvsend domain.adswiftnet.raha.exit exit
Chapter 3  

**Adapter Instance Options**

This chapter explains how to create an adapter instance and assign it services by configuring the standard settings. All configuration tasks are performed in TIBCO Designer and the information is stored in a project that is later used by the runtime adapter.

**Topics**

- **Overview of an Adapter Instance, page 20**
- **Adapter Instance Fields, page 21**
Overview of an Adapter Instance

An adapter instance tab in TIBCO Designer allows you to create, design, and run an adapter.

Configuration Tasks

Use the following steps to create and configure an adapter service:

1. Start TIBCO Designer and open the default project provided at installation. See TIBCO Designer User’s Guide for details.

2. Drag the SWIFTNetAdapterConfiguration icon from the Palette panel to the Design panel. This creates an adapter of the same name.

3. Define the adapter instance by assigning a new name and optionally changing the logging options. See Logging Tab on page 23 for details.

4. When the instance is ready to deploy, define the runtime connection options for the instance. See Runtime Connection Tab on page 22.

5. Add a service to the adapter instance (double-click the adapter instance first, then double-click the Adapter Services directory) by dragging the service icon from the Palettes panel to the Design panel.

6. In the Configuration tab, set the combination of options required for your service. For more information about the configuration settings for Request-Response service and Request-Response Invocation service, see Request-Response Service and Request-Response Invocation Service on page 32.

7. Save the project as a server repository project, and exit TIBCO Designer.

After configuring the adapter, you must create the runtime adapter property file and add the project name and adapter instance name.
Adapter Instance Fields

You can configure an adapter instance in the following tabs:

- **Configuration Tab, page 21**
- **Runtime Connection Tab, page 22**
- **Logging Tab, page 23**
- **Startup Tab, page 26**

**Configuration Tab**

After dragging the SWIFTNetAdapterConfiguration icon from the Palettes panel into the Design panel, the Configuration tab is selected by default. Table 2 lists and explains the contents in the Configuration tab.

*Table 2  Adapter Instance Configuration Tab*

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Instance Name</td>
<td>The name of the Adapter Service Engine instance. Use the default name or replace it with a name of your choice. See Guidelines for Choosing an Instance Name for more information.</td>
</tr>
<tr>
<td>Description</td>
<td>Provide information about the adapter instance stored in the project.</td>
</tr>
<tr>
<td>Version</td>
<td>The version string indicates the TIBCO ActiveEnterprise (AE) format in which the adapter instance is saved.</td>
</tr>
<tr>
<td>Message Filter</td>
<td>Specify a message filter. When you configure a message filter resource for use with the adapter, the plug-in allows you to manipulate incoming and outgoing data before sending it on the network, or handing it to the target application. See TIBCO Adapter SDK Programmer’s Guide for information about writing a message filter.</td>
</tr>
<tr>
<td>RA Config File Name</td>
<td>The name of the Remote API (RA) Configuration file that the adapter must use. It exists on the computer on which the adapter instance runs. The path for the file is specified by the SWNET_CFG_PATH environment variable on the host where the adapter instance will run. In the Runtime Connection tab, it provides the termination subject for stopping the adapter. The contents of a sample configuration file are shown below: HostName = 192.168.134.251 PortNumber = 48002 FtlaPortNumber = 48003 SSLMode = false</td>
</tr>
</tbody>
</table>
Guidelines for Choosing an Instance Name

Use the default name or replace it with a name of your choice.

- An instance name must use alphanumeric characters, and the length of the name must be less than 80 characters. An underscore (_) character can be used in an instance name, but the space character cannot be used.
- An instance name cannot use global variables.
- An instance name must be unique with respect to other adapter instances for the same adapter in the project. The same instance name can be used to name an adapter instance for a different adapter in the same project. For example, a SWIFT adapter instance named TEST and a Siebel adapter instance named TEST can coexist in the same project.
- Each instance name must be unique per adapter within a project even if each instance is defined in a different directory. That is, configuring same-named adapter instances in different directories will not make their names unique.

When you create an adapter instance, the palette automatically creates several resources for it. The names of these resources derive from the name of the instance they belong to. Changing the adapter instance name results in an automatic regeneration of the resources names. If you manually modify any resource name, that particular name will not be automatically regenerated next time you rename the adapter instance.

Runtime Connection Tab

Table 3 lists and explains all contents in the Runtime Connection tab.

Table 3  Adapter Instance Runtime Connection Tab

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAG Reconnect Attempt</td>
<td>Specify the number of times the adapter retries connecting to SWIFT Alliance Gateway before stopping. This is an integer field.</td>
</tr>
<tr>
<td>SAG Reconnect Attempt Interval (milliseconds)</td>
<td>Specify the time between two reconnection attempts in milliseconds.</td>
</tr>
<tr>
<td>Termination Subject or Topic</td>
<td>A message sent on the termination subject (if Rendezvous is the transport) or topic (if JMS is the transport) stops the adapter. In most cases, you should use the default value.</td>
</tr>
</tbody>
</table>
Table 3  Adapter Instance Runtime Connection Tab (Cont’d)

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of ActiveEnterprise Threads Per Session</td>
<td>Specify the number of threads to use in the thread pool for servicing ActiveEnterprise requests to the adapter.</td>
</tr>
<tr>
<td>ActiveEnterprise Invocation Timeout (milliseconds)</td>
<td>Specify the time the adapter service will wait to receive response from TIBCO ActiveEnterprise before throwing a timeout exception.</td>
</tr>
<tr>
<td>Maximum Number of Reconnect Attempts</td>
<td>Specify the total number of reconnection attempts before the runtime adapter or adapter service is stopped. A value of –1 means reconnection attempts will continue indefinitely.</td>
</tr>
<tr>
<td>Number of Reconnect Attempts Before Suspending Impacted Service(s)</td>
<td>Specify the number of reconnection attempts before suspending a runtime adapter or adapter service. The value of the Adapter Termination Criteria field determines whether an adapter or an adapter service is stopped.</td>
</tr>
<tr>
<td>Interval between Reconnect Attempts (milliseconds)</td>
<td>Specify the time interval in milliseconds between each reconnection attempt.</td>
</tr>
</tbody>
</table>

Logging Tab

Use these settings to configure a log file or log sinks, including which types of trace messages you want to log and where they are sent. Table 3 lists and explains all contents in the Logging tab.

Table 4  Adapter Instance Logging Tab

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
</table>
| Use Advanced Logging      | When Use Advanced Logging is unchecked (the default), you can set two standard output destinations (sinks) for trace messages and set the tracing level for the selected roles. When Use Advanced Logging is checked, you have complete control over selecting the destinations and associating desired roles with each of the destinations. The following are the sink types available:  
  • File  
  • Hawk  
  • Network  
  • STDIO  
  See Creating Log Sinks for more information |
Creating Log Sinks

When you check the Use Advanced Logging checkbox, you configure log sinks using icons in the TIBCO Designer project panel. This gives you complete control over selecting the destinations and associating desired roles with each of the destinations.

1. Check the Use Advanced Logging checkbox in the Logging tab, then click the Apply button.
2. In the Project panel, select the SWIFTNetAdapterConfiguration icon, then select Log Sinks directory from the Advanced directory.
3. Drag the Generic Log Sink icon from the Palette panel into the Design panel. Select the sink type from the Sink Type drop-down list in the Configuration panel, then click the Apply button.

