

TIBCO ActiveMatrix[®] Adapter for Tuxedo

Concepts

*Software Release 6.0
March 2010*

Important Information

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

USE OF TIBCO SOFTWARE AND THIS DOCUMENT IS SUBJECT TO THE TERMS AND CONDITIONS OF A LICENSE AGREEMENT FOUND IN EITHER A SEPARATELY EXECUTED SOFTWARE LICENSE AGREEMENT, OR, IF THERE IS NO SUCH SEPARATE AGREEMENT, THE CLICKWRAP END USER LICENSE AGREEMENT WHICH IS DISPLAYED DURING DOWNLOAD OR INSTALLATION OF THE SOFTWARE (AND WHICH IS DUPLICATED IN LICENSE.PDF) 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.

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

EJB, Java EE, J2EE, and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries.

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

THIS SOFTWARE MAY BE AVAILABLE ON MULTIPLE OPERATING SYSTEMS. HOWEVER, NOT ALL OPERATING SYSTEM PLATFORMS FOR A SPECIFIC SOFTWARE VERSION ARE RELEASED AT THE SAME TIME. SEE THE README.TXT 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 © 1998-2010 TIBCO Software Inc. ALL RIGHTS RESERVED.

TIBCO Software Inc. Confidential Information

Contents

Figures	v
Tables	vii
Preface	ix
Related Documentation	x
TIBCO ActiveMatrix Adapter for Tuxedo Documentation	x
Other TIBCO Product Documentation	xi
Third-Party Documentation	xi
Typographical Conventions	xii
How to Contact TIBCO Support	xv
Chapter 1 Introduction	1
What is an Adapter?	2
Adapter Components	3
Adapter Key terms	4
Adapter Services	5
Publication Service	5
Subscription Service	5
Request-Response Service	5
Request-Response Invocation Service	6
Adapter Services Summary	6
Choosing an Adapter Service	7
Adapter Life Cycle	9
Internationalization	10
How TIBCO Adapters Support Unicode	11
How TIBCO Messaging Encoding is Set	12
Chapter 2 TIBCO ActiveMatrix Adapter for Tuxedo	15
Overview	16
Adapter and Tuxedo Interaction	18
Adapter Features	21
Adapter Services	24
Publication Service	25

- Subscription Service 26
- Request-Response Service 28
- Request-Response Invocation Service 30
- Chapter 3 Adapter Infrastructure Tools 31**
 - TIBCO Runtime Agent 32
 - TIBCO Domain Utility 32
 - TIBCO Designer 33
 - TIBCO Administrator 34
 - TIBCO Administration Domain 34
 - TIBCO Administration Server 35
 - TIBCO Administrator GUI 35
 - TIBCO ActiveMatrix BusinessWorks 37
 - TIBCO Hawk 38
 - Adapter Microagents 38
 - TIBCO Business Studio 40
 - ActiveMatrix Resource Wizard 41
 - Composite Element Editors 41
 - Debugger 42
 - Service Assembly Editor 43
 - TIBCO ActiveMatrix Administrator 44
 - TIBCO ActiveMatrix Administrator Architecture 44
 - Enterprise and Environment Administration 46
 - Service Administration 46
- Appendix A Encoding Tables 49**
 - Encoding Tables 50
- Index 55**

Figures

Figure 1	Adapter Encoding Scenario	13
Figure 2	TIBCO ActiveMatrix Adapter for Tuxedo Services	24
Figure 3	Typical Publication Service Flow	25
Figure 4	Typical Subscription Service Flow	27
Figure 5	Typical Request-Response Service Flow.	29
Figure 6	Typical Request-Response Invocation Service Flow	30
Figure 7	TIBCO Designer main window	33
Figure 8	TIBCO Administrator GUI.	36
Figure 9	TIBCO Business Studio	40
Figure 10	Component Property Sheet	42
Figure 11	Debugger	42
Figure 12	Service Assembly Editor	43
Figure 13	TIBCO ActiveMatrix Administration Architecture	44
Figure 14	TIBCO ActiveMatrix Administrator	45

Tables

Table 1 General Typographical Conventions xii

Table 2 Syntax Typographical Conventions xiii

Table 3 Adapter Services Summary 6

Table 4 Communication Paradigms between Tuxedo and the Adapter for Each Adapter Service 18

Table 5 ISO Character Sets 50

Table 6 EBCDIC Character Sets 52

Table 7 Microsoft Windows Character Sets 54

Preface

This document explains the concepts of TIBCO ActiveMatrix Adapter for Tuxedo.

Topics

- [Related Documentation, page x](#)
- [Typographical Conventions, page xii](#)
- [How to Contact TIBCO Support, page xv](#)

Related Documentation

This section lists documentation resources you may find useful.

TIBCO ActiveMatrix Adapter for Tuxedo Documentation

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

- *TIBCO ActiveMatrix Adapter for Tuxedo Concepts* Read this manual to familiarize yourself with the concepts used by this product.
- *TIBCO ActiveMatrix Adapter for Tuxedo Installation* Read this manual to learn how to install TIBCO ActiveMatrix Adapter for Tuxedo.
- *TIBCO ActiveMatrix Adapter for Tuxedo Configuration and Deployment* Read this manual for instructions on creating, configuring, and deploying adapter projects.
- *TIBCO ActiveMatrix Adapter for Tuxedo Examples* Read this manual to work through the examples provided with the adapter.
- *TIBCO ActiveMatrix Adapter for Tuxedo Release Notes* Read the release notes for a list of new and changed features. This document also contains lists of known issues and closed issues for this release.