Table 4  Adapter Instance Logging Tab (Cont’d)

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Log to Standard I/O</td>
<td>(stdioSink only) When it is checked, trace messages are displayed in the command prompt window where the adapter is started. When it is unchecked, trace messages are not displayed in the window.</td>
</tr>
<tr>
<td>Log File</td>
<td>Specify the name of the log file (log sink) to which trace messages are written. Global variables can be used to specify the location of the log file. See Using Global Variables on page 51 for more information. The available roles of Log File are Info, Debug, Warning, and Error messages. The trace message generated depends on the roles selected. Turning on the roles can affect the performance of the adapter. Therefore, it is recommended that you turn on the required roles only.</td>
</tr>
<tr>
<td>Log Info/Debug/Warning/Error Messages</td>
<td>Trace messages of the selected level(s) are collected in the named log sink. You can configure what levels of trace messages you want to log, and where trace messages are sent. There are three types of logs (log sinks) that you can configure to hold trace messages, information, warning and error. The fourth level of trace messages, debug, is reserved and should not be enabled unless requested by the TIBCO Product Support Group. This option writes a lot of information to the log file and significantly reduces the speed of the adapter.</td>
</tr>
</tbody>
</table>
4. With the desired log sink icon selected in the Design panel, fill in the fields in the Configuration tab. You can also change the name and enter a description for each sink by right-clicking on the sink icon in the project panel.

   — When File and STDIO sinks are created from the generic log sink, more further configuration options are displayed. For the File sink, by the default, the file limit and the file count are specified as 30000 bytes and 3, and the append mode checkbox is checked. For the STDIO sink, the option to write to stdout or stderr can be selected. By default, stdout radio button is selected.

   — The Hawk sink is created and used by the adapter for monitoring purposes. It uses the hawk session to send tracing messages to the TIBCO Hawk monitor or display. For details on Hawk sessions, see Using Global Variables on page 51. The configuration for the Hawk sink involves specifying the MicroAgent Name that must be specified in the Configuration panel.

   — The Network sink is used to publish tracing messages on TIBCO Rendezvous. The configuration for the network sink involves specifying the session, and the subject on which the trace messages need to be published.
Startup Tab

This tab displays the startup behaviors. Table 5 lists and explains all contents in the Startup tab.

Table 5  Adapter Instance Startup Tab

<table>
<thead>
<tr>
<th>Items</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Show Startup Banner</td>
<td>When it is checked, the adapter’s startup banner displays the runtime adapter version, the infrastructure version on which the adapter is built, and the copyright information in the console window from which the adapter is started.</td>
</tr>
<tr>
<td>Metadata Search URL</td>
<td>The field specifies the location where the adapter searches for base schemas. The adapter searches for any schema that has been defined and saved at this location, and that should be loaded at startup.</td>
</tr>
</tbody>
</table>
Chapter 4  Adapter Services Options

After configuring an adapter instance, select one or multiple adapter services for the instance. The following sections describe the services and fields that are available to the adapter.

Topics

- Transport Specific Service Options, page 28
- Request-Response Service and Request-Response Invocation Service, page 32
Transport Specific Service Options

The transport type (Rendezvous or JMS) you select for the runtime adapter determines transport specific service options such as quality of service, delivery mode, wire format, and so on. Only options that are compatible with a service’s transport type will be available.

This section explains transport type and some of the transport specific service configuration options, as follows:

- Transport Type, page 28
- Quality of Service, page 29
- Wire Format, page 29
- Connection Factory Type, page 30
- Delivery Mode, page 30

Transport Type

There are two transport types available for the runtime adapter, namely JMS and Rendezvous. After selecting the transport type, you can configure the transport specific options.

The transport can be configured to use a trusted store and identity resource for use in SSL (Secure Sockets Layer) configurations. TIBCO Rendezvous session and JMS session have an SSL configuration field which uses a dialog to perform SSL configuration.

To enable and configure SSL:

1. In the Project panel, select the adapter instance.
2. Expand Advanced > Sessions, then select the DefaultRVCMSession or DefaultJmsTopicSession item.
3. Check the Use SSL? checkbox in the Configuration tab, then click the Configure SSL... button to do further configuration.

The SSL configuration options are explained in the online help associated with the session dialog.
Quality of Service

This drop-down list appears only when TIBCO Rendezvous is selected as the transport type in the Configuration tab. There are three types of services that can be selected:

- **Certified**
  Guarantees that every certified message reaches its intended recipient in the order sent. The message can be sent across network boundaries, and if a network fails, delivery attempts continue until delivery succeeds or until the message’s time limit expires. This is often called certified message delivery.

  If certified message delivery is used, data is stored in a ledger file. The size of the ledger depends on several factors, the most important of which is the retention rate of stored data. That is, the ledger grows fastest in response to the cumulative length of undeliverable messages. You must ensure that sufficient disk space is available for the expected size of the ledger.

- **Reliable**
  Ensures that each multicast or broadcast message is received as long as the physical network and packet recipients are working, and that the loss of a message is detected. This choice can compensate for brief network failures because it can retransmit a message on request if the first attempt failed. This choice is appropriate when message delivery is expected but some loss can be tolerated.

- **Distributed Queue**
  Indicates that load balancing is enabled. An RVCMQ Session allows applications to use distributed queues for certified delivery, to any number of listeners using queuing member sessions that act together to process inbound task messages.

  See *TIBCO Rendezvous Concepts* for details about quality of service.

**Wire Format**

Services must use the same wire format to exchange data.

- **XML Message (JMS only)**
  The XML Message wire format conforms to specifically constructed, and fully compliant XML Schema (XSD) based on the existing definition of the ActiveEnterprise schema.

- **ActiveEnterprise Message (TIBCO Rendezvous only)**
  Control information for validation is sent in the message. If no control information is included, an exception is returned to the subscriber.
ActiveEnterprise standard wire format provides class information and packing rules for the TIBCO Adapter SDK set of data types. This format allows ActiveEnterprise components to perform extra validation on messages sent or received.

See TIBCO Adapter SDK Programmer’s Guide for details about the control information generated and sent with TIBCO ActiveEnterprise messages.

Connection Factory Type

This drop-down list appears only when JMS is selected as the transport type in the Configuration tab. There are two types of services that can be selected:

- **Topic**
  
  A message published to a topic is broadcast to one or more subscribers. All messages published to the topic are received by all services that have subscribed to the topic. This messaging model is known as publish-subscribe.

- **Queue**
  
  One and only one receiver consumes a message sent to a queue. Each message has only one receiver though multiple receivers may connect to the queue. The first receiver that accesses the queue gets the message. The other receivers do not. This messaging model is known as point-to-point.

Delivery Mode

This drop-down list appears only when JMS is selected as the transport type in the Configuration tab.

- For Request-Response Invocation service, the delivery modes are:
  
  - **Persistent**
    
    In general, a message marked persistent will be available to a JMS client even if TIBCO Enterprise Message Service server goes down. Persistent messages are held in secondary storage in the server and have guaranteed delivery when sent to a topic that has durable subscribers. (If a topic has no durable subscribers, there are no subscribers that need messages resent in the event of a server failure and therefore messages do not need to be saved.) Performance is improved because disk I/O is not required.
  
  - **Non-persistent**
    
    A message marked non persistent will not be available to a JMS client if the TIBCO Enterprise Message Service server goes down. These messages are never written to persistent storage.
• For Request-Response service, the delivery modes are:
  
  — **Durable**
  
  Indicates that the service is registered with the EMS server. Messages sent to a durable Subscription service are held by the EMS server until they are consumed by the service. The service can be down and expect to receive its messages when it comes back up.
  