The following documents form the TIBCO ActiveMatrix Adapter Service Engine for Tuxedo documentation set:

- *TIBCO ActiveMatrix Adapter Service Engine for Tuxedo Installation* Read this manual to learn how to install TIBCO ActiveMatrix Adapter Service Engine for Tuxedo.
- *TIBCO ActiveMatrix Adapter Service Engine for Tuxedo Configuration and Deployment* Read this manual for instructions on creating, configuring, and deploying adapter projects.
- *TIBCO ActiveMatrix Adapter Service Engine for Tuxedo Examples* Read this manual to work through the examples provided with the adapter.
- *TIBCO ActiveMatrix Adapter Service Engine for Tuxedo Release Notes* Read the release notes for a list of new and changed features. This document also contains lists of known issues and closed issues for this release.

Other TIBCO Product Documentation

You may find it useful to read the documentation for the following TIBCO products. Each of the books is available from the `doc` directory in the product's installation area.

- TIBCO Designer™
- TIBCO Administrator™
- TIBCO ActiveMatrix® Administrator
- TIBCO ActiveMatrix BusinessWorks™
- TIBCO ActiveMatrix BusinessWorks™ Service Engine
- TIBCO Rendezvous®
- TIBCO Enterprise Message Service™
- TIBCO Hawk®
- TIBCO Adapter™ SDK
- TIBCO Runtime Agent™
- TIBCO Business Studio™
- TIBCO ActiveMatrix® Service Grid
- TIBCO ActiveMatrix® Service Bus

Third-Party Documentation

You may also find it useful to read the Oracle Tuxedo User documentation.

Typographical Conventions

The following typographical conventions are used in this manual.

Table 1 General Typographical Conventions

Convention	Use
<i>TIBCO_HOME</i>	<p>Many TIBCO products must be installed within the same home directory. This directory is referenced in documentation as <i>TIBCO_HOME</i>. The value of <i>TIBCO_HOME</i> depends on the operating system. For example, on Windows systems, the default value is C:\tibco.</p> <p>Other TIBCO products are installed into an installation environment. Incompatible products and multiple instances of the same product are installed into different installation environments. The directory into which such products are installed is referenced in documentation as <i>ENV_HOME</i>. The value of <i>ENV_HOME</i> depends on the operating system. For example, on Windows systems the default value is C:\tibco.</p> <p>TIBCO ActiveMatrix installs into a directory inside <i>ENV_HOME</i>. This directory is referenced in documentation as <i>AMX_HOME</i>. The value of <i>AMX_HOME</i> depends on the operating system. For example, on Windows systems the default value is C:\tibco\amx\.</p> <p>TIBCO ActiveMatrix Adapter Service Engine for Tuxedo is installed in a directory inside <i>AMX_HOME</i>.</p>
<i>ENV_HOME</i>	
<i>AMX_HOME</i>	
code font	<p>Code font identifies commands, code examples, filenames, pathnames, and output displayed in a command window. For example:</p> <p>Use MyCommand to start the foo process.</p>
bold code font	<p>Bold code font is used in the following ways:</p> <ul style="list-style-type: none">• In procedures, to indicate what a user types. For example: Type admin.• In large code samples, to indicate the parts of the sample that are of particular interest.• In command syntax, to indicate the default parameter for a command. For example, if no parameter is specified, MyCommand is enabled: MyCommand [enable disable]

Table 1 General Typographical Conventions (Cont?)




Convention	Use
<i>italic font</i>	<p>Italic font is used in the following ways:</p> <ul style="list-style-type: none"> To indicate a document title. For example: See <i>TIBCO ActiveMatrix BusinessWorks Concepts</i>. To introduce new terms For example: A portal page may contain several portlets. <i>Portlets</i> are mini-applications that run in a portal. To indicate a variable in a command or code syntax that you must replace. For example: <code>MyCommand <i>PathName</i></code>
Key combinations	<p>Key name separated by a plus sign indicate keys pressed simultaneously. For example: <code>Ctrl+C</code>.</p> <p>Key names separated by a comma and space indicate keys pressed one after the other. For example: <code>Esc, Ctrl+Q</code>.</p>
	The note icon indicates information that is of special interest or importance, for example, an additional action required only in certain circumstances.
	The tip icon indicates an idea that could be useful, for example, a way to apply the information provided in the current section to achieve a specific result.
	The warning icon indicates the potential for a damaging situation, for example, data loss or corruption if certain steps are taken or not taken.

Table 2 Syntax Typographical Conventions

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

Table 2 Syntax Typographical Conventions

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

How to Contact TIBCO Support

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

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

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

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

<https://support.tibco.com>

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

Chapter 1 **Introduction**

This chapter introduces adapters by explaining adapter key components, terms, and services.

Topics

- [What is an Adapter?, page 2](#)
- [Adapter Components, page 3](#)
- [Adapter Key terms, page 4](#)
- [Adapter Services, page 5](#)
- [Choosing an Adapter Service, page 7](#)
- [Adapter Life Cycle, page 9](#)
- [Internationalization, page 10](#)

What is an Adapter?

To deploy the best solution for each aspect of your business, you usually have to purchase applications from several different application vendors. Unfortunately, vendors typically have their own way to format and expose data. Therefore integrating the various applications across your enterprise poses significant challenges.

An adapter provides a bridge between an application and your TIBCO integration environment. Using a no-coding approach to integration, TIBCO Adapters enable packaged applications, databases, and other technologies to become active participants in the enterprise information flow, regardless of their data formats or communication protocols. Integration of new applications does not require programming skills and does not interfere with existing infrastructure.

Adapters isolate the application from more complex actions. Message transformation and business process automation can be handled once the data is published to the TIBCO infrastructure.

As shown in the next diagram, adapters allow data to be exchanged among different technologies.