  — **Non-durable**
  
  Indicates that the service is not registered with the EMS server. Messages sent to a non-durable Subscription service are not held by the EMS server. If the service is down, it will not receive the messages that arrived at the EMS server while the service is down.
Request-Response Service and Request-Response Invocation Service

To configure a client, you must use the Request-Response service. To configure a server, you must use the Request-Response Invocation service. The following tabs are available for both services:

- Configuration Tab, page 32
- Transport Tab, page 33
- Schema Tab, page 35
- SWIFTNet Connection Tab, page 36

Configuration Tab

You can specify a name and select the transport type for a Request-Response service or a Request-Response Invocation service in this tab.

Table 6  Configuration Tab of a Request-Response Service or a Request-Response Invocation Service

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>You can use the default name or replace it with a name of your choice.</td>
</tr>
<tr>
<td></td>
<td>• An instance name must use alphanumeric characters, and the length of the</td>
</tr>
<tr>
<td></td>
<td>name must be less than 80 characters. An underscore (_) character can be</td>
</tr>
<tr>
<td></td>
<td>used in an instance name, but the space character cannot be used.</td>
</tr>
<tr>
<td></td>
<td>• An instance name cannot use global variables.</td>
</tr>
<tr>
<td>Description</td>
<td>Provide additional information about the Request-Response service or</td>
</tr>
<tr>
<td></td>
<td>Request-Response Invocation service being configured.</td>
</tr>
<tr>
<td>Transport Type</td>
<td>Select the transport type from the drop-down list, to be used by the runtime</td>
</tr>
<tr>
<td></td>
<td>adapter, JMS or Rendezvous. After selecting the transport, the transport-</td>
</tr>
<tr>
<td></td>
<td>specific configuration fields are displayed.</td>
</tr>
<tr>
<td></td>
<td>The transport can be configured to use a trusted store and identity resource</td>
</tr>
<tr>
<td></td>
<td>for use in SSL (Secure Sockets Layer) configurations. TIBCO Rendezvous</td>
</tr>
<tr>
<td></td>
<td>sessions and JMS topics have an SSL configuration field which uses a dialog</td>
</tr>
<tr>
<td></td>
<td>to perform SSL configuration. See Transport Type for enabling and</td>
</tr>
<tr>
<td></td>
<td>configuring the SSL.</td>
</tr>
</tbody>
</table>
### Transport Tab

Message Transport options can be set for a Request-Response service or a Request-Response Invocation service depending on the transport type selected in the Configuration tab.

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>When TIBCO Rendezvous is selected as the transport type, the following options are available.</strong></td>
<td></td>
</tr>
<tr>
<td>Message Subject</td>
<td>By default, a service uses a message subject that is generated using the Domain and Deployment global variables, the adapter acronym, the adapter instance name, and the service name. If you use this default subject, make sure the values for Domain and Deployment are not empty. You can type a TIBCO Rendezvous subject name different from the default in this field. See TIBCO Rendezvous Concepts for information about specifying subject names.</td>
</tr>
<tr>
<td>Quality of Service</td>
<td>Select the level of service that determines how messages are sent. For Request-Response service, there are three options can be selected:</td>
</tr>
<tr>
<td></td>
<td>• Reliable</td>
</tr>
<tr>
<td></td>
<td>• Certified</td>
</tr>
<tr>
<td></td>
<td>• Distributed Queue</td>
</tr>
<tr>
<td></td>
<td>• For Request-Response Invocation service, there are two options can be selected:</td>
</tr>
<tr>
<td></td>
<td>• Reliable</td>
</tr>
<tr>
<td></td>
<td>• Certified</td>
</tr>
<tr>
<td></td>
<td>See Quality of Service for a description of these options.</td>
</tr>
<tr>
<td>Wire Format</td>
<td>The wire format in which messages are to be sent. See Wire Format for a description of the format.</td>
</tr>
<tr>
<td></td>
<td>• ActiveEnterprise Message</td>
</tr>
<tr>
<td>Session Reference</td>
<td>When you create a service, TIBCO Designer creates a corresponding session resource in the Advanced &gt; Sessions directory. The session resource displays in this field. If you have explicitly created a custom session of the same type, you can click the Browse icon to replace the autocrat session. Changing the session for a service is not recommended.</td>
</tr>
</tbody>
</table>
Table 7  Transport Tab of a Request-Response Service and a Request-Response Invocation Service (Cont’d)

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Endpoint</td>
<td>Specify the location in TIBCO Repository where the endpoint information will be stored. Endpoints store information, such as message subject, endpoint type (reliable publication service and certified publication service), and startup state (active and inactive). Note that it is recommended that you retain pre-configured endpoint references.</td>
</tr>
<tr>
<td>Reference</td>
<td></td>
</tr>
</tbody>
</table>

When JMS is selected as the transport type, the following options are available.

- **Destination**: By default, a service uses a dynamic destination that is generated using the Domain and Deployment global variables, the adapter acronym, the adapter instance name, and the service name. If you use this default dynamic destination, make sure the values for Domain and Deployment are not empty. You can override the default dynamic destination by specifying the static destination in this field. The static destination must be defined on the JMS server before it can be used by the runtime adapter. See TIBCO Enterprise Message Service User’s Guide for information about destination.

- **Wire Format**: The wire format in which messages are to be published. See Wire Format for a description of the format.
  - XML Message

- **Connection Factory Type**: Select one of the following:
  - Topic
  - Queue

  See Connection Factory Type for a description of connection factory type.

- **Delivery Mode**: The delivery mode for each message sending operation.
  - For Request-Response service, select the following two options:
    - Durable
    - Non-Durable
  - For Request-Response Invocation service, select the following two options:
    - Persistent
    - Non-Persistent

  See Delivery Mode for a description of each mode.
Schema Tab

Schemas for the services are automatically configured depending on the configuration in the SWIFTNet Connection tab. Schemas are loaded automatically under AESchemas/ae/adraha.

Class Reference

Class reference points to the schema automatically configured for the underlined service. The schema is located under AESchemas/ae/adraha.
## SWIFTNet Connection Tab

Table 8 lists and explains all contents in the SWIFTNet Connection tab.

### Table 8  A Request-Response Service or a Request-Response Invocation Service

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Message Partner Name</td>
<td>Application Interface name on SWIFT Alliance Gateway (SAG). Make sure that the Application Interface is correctly configured for working with Remote API Host Adapter. Ensure that the Application Interface is configured in the Client mode. If you are working with SAG 6.x or later, make sure the Local Authentication setting for this message partner is disabled.</td>
</tr>
<tr>
<td>Certificate Name</td>
<td>The name of the security certificate used for exchanging messages with SWIFT Alliance Gateway. Note that requests sent on this service will have the requester field as the Destination Name for this certificate.</td>
</tr>
<tr>
<td>Certificate Password</td>
<td>The password for the security certificate.</td>
</tr>
<tr>
<td>Service Name</td>
<td>The SWIFTNet service for which this client is configured.</td>
</tr>
<tr>
<td>NR Indicator</td>
<td>(Request-Response service only) Non Repudiation setting for the service. If SWIFTNet mandates the Non Repudiation setting for the service configured, then you must check this checkbox.</td>
</tr>
<tr>
<td>Message Priority</td>
<td>(Request-Response service only) Priority to be used to send the message in case it is not specified at the message level. There are two options: Normal and Urgent. Normal is the default.</td>
</tr>
</tbody>
</table>
Table 8  A Request-Response Service or a Request-Response Invocation Service (Cont’d)