- Adapters are available for off-the-shelf applications from leading vendors. Each adapter integrates with at least one, and usually several, of the interfaces exposed by the vendor application.
- Database adapters enable an enterprise's database to initiate important business processes based on exception data they identify. Database adapters also make data available to the enterprise.
- Mainframe adapters enable real-time two way communication between them and the rest of a companies' business applications and databases.
- Adapters can also enable integration with component or object development models and other messaging technologies.

Adapter Components

The adapter can run either as a standalone process or as a service. When run as a service, the adapter participates in the Service Oriented Architecture (SOA) environment.

The adapter components are:

- Standalone

Using this component you create adapter projects which run as a standalone process. This adapter component is referred to as the *standalone adapter*.

Standalone adapter projects are created and configured using TIBCO Designer and deployed using TIBCO Administrator.

- Adapter Service Engine

Using this component you create and configure adapter projects that can be deployed as a service in the TIBCO ActiveMatrix environment. This adapter component is referred to as the *adapter service engine*.

Adapter service engine projects are created using TIBCO Business Studio and deployed using TIBCO ActiveMatrix Administrator.

The adapter component is wired with other composite elements in the ActiveMatrix SOA Project.

Existing standalone adapter configurations can also be deployed as services.



Throughout this book, references to an adapter include both the standalone and adapter service engine components.

Adapter Key terms

The following key terms are used when describing adapter interactions in this manual.

- A *palette* is a standalone adapter component that contains the screens used to gather input at design-time when configuring an adapter with a service. The palette is accessed via TIBCO Designer.
- A *project* is a collection of all configured adapter resources. A *project* contains configuration information for one or more adapter instances. A local project is typically used at design-time for testing adapter instances. For production, a project is typically managed by an administration server provided by the TIBCO Administrator for the standalone adapter and TIBCO ActiveMatrix Administrator for the adapter service engine.
- An *.ear* of an application contains global variables with values set at design-time by the standalone adapter. The global variables can be changed at deployment at the application level, service level, or service instance level.
- *Service Oriented Architecture (SOA)* is a software architecture in which applications and data are decomposed into discrete, operationally independent services, which can be executed in a highly distributed manner.
- A *business object* is the representation of the data model of the entities that the adapter service engine connects to. The business objects are downloaded during also the design phase and are used by the adapter services.
- A *container* is an ActiveMatrix runtime entity that hosts component implementations and service bindings.
- A *service assembly* is an ActiveMatrix deployment package. It contains service units and a descriptor that indicates the container into which each service unit is to be deployed. The suffix of a service assembly file is `.saf`.
- The ActiveMatrix services are described in documents expressed in *Web Services Description Language (WSDL)*. The WSDL documents specify the messages that are required to access a service.
- During any service interaction, each service will adopt one of two roles: provider or consumer. A service *provider* publishes a WSDL document that describes the services it offers. A service *consumer* uses the WSDL document to determine the available services and the messages required to access the services. A *message exchange pattern (MEP)* defines the sequence and cardinality of messages sent between the provider and the consumer.

Adapter Services

Adapters are responsible for making information from different applications available to other applications across an enterprise. To do so, an adapter is configured to provide one or more of the following services:

Publication Service

An adapter *publication service* recognizes when business events happen in a vendor application, and asynchronously sends out the event data in realtime to interested systems in the TIBCO environment.

For example, an adapter can publish an event each time a new customer account is added to an application. Other applications that receive the event can then update their records just as the original application did.

Subscription Service

An adapter *subscription service* asynchronously performs an action—such as updating business objects or invoking native APIs—on a vendor application. The adapter service listens to external business events, which trigger the appropriate action.

Referring to the previous example, an adapter subscription service can listen for customer record creation events (happening in an application and published to the TIBCO infrastructure) and update another application.

Request-Response Service

In addition to asynchronously publishing and subscribing to events, an adapter can be used for synchronously retrieving data from or executing transactions within a vendor application. After the action is performed in the vendor application, the adapter service sends a response back to the requester with either the results of the action or a confirmation that the action occurred. This entire process is called *request-response*, and it is useful for actions such as adding or deleting business objects.

For example, an adapter receives a request message from the TIBCO infrastructure and sends it to an application. The adapter gets a response from the application and returns it.

Request-Response Invocation Service

An adapter *request-response invocation* service is similar to the request-response service, except that the roles are reversed. The vendor application is now the requester or initiator of the service, instead of the provider of the service. The adapter service acts as a proxy, giving the vendor application the ability to invoke synchronously functionality on an external system.

For example, the adapter sending a request message from application Y to application X. After it processes the message, it is returned to the adapter, which sends the response back to application Y.

Adapter Services Summary

The next table summarizes the services introduced in this section.

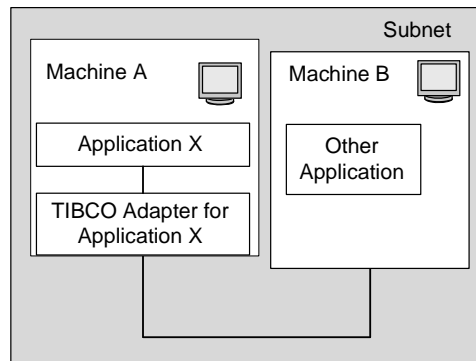
Table 3 Adapter Services Summary

Service	Initiator	Target	Event Mode
Publishing service (sends to target)	Vendor application	TIBCO infrastructure	Asynchronous
Subscribing service (gets from initiator)	TIBCO infrastructure	Vendor application	Asynchronous
Request-response service (gets from initiator, waits for response then sends response to target)	TIBCO infrastructure	Vendor application	Synchronous
Request-response invocation service (sends to target, waits for response, then sends response to initiator)	Vendor application	TIBCO infrastructure	Synchronous

Choosing an Adapter Service

A business integration scenario drives the choice of one adapter service or another. This section provides a simple flow chart that helps you to choose the service to use. Not all adapters provide all services and some adapters may provide additional services not listed here. See your adapter user's guide for details.