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Messaging Mode</td>
<td>The messaging modes for sending or receiving the SWIFTNet InterAct messages from the adapter to SWIFT Alliance Gateway (SAG) and from SAG to the adapter. It can be one of the following:</td>
</tr>
<tr>
<td></td>
<td>• <strong>Interact Realtime</strong>—This is the default. Meant for correspondents that are online at the time of transmission. As a result, it is suited for sending individual instructions, confirmations, and reports to a few large recipients, or for messages to market infrastructures.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Interact Store-and-Forward</strong>—Meant for a large number of recipients, many of whom may not be online at the time of transmission.</td>
</tr>
<tr>
<td></td>
<td>For a Request-Response service, when this radio button is selected, the following options need to be specified:</td>
</tr>
<tr>
<td></td>
<td>— <strong>Delivery Notification Queue</strong></td>
</tr>
<tr>
<td></td>
<td>The delivery notification queue corresponds to the SWIFTNet queue in which the SWIFTNet should place the delivery notifications for the sent messages on this client. For more information on delivery notification, see the Store-and-Forward mode documentation provided by SWIFT.</td>
</tr>
<tr>
<td></td>
<td>— <strong>Positive Delivery Notification Required</strong></td>
</tr>
<tr>
<td></td>
<td>Check the <strong>Positive Delivery Notification Required</strong> checkbox to request delivery notification for successful delivery of the requests to the server.</td>
</tr>
<tr>
<td></td>
<td>For a Request-Response Invocation service, when this radio button is selected, the following options need to be specified:</td>
</tr>
<tr>
<td></td>
<td>— <strong>Receiver Queue</strong></td>
</tr>
<tr>
<td></td>
<td>The queue applies to store-and-forward transfer mode of InterAct messages only. Provide a valid queue name on the SWIFTNet that the adapter will poll for any requests coming from the queue.</td>
</tr>
<tr>
<td></td>
<td>— <strong>Delivery Notification</strong></td>
</tr>
<tr>
<td></td>
<td>Check this checkbox to propagate the delivery notifications to the applications. If this checkbox is unchecked, and the adapter detects any delivery notification in the queue, it will automatically rejects those delivery notifications.</td>
</tr>
</tbody>
</table>
Chapter 5  Deploying an Adapter Using TIBCO Administrator

This chapter provides an overview about deploying, starting, stopping, and monitoring adapter services using the TIBCO Administrator.

See the TIBCO Administrator documentation set for details about using TIBCO Administrator.

Topics

- Creating an EAR File in TIBCO Designer, page 40
- Deploying a Project, page 41
- Starting or Stopping the Adapter, page 42
- Monitoring the Adapter, page 43
Creating an EAR File in TIBCO Designer

This section introduces how to generate an Enterprise Archive file (EAR) that contains information about the adapter services to deploy.

You can deploy one or more adapter services, one or more TIBCO ActiveMatrix BusinessWorks process engines, or both.

Building an archive creates the EAR file, which you can then deploy from TIBCO Administrator. If you make changes to the business processes or adapter services included in the archive, you need to rebuild the archive. Saving the project does not affect the archive.

In TIBCO Designer, follow these steps to create an EAR:

1. Configure the adapter services for an adapter instance.
2. Drag the Enterprise Archive icon from the Palette panel to the Design panel. If there are any configured adapter services in your project, an Adapter Archive resource becomes available in the Palette panel.
3. Specify information in the Configuration tab, then click the Apply button.
4. Click the Build Archive button to create the archive file.

You can also create an adapter archive from Tools > Create Project EAR if you have an existing process or adapter instance.

See Also

See TIBCO Designer User’s Guide for more information about this procedure. The guide is available from the Designer Help menu.
Deploying a Project

Before deploying a project, the machine on which the adapter is installed must be part of a TIBCO administration domain. After you have installed the TIBCO Administration Server, any machine on which you install TIBCO Runtime Agent (required by an adapter) can be added to the administration domain. The TIBCO software installed on the machine is then visible and accessible via the TIBCO Administrator GUI.

When you deploy a project, startup scripts and other information about the different components are sent to the machines to which the components were assigned. The project data store and TIBCO Administration Server are updated with the deployed components.

To deploy a project:

1. Start TIBCO Administrator and import the EAR file into TIBCO Administrator Enterprise Edition.
2. Assign adapter archives in the EAR file to adapters installed in the administration domain and likewise assign process archives to process engines.
3. Specify startup options for each adapter service.
4. If desired, set up your deployment for fault tolerance by specifying more than one adapter service to run on different machines. The secondary services will run in standby mode until they are needed.

See Also

See the TIBCO Administrator User’s Guide for an introduction to the TIBCO administration domain and detailed information about the above steps.
Starting or Stopping the Adapter

The TIBCO Administrator Application Management module allows you to start and stop deployed applications.

Starting the Adapter

To start an adapter service from the module:

1. In the Administrator GUI left pane, expand Application Management > Application_Name > Service Instances.
2. In the Service Instance panel, check the checkbox next to the adapter service.
3. Click the Start button. The service status changes from Stopped to Starting up to Running.

Stopping the Adapter

To stop the adapter service, click the Stop button. The service status changes from Running to Shutting Down to Stopped.

See Also

Monitoring the Adapter

TIBCO Administrator offers a number of monitoring options.

- Specify alerts and TIBCO Hawk rulebases for each machine in the domain.
- Specify alerts and TIBCO Hawk rulebases for each adapter service.
- View the log for each adapter service.

See Also

See *TIBCO Administrator User’s Guide* for information about configuring the above monitoring options.
Chapter 6  

Advanced Configuration Options

This chapter explains the advanced topics, such as modifying TIBCO Rendezvous session parameters, modifying TIBCO JMS session parameters, using a revision control system, and using global variables.

Topics

- Modifying TIBCO Rendezvous Session Parameters, page 46
- Modifying TIBCO JMS Session Parameters, page 47
- Using the Adapter with a Revision Control System, page 48
- Using Global Variables, page 51
Modifying TIBCO Rendezvous Session Parameters

By default, TIBCO Runtime Agent uses predefined TIBCO Rendezvous sessions named DefaultRVSession. These sessions are configured to use the default TIBCO Rendezvous session parameters, but these parameters can be modified to fit your environment. For instance, if you are running a remote daemon, you will need to specify the Service, Network and Daemon parameters.

1. In the Project panel, expand the required adapter configuration directory, then expand the Advanced > Session directory, and click the DefaultRVSession item.

2. Click the Help for DefaultRVSession button for a description of field values.

Figure 9 TIBCO Rendezvous Session Configuration Screen
Modifying TIBCO JMS Session Parameters

When the transport type for an adapter service is changed from TIBCO Rendezvous to JMS, DefaultJMSQueueSession or DefaultJMSTopicSession is created.

1. In the Project panel, expand the required adapter configuration directory, then expand the Advanced > Sessions directory, and click the DefaultJmsTopicSession item.

2. Click the Help for DefaultJmsTopicSession icon for a description of field values.

Figure 10  TIBCO Enterprise Message Service Session Configuration Screen
Using the Adapter with a Revision Control System

TIBCO Designer supports revision control systems, such as Microsoft Visual SourceSafe and Perforce. If you are using a revision control system, you must manually add some configured resources to the revision control system, and check in the resources when completing the instance configuration.