Consider the following environment that involves application X, an adapter, and another application:

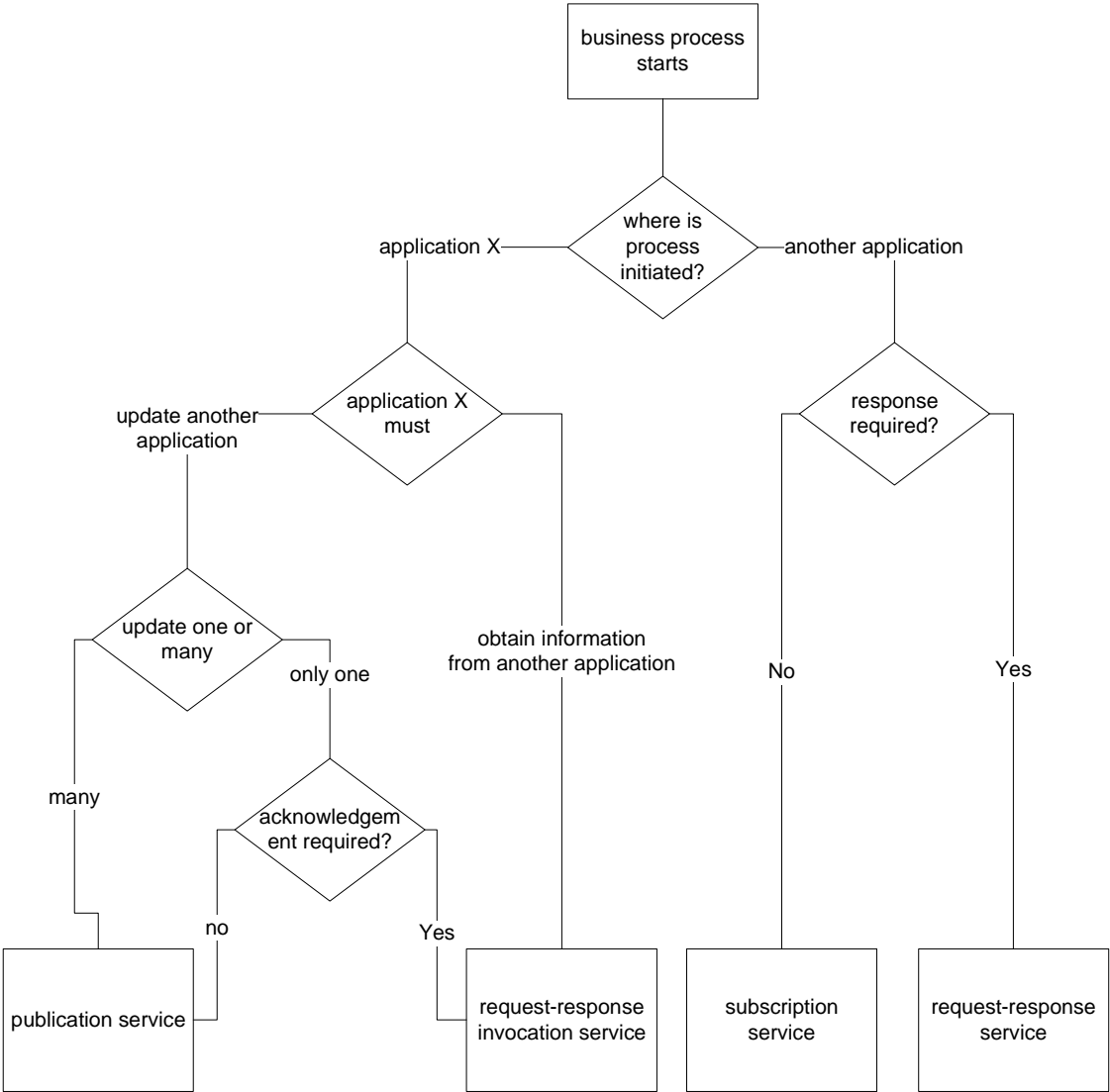


In this scenario, data must be exchanged between the application X and another application. The other application could be a customer management system, such as PeopleSoft, or another TIBCO application, such as TIBCO ActiveMatrix BusinessWorks.

To decide the adapter service to configure in the adapter, start by finding out where the scenario begins—what triggers it. Is the scenario triggered by an event inside the application X, or inside the other application?

For example, when a new customer account is created in application X, must the account information also be propagated via the adapter to the other application? Or does a batch business process in TIBCO ActiveMatrix BusinessWorks need information from application X to generate some report?

This question is the starting point of the decision chart provided below:



Working through the decision chart, if the business process is the creation of a customer record in application X and if many other applications need to be updated when the event occurs, but no acknowledgements are required, the adapter's publication service should be used.

Adapter Life Cycle

In general, the life cycle of an adapter includes four stages: installation, configuration, deployment, and monitoring.

Installation

The installation stage includes installing the vendor application to which the adapter connects and other software from TIBCO on which the adapter depends.

For many adapters, the adapter and vendor application need not be installed on the same machine, while the TIBCO Runtime Agent software must be installed on each computer that runs the adapter.

Configuration

In the configuration stage, an adapter instance can be created and configured with a design-time tool. The configuration information is required for a runtime adapter to interact with the vendor application and other applications.

The standalone adapter uses TIBCO Designer as its design-time tool, while the adapter Service Engine uses TIBCO Business Studio.