As part of service configuration, the adapter creates schema files in root/AESchemas/ae/adraha. For example, if you configure a service in an adapter configuration "Instance1", the following files are created:

Project_root/AESchemas/ae/adraha/Instance1.aeschema  Project_root/AESchemas/ae/adraha/Instance1/businessObjects.aeschema

When the project is saved and a revision control system has been specified, the adapter displays a warning that additional files were created and should be added to the revision control system. This warning appears only when the files are created for the first time. The warning displays a Go To Resource button that helps in navigating to the resource. You should use the Multi-User > Add Resources to RCS menu command to add these files to the revision control system.

See TIBCO Designer User’s Guide for information about how to use the Multi-User feature.

Copy, Cut, Paste and Move Operations

- To successfully copy and paste a service from adapter Instance1 to Instance2, the adapter configuration and schema files for the Instance2 must be checked out.

- To successfully cut and paste a service from adapter Instance1 to Instance2, the adapter configuration and schema files for both Instance1 and Instance2 must be checked out.

- To successfully move a service from adapter Instance1 to Instance2, select the Extended Copy item and the Paste item from the Edit menu in TIBCO Designer and then delete the resource.

- To copy a resource, select the Extended Copy item from the Edit menu in TIBCO Designer.

Creating Two Adapter Instances in the Same Project by Different Users

Use the following steps as guidelines when creating two adapter instances in the same project by two users:

1. Create a new project named myTestProj.
2. Save the project using a revision control system, for example, MicroSoft Visual Source Safe.

3. Click the root directory and add the files to the revision control system. Click the **Yes** button for recursive addition.

4. Check in the project.

5. Sync the project on two machines, Machine A and Machine B.

6. Open the project on Machine A and configure Instance1. Add Instance1 and the related schema files to the revision control system. Check in the changes.

7. Open the project on Machine B and configure Instance2. Add Instance2 and the related schema files to the revision control system. Check in the Changes.

The global variables must be populated first, otherwise, you cannot configure the instances.

**Password Handling**

At design-time, the adapter uses a password to connect to the back-end application and fetch metadata. At runtime, the adapter uses a password to connect to the back-end application and interoperate with it. If you create a 4.x configuration using TIBCO Designer 5.1.2, and use the configuration against a 4.x adapter version, some special considerations are required for security.

If you plan to run the adapter locally, define the runtime password value to be a global variable. Before starting the adapter, include the runtime password as the client variable in the adapter’s .tra file and obfuscate it using the obfuscate tool. For example, if the password value is defined as `%%myPassword%%`, create a global variable named `myPassword` in the global variables section with no value and include the following entry in the adapter’s .tra file:

```
tibco.clientVar.myPassword
```

See "Obfuscating a Password in the Properties File" for details on how to use the obfuscate tool. If you plan to deploy the adapter using TIBCO Administrator, checkmark the **Service** property of the global variable in the global variables section. Before deploying the adapter, go to the Advanced tab of the adapter archive and set the password value in the Runtime Variables section.

Do not set the password to type **Password** in the global variables section for version AE Version 4.0 or AE Version 5.0 or any intermediate version.
Regeneration differences between adapters:

- Default subjects are not regenerated to reflect the new instance name when a service is moved.
- Manually changed certified messaging and certified messaging queue ledger file names are regenerated to defaults when a service is moved, or copied and pasted to a new instance.
- If a service associated with a custom session is moved, or copied and pasted, the custom session is not moved, or copied and pasted. The session is regenerated as a default session.
Using Global Variables

The variable substitution mechanism can override the global variables predefined in the project in a restricted manner. Predefined variables can be viewed and set in TIBCO Designer. Variables are specified as `%%VARNAME%%` and cannot contain any white space.

Variable substitution allows you to accomplish the following:

- Substitute string variables specified in the project at startup time.
- Locally define the value for a variable for a specific project. The local value takes precedence over any global value.
- Specify the value for a variable in a properties file. This overrides the project repository and values set in code, but not variables set on the command line.
- Enforce the predefined variables listed in Predefined Global Variables on page 53.

Variables can be used anywhere in the configuration and will be replaced by the locally defined adapter instance.

Variable Specification

The adapter can specify variables in the project during configuration using TIBCO Designer. It can also specify variables in a properties file. The properties file values overwrite values set in the project.

Specifying Variables Using TIBCO Designer

Global variables provide an easy way to set defaults for use throughout your project. There are several ways in which they can be used:

- Define a variable using TIBCO Designer, then override the value for individual applications at deployment time using TIBCO Administrator. You can also override values for predefined variables, unless the GUI does not allow you to make them settable later.
- Predefine a variable using TIBCO Designer, then override the value for individual services (for example, publication service or TIBCO ActiveMatrix BusinessWorks process) at deployment time using TIBCO Administrator. The values you specify are then used at runtime. You can also override values for predefined variables, unless the GUI does not allow you to make them settable later.
For example, you could assign the value 7474 to the predefined global variable \texttt{RvDaemon}. You can then use the variable in different sessions in your adapter. If you wish to change the TIBCO Rendezvous daemon for your adapter, you can globally set it to a different value or override it from the command line.

To use global variables in your project, follow these steps:

1. Click the \texttt{Global Variables} panel, then click the Open Advanced Editor button \(\text{\textbullet}\). You now have these choices:
   
   — To assign or change a variable value, select that region and triple-click the variable, then you can change either the variable name or the variable value. Click the \textbf{OK} button when you are done.
   
   — To add a new global variable group, click the \textbf{Add a Variable Group} button \(\text{\textbullet}\) at the bottom of the dialog box. Specify the name of the group, then click the \textbf{OK} button.
   
   — To add a global variable, click the \textbf{Add a Variable} button \(\text{\textbullet}\). A new global variable item is added to the bottom of the list. Supply the variable name and, optionally, the value. Click the \textbf{OK} button when you are done.

   The global variable is now displayed in the global variables list.

2. When you want to use the global variable in the fields of a resource, enter the variable name surrounded by \texttt{%%} on both sides.

   When the project is deployed and the configured components are run, all occurrences of the global variable name are replaced with the global variable value (unless it was overridden in a way that had higher precedence).

   A number of global variables are predefined. See \textbf{Predefined Global Variables on page 53} for information. You may add definitions of any variables you need to predefine.

**Changing Global Variable Values at Runtime**

You can change the value of a global variable when you deploy your project in TIBCO Administrator. See the section on modifying runtime variables in \textit{TIBCO Administrator User’s Guide} for more information on using TIBCO Administrator.

You can also specify values for global variables when starting a process engine on the command line. To do this, specify the following as a command line argument when starting the process engine:

\[-\texttt{tibco.clientVar.variablePathAndName value}\]
where \textit{variablePathAndName} is the name of the variable you wish to set, including the path to the variable if it is contained in a directory. \textit{value} is the value you wish to set the variable to. For example, if you have a global variable named \texttt{item1} contained in a directory named \texttt{myGroup} and you wish to set its value to 500, add the following argument to the command line when starting the process engine:

\begin{verbatim}
-tibco.clientVar.myGroup/item1 500
\end{verbatim}

\section*{Predefined Global Variables}

Table 9 lists and explains the predefined global variables. Some global variables are automatically used within the system when an adapter instance is configured.

\begin{table}[h]
\centering
\begin{tabular}{|l|p{15cm}|}
\hline
\textbf{Variable} & \textbf{Description} \\
\hline
Deployment & Defaults to the TIBCO Designer project name. This global variable is used by the system to partially define the subject name defined for a service. \\
\hline
DirLedger & Used by the system when defining the path name of the TIBCO Rendezvous certified messaging ledger file. The default is root installation directory. \\
\hline
DirTrace & Used by the system to partially create the path name for log file used by the adapter. The default is the root installation directory. \\
\hline
Domain & The default value for file-based local projects is \texttt{MyDomain}. This value for server-based projects is the domain to which the project was saved. \\
\hline
Env & Included for backward compatibility. Do not use. \\
\hline
HawkEnabled & Indicates whether TIBCO Hawk is used to monitor the adapter. True indicates that a TIBCO Hawk microagent is defined for the adapter. False indicates the microagent is not to be used. \\
\hline
JmsProviderUrl & Tells applications where the JMS daemon is located. Setting this value mostly makes sense in early stages of a project, when only one JMS daemon is used. \\
\hline
JmsSslProviderUrl & Specifies where the JMS SSL daemon is located. \\
\hline
RaConfigFile & The name of the Remote API (RA) Configuration file that the adapter must use. \\
\hline
RahaInstanceId & The ID of the SWIFTNet Adapter instance. \\
\hline
RemoteRvDaemon & Used by the system to identify the TIBCO Rendezvous routing daemon. See \textit{TIBCO Rendezvous Administration} for details about specifying the routing daemon name. \\
\hline
\end{tabular}
\caption{Predefined Global Variables}
\end{table}
Table 9  Predefined Global Variables (Cont’d)

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RvDaemon</td>
<td>Used by the system to identify the TIBCO Rendezvous daemon parameter. The parameter instructs the transport object about how and where to find the Rendezvous daemon and establish communication. The default value is 7500.</td>
</tr>
<tr>
<td>RvNetwork</td>
<td>Used by the system to identify the TIBCO Rendezvous network parameter. Every network transport communicates with other transports over a single network interface. On computers with more than one network interface, the network parameter instructs the TIBCO Rendezvous daemon to use a particular network for all outbound messages from this transport. In most cases, you can leave the default.</td>
</tr>
<tr>
<td>RvService</td>
<td>Used by the system to identify the TIBCO Rendezvous service parameter. The TIBCO Rendezvous daemon divides the network into logical partitions. Each transport communicates on a single service; a transport can communicate only with other transports on the same service. Unless you are using a non-default TIBCO Rendezvous configuration, you should leave the default (7500).</td>
</tr>
<tr>
<td>RvaHost</td>
<td>Used by the system to identify the computer on which the TIBCO Rendezvous agent (rva) runs. See TIBCO Rendezvous Administration for details about specifying the rva parameters.</td>
</tr>
<tr>
<td>RvaPort</td>
<td>Used by the system to identify the TIBCO Rendezvous agent (rva) TCP port where the agent listens for client connection requests. See TIBCO Rendezvous Administration for details about specifying the rva parameters.</td>
</tr>
<tr>
<td>TIBHawkDaemon</td>
<td>Used by the system to identify the TIBCO Hawk daemon parameter. See TIBCO Hawk Installation and Configuration manual for details about this parameter.</td>
</tr>
<tr>
<td>TIBHawkNetwork</td>
<td>Used by the system to identify the TIBCO Hawk network parameter. See TIBCO Hawk Installation and Configuration manual for details about this parameter.</td>
</tr>
<tr>
<td>TIBHawkService</td>
<td>Used by the system to identify the TIBCO service parameter. See TIBCO Hawk Installation and Configuration manual for details about this parameter.</td>
</tr>
</tbody>
</table>
Appendix A  Trace Messages

This appendix explains the trace messages that are logged to a location specified at configuration time.

Topics

- Overview of Trace Messages, page 56
- Trace Message Fields, page 57
- Status Messages, page 60
Overview of Trace Messages

Trace messages provide information about adapter activities. The messages are logged to the console where the runtime adapter was started and to a log file. Trace messages can also be redirected to the TIBCO Hawk Display application, or sent to other applications using the TIBCO Rendezvous transport.

Each trace message can include the following fields:

*Timestamp*  *Adapter Identifier*  *Role*  *Category*  *Status Code*  *Tracking Identifier*

The above fields are explained in Trace Message Fields on page 57. The following diagram shows an example trace message and calls out the fields.

```
2003 Feb 22 20:15:12:937 GMT -8
SWIFTAdapter.SWIFTAdapterConfiguration  Info [Adapter]
2003 Mar 07 15:13:36:512 GMT +5 SubCancelPO Info [System]
tracking=#MU3oTJ/WWCV1MU96j0zzwA9kzzw#
AESWFT-4003018 Adapter Initialized Successfully....
```
Trace Message Fields

Each trace message includes the following fields:

Table 10  Tracing Fields (Sheet 1 of 3)

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adapter Identifier</td>
<td>Name of the adapter that wrote the trace message. This is a combination of the adapter acronym and adapter configuration name. For example, the application identifier, adswift.publisher1 identifies a TIBCO ActiveMatrix Adapter for SWIFT service named publisher1.</td>
</tr>
</tbody>
</table>

Role

A role can be:

- **Info.** indicates normal adapter operation. No action is necessary. A tracing message tagged with Info. indicates that a significant processing step was reached and has been logged for tracking or auditing purposes. Only info. messages preceding a tracking identifier are considered significant steps.

- **Warn.** an abnormal condition was found. Processing will continue, but special attention from an administrator is recommended.

- **Error.** an unrecoverable error occurred. Depending on the error severity, the adapter may continue with the next operation or may stop altogether.

- **Debug.** a developer-defined tracing message. In normal operating conditions, debug messages should not display.

When configuring the adapter you define what roles should or should not be logged. For example, you may decide not to log Info roles to increase performance.
Table 10  Tracing Fields (Sheet 2 of 3)

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Category</td>
<td>One of the following:</td>
</tr>
<tr>
<td>Adapter</td>
<td>the adapter is processing an event.</td>
</tr>
<tr>
<td>Application</td>
<td>the adapter is interacting with SWIFT.</td>
</tr>
<tr>
<td>Configuration</td>
<td>the adapter is reading configuration information.</td>
</tr>
<tr>
<td>Database</td>
<td>the adapter is interacting with a database.</td>
</tr>
<tr>
<td>Metadata</td>
<td>the adapter is retrieving metadata from SWIFT.</td>
</tr>
<tr>
<td>Palette</td>
<td>the adapter is interacting with the palette.</td>
</tr>
<tr>
<td>Publication Service</td>
<td>the Publication service is reporting this trace message.</td>
</tr>
<tr>
<td>Request-Response Client Service</td>
<td>the Request-Response Invocation service is reporting this trace message.</td>
</tr>
<tr>
<td>Request-Response Server</td>
<td>the Request-Response service is reporting this trace message.</td>
</tr>
<tr>
<td>Shutdown</td>
<td>the adapter is shutting down.</td>
</tr>
<tr>
<td>Startup</td>
<td>the adapter is starting.</td>
</tr>
<tr>
<td>Subscription Service</td>
<td>the Subscription service is reporting this trace message.</td>
</tr>
<tr>
<td>System</td>
<td>this category is not linked to a specific event process.</td>
</tr>
<tr>
<td>TibRvComm</td>
<td>the adapter is communicating with TIBCO Rendezvous.</td>
</tr>
<tr>
<td>XML</td>
<td>the adapter is parsing XML documents.</td>
</tr>
</tbody>
</table>

| Status Code     | Unique code for the message and description. Status codes are identified by a unique number and description. If a trace message includes an error or warn role, the status code documentation includes a resolution. |
Table 10  Tracing Fields (Sheet 3 of 3)

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tracking Identifier</td>
<td>A unique identifier that is &quot;stamped&quot; on each message by the originating adapter. The tracking identifier remains in effect from a message’s beginning to its completion as it is exchanged by TIBCO applications. If the adapter is the termination point of the message, the tracking identifier is not displayed in the trace message. You cannot modify the tracking identifier format or configure what information is displayed.</td>
</tr>
<tr>
<td>Application Information</td>
<td>Application-specific information added to the tracking info to trace the message back to its source. Set initially by the originating adapter and carried forward. It is augmented by each intermediate component.</td>
</tr>
</tbody>
</table>
Status Messages

In environments where multiple applications are used simultaneously, the possible status of messages increases as well. This section lists the various messages in numerical order.