Deployment

An adapter instance created by TIBCO Designer can be deployed using TIBCO Administrator or TIBCO ActiveMatrix Administrator. In the latter case, you must first import the Designer project into TIBCO Business Studio or convert the EAR file to a service assembly.

An adapter instance created by TIBCO Business Studio can only be deployed using TIBCO ActiveMatrix Administrator.

Monitoring

In this stage, use one of the following tools to manage and monitor the adapter:

- the built-in monitoring tools provided by TIBCO Administrator or TIBCO ActiveMatrix Administrator
- the TIBCO Hawk microagents

Internationalization

TIBCO Adapter products are based on the TIBCO messaging infrastructure, which supports data transmission in English and most European, Mid-Eastern and Asian languages. This section starts by describing basic encoding concepts, then describes how an adapter works in transmitting data in various languages.

Encoding

When the ASCII character set was extended to support all western European languages, another bit was added creating 128 more spaces. This 8-bit or 1-byte character set is ISO-8859-1 (or Latin1). In ISO-8859-1, the most significant bit is set to represent the additional 128 slots, and most of them are used to encode Western European characters. When the most significant bit is not set, the eight bits still represent ASCII characters. Thus ASCII is a subset of these extended character sets.

Because one byte was not sufficient to include the Eastern European characters, a separate encoding or character set, called ISO-8859-2 (or Latin2) was created, which is not compatible with ISO-8859-1. Other ISO 8859 series character sets were also invented to handle a larger selection of languages. Again, all of these ISO-8859 character sets are a superset of and backward compatible with ASCII.

However, the code spaces provided by 1 byte (or 8 bits) are not enough for representing the tens of thousands of characters used in languages such as Chinese, Japanese, and Korean (CJK). Thus, multibyte character sets were developed, together with the corresponding encoding methods for each character set. These include those locale-independent encodings such as ISO-2022, EUC, and locale-dependent encodings like Shift_JIS (Japanese), GB2312 (Simplified Chinese), Big5 (Traditional Chinese), and so on.

Unicode Basics

Unicode is a way to represent characters of all known languages of the world, which are defined in a character set named UCS (Universal Character Set). It provides a unique code point for every character regardless of platform, program or language.

The advantage of Unicode is that characters from the world's major scripts are uniformly supported. Thus applications running on different platforms with different locales can exchange information without misinterpretation, as long as they follow this uniform character set.

Locale

Locale is the information for a specific combination of language, territory (cultural data), and codeset. Examples are `en_US.ASCII`, `ja_JP.Shift_JIS`, etc. Normally, an operating system is started with a particular locale.

UCS, Unicode and UTF-8

UCS is the abbreviation for Universal Character Set, which is specified by International Standard ISO/IEC 10646. UCS contains the characters of almost all the world's major scripts.

Unicode is a standard that defines a character code set (UCS, or Universal Character Set, defined in ISO/IEC 10646) that assigns a unique code point (scalar value) to each character of almost all the world's major languages. The Unicode standard also includes a series of character encodings that represent each of these code points, such as UTF-8, UTF-16 (UCS-2), UTF-32, etc. For more information about Unicode, refer to Unicode Consortium's official web site: www.unicode.org.

UTF-8 denotes Unicode Transformation Format 8 bit, a common Unicode encoding that serializes a Unicode scalar value as a sequence of one to four bytes. The purpose of the transformation encoding is that the data represented this way can be passed reliably through single byte environments.

UTF-8 is popular for data exchange between applications and protocols such as HTTP, XML, MIME, and SOAP. One nice feature of UTF-8 is that it is backward compatible with ASCII.

How TIBCO Adapters Support Unicode

TIBCO Adapter products provide Unicode encoding support by taking advantage of UTF-8 as the TIBCO messaging encoding when exchanging data among TIBCO components (TIBCO applications and adapters). Obviously, this can apply only to text data.

In an adapter project where only ASCII or Latin-1 (ISO8859-1) data is exchanged between adapters and other TIBCO products, ISO8859-1 can be used as the TIBCO messaging encoding.

TIBCO Adapter products still use other native encodings (like ISO-8859 series, MS-Windows series, Shift_JIS, Big5, GB2312, etc.) to communicate with vendor applications (SAP, Oracle, Siebel, as examples). But the adapters internally use UCS-2 to represent text data. So the adapters are responsible for the encoding conversion between the vendor encoding and UCS-2, and between the TIBCO messaging encoding and UCS-2.

How TIBCO Messaging Encoding is Set

The encoding property is set on the project itself at design time, and in the TIBCO administration server's property file when creating a TIBCO Administration Domain.

Setting TIBCO Messaging Encoding in TIBCO Designer

The project setting is used at design time when using the Adapter Tester or TIBCO ActiveMatrix BusinessWorks tester to verify an adapter instance or BusinessWorks process configuration. The project setting is also used when the project is exported as a local repository (in `.dat` format).

The encoding value is set on the root project folder. By default, the value is set to ISO8859-1. You can change the value by selecting the folder and under the Project Settings tab, changing the value for the TIBCO Message Encoding field.

Setting TIBCO Messaging Encoding for the TIBCO Administration Server

The TIBCO administration server setting is used when the project is exported to a server repository or deployed using TIBCO Administrator.

For a server based project, the TIBCO messaging encoding is set by the `repo.encoding` property in the server's `tibcoadmin-domain-name.tra` configuration file (located in `TIBCO_HOME/tibco/administrator/version_num/bin/`).

The encoding is set when using the TIBCO Domain Utility to create the domain or by editing the `repo.encoding` property in the `.tra` configuration file.

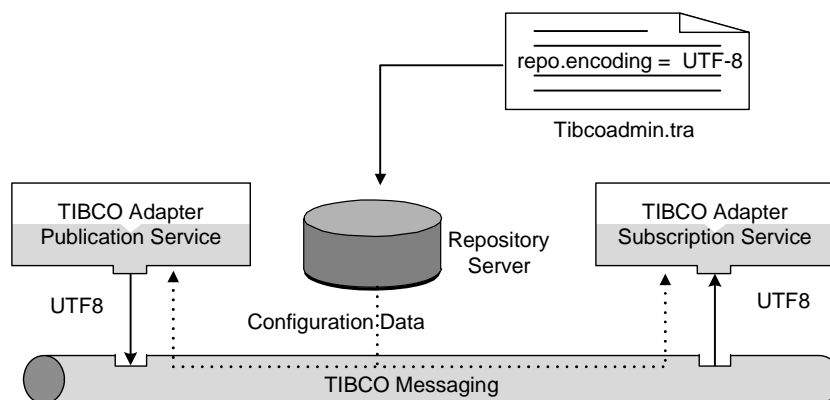
Each adapter or TIBCO application that uses the same server for storing and retrieving configuration data uses this encoding setting when communicating to each other. This assures that all TIBCO components (including adapters and other TIBCO applications) that belong to the same project use the same encoding value to communicate.



For TIBCO Adapter release 4.x, the `repo.encoding` property is defined in the server's `server.ini` file under the `TIBCO_HOME/tibco/repository/bin/` directory

The following diagram shows a scenario where the TIBCO messaging encoding of two adapter services is controlled by the server's encoding property. The property is set to UTF-8.

Figure 1 Adapter Encoding Scenario



The encoding property set in the project file is always superseded by the server's encoding property.

The encoding property discussed above is the encoding used by the TIBCO messaging between adapters and applications, not the encoding used for the persistent storage of the project files.

Available TIBCO Messaging Encodings

In a typical integration environment, all TIBCO ActiveEnterprise adapters and applications use the same project, connect to the same administration server, and use the server encoding property. Two encoding choices are available for this kind of project, ISO8859-1 and UTF-8.

- ISO8859-1 should be used for projects in which the text data to be exchanged is only ASCII or Western European data.
- UTF-8 should be used for projects that handle data other than ASCII and Western European character set.

Note that all the world's major language characters can be represented by UTF-8. Also, note that additional encodings can be used by an adapter to communicate with its vendor application or database. See for a list of additional encodings supported by most adapters.

Chapter 2 **TIBCO ActiveMatrix Adapter for Tuxedo**

Topics

- [Overview, page 16](#)
- [Adapter and Tuxedo Interaction, page 18](#)
- [Adapter Features, page 21](#)
- [Adapter Services, page 24](#)

Overview

TIBCO ActiveMatrix Adapter for Tuxedo connects to Tuxedo integrated applications and exchanges data in real-time with the Tuxedo application and other applications configured for the TIBCO environment. The adapter maintains data integrity, and integrates other software in the TIBCO environment suite seamlessly into an enterprise.

Oracle Tuxedo is a distributed transaction middleware that allows you to monitor and process distributed transactions across various applications, databases and platforms used in an enterprise. It serves as an application server that is used to develop applications and handle functions like business logic creation, transaction management, fault-tolerance, and load balancing. An application built using Tuxedo offers a variety of functionality that is collectively termed Tuxedo services.

Components and concepts of Tuxedo that are used to facilitate integration between a Tuxedo application and other applications are briefly described next:

Application-to-Transaction Monitor Interface (ATMI) — ATMI is the application programming interface to the Tuxedo system. It includes transaction routines, message handling routines, service interface routines, and buffer management routines.

CArray — CArray is an undefined array of characters, any of which can be `NULL`. This type of buffer is used to handle the data opaquely, as the Tuxedo system does not interpret the semantics of the array. As the CArray is not self-describing, the length must always be provided during transmission. Encoding and decoding are not supported for messages sent between machines because the bytes are not interpreted by the system.

STRING Buffer — Array of characters that terminates with a `NULL` character.

FML32 Buffer Type — A buffer of self-describing data items accessed through the field manipulation language API. Tuxedo uses fielded buffers to define the format and layout of data flowing from the client to the server. The messages returned by the ATMI routines are packed as fielded buffers and FML (Field Manipulation Language) functions are used to manipulate the messages.

Tuxedo Services — An application routine available for request by a client in the Tuxedo system or a module of application code that carries out a service request is called a Tuxedo service. In the context of an application that is built using Tuxedo, the functionality provided is collectively known as Tuxedo services.

Tuxedo Client — A Tuxedo client invokes the Oracle Tuxedo libraries of functions and procedures known collectively as the Application-to-Transaction-Monitor Interface, or ATMI. A client joins a Tuxedo application, defines transaction boundaries, calls ATMI functions that enable it to communicate with other programs in the application and finally leaves the Oracle Tuxedo ATMI application. By joining an application only when necessary and leaving it once the appropriate task is complete, a client frees Oracle Tuxedo system resources for use by other clients and servers.

Tuxedo Server — A Tuxedo Server is a process that oversees a set of services, dispatching them automatically for clients that request them. The server dispatches each request from a Tuxedo Client to the appropriate service. The servers on which these services reside then reply to the clients or forward client requests to a new service.

EventBroker — EventBroker tracks events in the Tuxedo system as well as the adapter related events. It provides Tuxedo specific publish-subscribe functionality. The EventBroker can be used to set up automatic notifications to the adapter when an event occurs. Automatic notifications can be set up by registering with the EventBroker for the required events.