Resolutions are provided wherever possible for error and warning messages. If there is no resolution provided, or if you need additional help, contact TIBCO Support at http://support.tibco.com.

Table 11 Status Messages (Sheet 1 of 3)

<table>
<thead>
<tr>
<th>Message</th>
<th>Role</th>
<th>Category</th>
<th>Resolution</th>
</tr>
</thead>
<tbody>
<tr>
<td>AESWIFTNET-350001: Error during create resource: Read-Only/The resource type you tried to drop, %1, could not be created</td>
<td>errorRole Palette</td>
<td>Ensure that the file is checked out. You can select the resource to be checked out by clicking the Go To Resource button</td>
<td></td>
</tr>
<tr>
<td>AESWIFTNET-350002: Unable to Rename Resource/Designer was not able update the resources name to %1</td>
<td>errorRole Configuration</td>
<td>There are several possible reasons. 1. A name must be non-null with non-illegal characters. It is not duplicated in the group, and not be a whole number enclosed in the square brackets ([123]). 2. Spaces are not allowed in the adapter instance name field. Note: See documentation for a list of illegal characters.</td>
<td></td>
</tr>
<tr>
<td>AESWIFTNET-350003: Missing value for the mandatory field : %1</td>
<td>errorRole Palette</td>
<td>This field is a mandatory field for this service. Ensure that it has been correctly filled in.</td>
<td></td>
</tr>
<tr>
<td>AESWIFTNET-350004: Invalid File Path for field : %1</td>
<td>errorRole Palette</td>
<td>Verify the path entered for the file.</td>
<td></td>
</tr>
<tr>
<td>AESWIFTNET-350005: Invalid value entered for numeric field : %1, Permissible values are -1, 0 and 1 to %2</td>
<td>errorRole Palette</td>
<td>The value entered in this field should be a numeric value. Ensure that number of reconnect attempts should be less than the maximum number of reconnect attempts.</td>
<td></td>
</tr>
</tbody>
</table>
Table 11  Status Messages (Sheet 2 of 3)

<table>
<thead>
<tr>
<th>Message</th>
<th>Role</th>
<th>Category</th>
<th>Resolution</th>
</tr>
</thead>
<tbody>
<tr>
<td>AESWIFTNET-350006:</td>
<td>errorRole</td>
<td>Palette</td>
<td>The value entered in this field should be a numeric value.</td>
</tr>
<tr>
<td>AESWIFTNET-050000:</td>
<td>errorRole</td>
<td>Adapter</td>
<td>Ensure that a valid RV session has been configured in the repository.</td>
</tr>
<tr>
<td>AESWIFTNET-050001:</td>
<td>errorRole</td>
<td>Adapter</td>
<td>Ensure that a valid RVCM session has been configured in the repository.</td>
</tr>
<tr>
<td>AESWIFTNET-050002:</td>
<td>errorRole</td>
<td>Adapter</td>
<td>Ensure that a valid RVCMQ session has been configured in the repository.</td>
</tr>
</tbody>
</table>
Table 11  Status Messages (Sheet 3 of 3)

<table>
<thead>
<tr>
<th>Message</th>
<th>Role</th>
<th>Category</th>
<th>Resolution</th>
</tr>
</thead>
<tbody>
<tr>
<td>AESWIFTNET-050004: Error retrieving JMS session</td>
<td>errorRole</td>
<td>Configuration</td>
<td>Ensure that a valid JMS session has been configured in the repository.</td>
</tr>
<tr>
<td>AESWIFTNET-050005: Mandatory parameter responder value is null or empty</td>
<td>errorRole</td>
<td>Adapter</td>
<td>Ensure that non-null value is set for parameter responder.</td>
</tr>
<tr>
<td>AESWIFTNET-050006: Mandatory parameter payload value is null or empty</td>
<td>errorRole</td>
<td>Adapter</td>
<td>Ensure that non-null value is set for parameter payload.</td>
</tr>
<tr>
<td>AESWIFTNET-050007: Mandatory parameter requestType value is null or empty</td>
<td>errorRole</td>
<td>Adapter</td>
<td>Ensure that non-null value is set for parameter requestType.</td>
</tr>
<tr>
<td>AESWIFTNET-050008: Exception occurred while sending request to SAG</td>
<td>errorRole</td>
<td>Adapter</td>
<td>Try to send the request again.</td>
</tr>
<tr>
<td>AESWIFTNET-050009: Connection to SAG is down</td>
<td>errorRole</td>
<td>Adapter</td>
<td>Try to send the request again.</td>
</tr>
<tr>
<td>AESWIFTNET-050010: Adapter received an invalid AE request</td>
<td>errorRole</td>
<td>Adapter</td>
<td>Verify the ActiveEnterprise request is proper and try to send the request again.</td>
</tr>
</tbody>
</table>
Appendix B  Usage Scenarios and Supported Schemas

This appendix explains some usage scenarios and the schemas used by the adapter. Schemas are selected based on the mode (Real Time and Store-and-Forward) and the additional configuration related to the delivery notifications.

Topics

- InterAct Real Time Client, page 64
- InterAct Real Time Server, page 66
- InterAct Store-and-Forward Client, page 68
- InterAct Store-and-Forward Server, page 69
- InterAct Store-and-Forward Server To Receive Delivery Notifications, page 70
InterAct Real Time Client

To configure the InterAct real time client, follow these steps:

1. Users configure a SWIFTNet adapter instance with an appropriate Remote API configuration file name.

2. In the SWIFTNet adapter instance mentioned in step 1, users configure a Request-Response service for Real Time mode. Users need to configure all the required fields.

3. Users configure the Invoke an Adapter Request-Response Service in a TIBCO ActiveMatrix BusinessWorks process to invoke the service configured in step 2.

4. A message of a Real Time service is built in TIBCO ActiveMatrix BusinessWorks.

5. The message built in the step 4 is given as input to the activity in step 3 (the Invoke an Adapter Request-Response Service activity) with other parameters, such as request type, priority, and so on.

6. The adapter receives an invoke request from the Invoke an Adapter Request-Response Service activity. It will send the request to the SWIFT Alliance Gateway (SAG), then the SAG will send it to the receiver. The receiver sends a response to the SAG, and the adapter gets the response from the SAG, then processes it. The payload field in the request will be published to the Invoke an Adapter Request-Response Service.

7. The Invoke an adapter Request-Response service activity configured in step 3 gets the response mentioned in step 6.

8. The response is processed in TIBCO ActiveMatrix BusinessWorks.

The following schemas are used in the adapter:

- Request Schema—InterActRealTimeClient class

  Request Schema contains the parameters that are sent to the adapter to generate a primitive request that should be sent to SWIFT Alliance Gateway.