Communication Paradigms — Tuxedo provides the following five communication paradigms for messaging between client and server:

- **Synchronous** — Request-response communication where the function call is a blocking call that waits for the reply.
- **Asynchronous** — Request-response communication where the function call does not wait for the requested service to be executed before returning to the caller. The client synchronizes the results at a later stage in the processing.
- **Conversational** — A number of messages exchange until a logical conclusion is reached. This type of communication is used for interactions between modules that require an extended exchange.
- **Queue-based** — This is a guaranteed mode of message delivery that uses batch storage and retrieval of messages through the Tuxedo stored messaging facility called Queue Services.
- **Event-based** — A publish-subscribe communication, where client programs register an event in the Event Broker that should be invoked when unsolicited messages are sent to them. Such event-handling routines define responses to classes of events. The EventBroker will inform all registered Tuxedo subscribers of that event.

Adapter and Tuxedo Interaction

TIBCO ActiveMatrix Adapter for Tuxedo acts as a gateway to Tuxedo services, allowing you to invoke Tuxedo services. The adapter facilitates real-time integration between Tuxedo applications and other applications that are configured for the TIBCO environment. The adapter uses Tuxedo's ATMI interface and the communication paradigms provided by Tuxedo to interface with Tuxedo applications.

For subscription and request-response services, the adapter uses the communication paradigms to invoke a Tuxedo Service. The adapter reads, writes from, or writes into buffers.

For publication service, all communication between Tuxedo and the adapter is through EventBroker, Adapter Agent (Rendezvous Adapter Agent or JMS Adapter Agent), or Tuxedo Queue.

Table 4 lists the supported communication paradigms and the buffer type between Tuxedo and the Adapter for each adapter service.

Table 4 Communication Paradigms between Tuxedo and the Adapter for Each Adapter Service

Adapter Service	Communication Paradigms	Buffer Type
Publication	• Event-based	• FML32
	• Queue-based	• FML32, STRING, CArray
	• Rendezvous Adapter Agent-based	• FML32, STRING, CArray
	• JMS Adapter Agent-based	• FML32, STRING, CArray
Subscription	• Conversational	• FML32
	• Event-based	• FML32, STRING, CArray
	• Asynchronous	• FML32, STRING, CArray
	• Queue-based	• FML32, STRING, CArray
Request-Response	• Synchronous	• FML32, STRING, CArray
	• Asynchronous	• FML32, STRING, CArray
Request-Response Invocation	• Rendezvous Adapter Agent-based	• FML32, STRING, CArray

EventBroker

The adapter publication and subscription services support the Event-based communication paradigm. The publication service is registered with the EventBroker to listen for events that occur within the Tuxedo System. The adapter subscribes to the Tuxedo event, forms the message using the data received, and then publishes the message onto the TIBCO environment on the subject name or destination name configured in the repository. In the case of the subscription service, data from the TIBCO environment is posted on EventBroker. The Tuxedo service listening on the specified event name can access this information.

Conversational

The adapter subscription service supports the Conversational communication paradigm. The subscriber listens for the message on the subject name configured in the repository. After getting the message, the adapter uses the conversational communication mechanism to call the specified Tuxedo service. The Tuxedo service processes the request and returns the reply data to the adapter. The adapter sends the reply message to the TIBCO environment and logs the status. This process continues until the conversation is over.

Tuxedo Queue

The adapter publication and subscription services support the Queue-based communication paradigm. The Tuxedo client or server puts the data into the Tuxedo Queue. There is a timer that will trigger the adapter to check the message in the Queue of Tuxedo. The adapter receives the data from the Queue and forms an AE message using the data received, and then publishes the message onto the TIBCO environment.

Adapter Agent

A Tuxedo application invokes an Adapter Agent, and then the Adapter Agent posts the incoming data to the adapter. The Adapter Agent can be used for publication and request-response invocation services. In the case of publication service, the adapter forms the message using the data received and then publishes the message onto the TIBCO environment on the subject name or the destination name configured in the repository. In the case of request-response invocation service, in addition to posting the message on the TIBCO environment, the adapter routes the reply back to the requesting Tuxedo application through the Adapter Agent. There are two types of Adapter Agent, Rendezvous Adapter Agent and JMS Adapter Agent.

- Rendezvous Adapter Agent

Rendezvous Adapter Agent is the Adapter Agent which uses TIBCO Rendezvous to transfer the data. It can be used for publication and request-response invocation services.

In the case of publication service, it includes two situations, as follows:

- Asynchronous Rendezvous Adapter Agent. When a Tuxedo application invokes an Adapter Agent, the agent posts the incoming data to the adapter using TIBCO Rendezvous. The adapter then publishes the message onto the TIBCO environment.
- Synchronous Rendezvous Adapter Agent. When a Tuxedo application invokes an Adapter Agent, the agent posts the incoming data to the adapter using TIBCO Rendezvous, and then waits for the confirmation from the adapter. If the adapter successfully publishes the message onto the TIBCO environment, it will send a success reply to the agent. Otherwise, the adapter will send a failure reply to the agent. After that, the agent will send a reply to the Tuxedo application.

In the case of request-response invocation service, Rendezvous Adapter Agent is synchronous.

- JMS Adapter Agent

JMS Adapter Agent is the Adapter Agent which uses JMS to transfer the data. The Adapter Agent posts the received data to the JMS server, from which the adapter gets the message. The JMS Adapter Agent can be used for publication service.

Adapter Features

The following features are described in detail in *TIBCO ActiveMatrix Adapter for Tuxedo Configuration and Deployment*.