- Operation Name—send

- Response Schema—InteractRTC1ientResponse

  The Response schema is what the application gets in return of the Request Response Service invocation.

These schemas are automatically loaded under AESchemas/ae/adraha.

Figure 11 is described in details:
Figure 11   InterAct Real Time Client

Note: AE or XML Message contains XML Payload that conforms to the SWIFTNet solution that is being implemented.
InterAct Real Time Server

To configure the InterAct real time server, follow these steps:

1. Users configure a SWIFTNet adapter instance with the appropriate Remote API configuration file name.

2. In the SWIFTNet adapter instance mentioned in step 1, users configure a Request-Response Invocation service for the Real Time mode. Users need to configure all the required fields.

3. Users configure an Adapter Request-Response Server in a TIBCO ActiveMatrix BusinessWorks process to receive the server requests coming from the service configured in step 2.

4. The adapter receives a request message from SWIFT Alliance Gateway (SAG) that is sent by some other parties on SWIFTNet. The adapter processes the request and publishes the payload field along with the other details to TIBCO environment.

5. The Blocking Activity configured in step 3, which was waiting for requests, receives the published request. The request message is processed in TIBCO ActiveMatrix BusinessWorks and the response is generated.


7. The adapter receives the response, then sends it to SAG.

The following schemas are used in the adapter:

- **Request Schema**—`InterActRealTimeServer` class
  The Request Schema contains the parameters that the application gets from the adapter as a Server request.

- **Operation Name**—`receive`

- **Response Schema**—`RealTimeResponse`
  The Response Schema is what the application should send in return to the adapter, then the adapter will generate a Primitive Response to send to SAG.
Figure 12  InterAct Real Time Server

Note: AE or XML Message contains "XML Payload" that conforms to the SWIFTNet solution that is being implemented.
To configure the InterAct store-and-forward client, follow these steps:

1. Users configure a SWIFTNet adapter instance with the appropriate Remote API configuration file name.

2. In the SWIFTNet adapter instance mentioned in step 1, users configure a Request-Response service for the Store-and-Forward (SnF) mode. Users need to configure all the required fields.

3. Users configure the Invoke and Adapter Request-Response Service in a TIBCO ActiveMatrix BusinessWorks process to invoke the service configured in step 2.

4. A message of a Store-and-Forward service (for example, EnI) is built in TIBCO ActiveMatrix BusinessWorks.

5. The message built in step 4 is given as input to the Invoke and Adapter Request-Response Service activity with other parameters, such as request type, priority, and so on.

6. The adapter receives the invoke request from the Invoke and Adapter Request-Response Service activity. It sends the request to SWIFT Alliance Gateway to place it on the receivers queue. SAG returns a Success/Failure status depicting whether the message been put on the receivers queue or not. The adapter publishes this response to the Invoke an Adapter Request-Response Service activity. In case of a "Success", it also publishes the SwiftRef for the messages received from SAG.

7. The Invoke and Adapter Request-Response Service configured in step 4 gets the response.

8. The response is processed in TIBCO ActiveMatrix BusinessWorks. The SwiftRef is stored for future reconciliation of delivery notifications.

The following schemas are used in the adapter:

- **Request Schema**—InterActSnFClient class
- **Operation Name**—send
- **Response Schema**—SnFResponse
To configure the InterAct store-and-forward server, follow these steps:

1. Users configure a SWIFTNet adapter instance with the appropriate Remote API configuration file name.

2. In the SWIFTNet adapter instance mentioned in step 1, users configure a Request-Response Invocation Service for Store-and-Forward (SnF) mode. Users need to configure all the required fields along with the receiver queue from where the adapter should pick up the requests.

3. Users configure an Adapter Request-Response Server in a TIBCO ActiveMatrix BusinessWorks process to receive the server requests that are coming from the above Request-Response Invocation service.

4. The adapter receives a request message from SAG that is sent by the SWIFTNet central SnF system. The adapter processes the request and publishes the payload field along with the other details to TIBCO environment.

5. The Blocking Activity configured in step 3, which was waiting for the requests, receives the published request. The request message is processed in TIBCO ActiveMatrix BusinessWorks.

6. The adapter Request-Response server publishes the accept status to the adapter using the Respond to Adapter Request activity configured for the activity in step 3.

7. The adapter receives the response, then sends the response to SAG. SAG sends the response to the SWIFTNet central SnF system.

The following schemas are used in the adapter:

- Request Schema—InterActSnFServer class
- Operation Name—receive
- Response Schema—InteractSnFServerResponse

If the Request-Response Invocation service schema is configured for receiving delivery notifications, the request schema used will be InterActSnFServerWithDeliveryNotif class. The Operation name remains the same.
InterAct Store-and-Forward Server To Receive Delivery Notifications

To configure the InterAct store-and-forward server for receiving delivery notifications, follow these steps:

1. Users configure a SWIFTNet adapter instance with the appropriate Remote API configuration file name.

2. In the SWIFTNet adapter instance mentioned in step 1, users configure a Request-Response Invocation Service for the Store-and-Forward (SnF) mode. Users check the **Delivery Notification** checkbox when configuring the service. Users also need to configure all other required fields along with the receiver queue from where the adapter should pick up the delivery notifications.

3. Users configure an Adapter Request-Response Server in a TIBCO ActiveMatrix BusinessWorks process to receive the delivery notifications details coming from the above Request-Response Invocation service.

4. The adapter receives a delivery notification from SAG that is sent by SWIFTNet central SnF system. The adapter processes it and publishes the SwiftRef and AcceptStatus along with the other details to TIBCO environment.

5. The Blocking Activity configured in step 3, which is waiting for events in the adapter, receives the published delivery notification details.

Receiving delivery notifications is a one-way invocation.

The following schemas are used in the adapter:

- Request Schema—InterActSnFServerWithDeliveryNotif class
- Operation Name—notifyDelivery
Index

A
adapter instance fields
  configuration tab 21
  logging tab 23
  runtime connection tab 22
  startup tab 26
adapter overview
  components 4
  features 4
adapter services
  request-response invocation service 7
  request-response service 6
Adapter Services Options 27
Advanced Configuration Options 45

C
CONFIG_HOME xiii
configuring an adapter instance 13
configuring the services 14
creating a project 11
creating an EAR file in TIBCO Designer 40
customer support xv

D
delivery mode
  durable 31
  non-durable 31
  non-persistent 30
  persistent 30
deploying and testing the adapter 16
deploying the project 41

E
ENV_NAME xiii
exporting the project 15

I
InterAct real time client 64
InterAct real time server 66
InterAct store-and-forward client 68
InterAct store-and-forward server 69
InterAct store-and-forward server to receive delivery notifications 70

J
JMS connection types
  Queue 5
  Topic 5

M
messaging
  real-time 3
  store-and-forward 3
monitoring the adapter 43

O
overview of an adapter instance 20
P

prerequisites for the configuration exercise 10

Q

qualities of service
  certified delivery 5
  distributed queue 5
  reliable 5

S

session parameters
  JMS 47
    TIBCO Rendezvous 46
starting or stopping the adapter 42
support, contacting xv
SWIFTNet overview
  SWIFT Alliance Gateway 2
  SWIFTNet messaging services 3

T

technical support xv
terminology conventions 45
trace messages
  status messages 60
  trace message fields 57
  trace messages overview 56

U

using global variables
  predefined global variables 53
  variable specification 51
using the adapter with a revision control system 48

TIBCO ActiveMatrix Adapter for SWIFTNet User's Guide