- **Support for Refreshing of Schemas** The adapter now supports the refreshing of schema. Any changes to the Tuxedo service header files loaded in the TIBCO Designer will now be reflected in the adapter schema while the previously configured sequences are maintained.
- **Connection Management** The adapter now supports connection management. The adapter attempts to reconnect to the server based on the reconnection parameters specified at the time of configuration.
- **Support for Identification of FML32 Fields using Field Names** The adapter supports identification of FML32 fields using field names. Field IDs for FML32 fields will be loaded at run-time, thus avoiding changes in the adapter whenever new versions of Tuxedo services are released.
- **Support for FML32, STRING, and CArray Buffer Types** The adapter supports FML32, STRING, and CArray buffer types for communication with Tuxedo.
- **Support for Repeating Fields Within an FML32 Structure** The adapter supports the sequencing of chosen fields within an FML32 buffer for all adapter services, except for the subscription services using the Conversational communication paradigm.
- **Support for Optional FML32 Buffer Fields** The adapter accepts null values for fields, which are part of the FML32 buffer.
- **FML32 Field Type**
 - Supported field types are FLD_SHORT, FLD_LONG, FLD_CHAR, FLD_FLOAT, FLD_DOUBLE, FLD_STRING, FLD_CARRAY.
 - Not supported field types are FLD_PTR, FLD_FML32, FLD_VIEW32, FLD_MBSTRING.
- **A Convenient Easy-to-use Configuration Tool** The adapter is configured using TIBCO Designer. TIBCO Designer connects to a specified Tuxedo system, downloads schema for the required services, and transfers the schema and message routing rules for individual services to a central repository. Support for usage of global and client variables ensures easy and quick migration of adapter configurations from development to testing, and from testing to production environments.
- **Support for TIBCO ActiveMatrix BusinessWorks** The adapter can be used in a TIBCO ActiveMatrix BusinessWorks process.

- **Support for Tuxedo Communication Paradigms** The adapter supports the following communication paradigms provided by Tuxedo.
 - Synchronous
 - Asynchronous
 - Conversational
 - Queue-based
 - Event-based
- **Support for Adapter Agent-Based Communication Paradigms** The adapter supports two types of Adapter Agent-based communication paradigms.
 - Rendezvous Adapter Agent-based
 - JMS Adapter Agent-based
- **Asynchronous Calls to Tuxedo from the Adapter Request-Response Service** The adapter (Workstation client or native client mode) Request-Response service can issue up to a certain number of requests in parallel without waiting for the reply, thereby decreasing the overall response time.
- **Native and Workstation Client Types** The adapter can work as a Tuxedo Workstation client or a Native client.
- **Multithreading** The adapter supports multithreading which allows the adapter to process multiple messages concurrently.
 - For TIBCO ActiveMatrix Adapter for Tuxedo, the number of threads can be specified at the time of configuration.
 - For TIBCO ActiveMatrix Adapter Service Engine for Tuxedo, the number of threads can be specified only for the publication and request-response invocation services in the adapter properties file. The number of threads for the subscription and request-response services cannot be configured.
- **Delayed Acknowledgement** Delayed acknowledgment is a feature built upon certified messaging. This is applicable only for Subscription services. When enabled, the adapter will not send the reception confirmation immediately upon receiving the message. Instead, it will hold on to the confirmation until the message is received by the target application (Tuxedo). This way the sender can be sure that its message has reached its final destination: Tuxedo (and not just the TIBCO ActiveMatrix Adapter for Tuxedo).
- **Exception Handling, Monitoring and Message Tracking** The adapter employs effective exception-handling techniques and extensive audit trails. The adapter can be configured to work in concert with the TIBCO Hawk monitoring component to detect and handle exception situations. Each message that is processed by an adapter service is tagged with a tracking-ID

that enables complete end-to-end tracking of messages in the TIBCO environment.

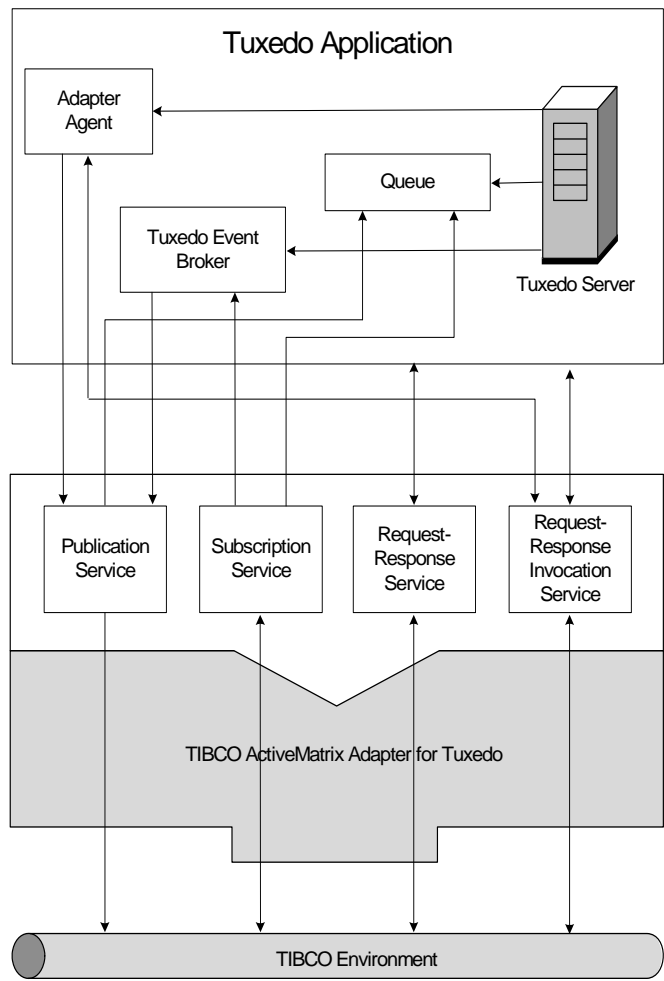
- **Examples** A set of examples demonstrates configuration of the adapter as well as interaction with TIBCO ActiveMatrix BusinessWorks. Pre-configured configuration files and sample Tuxedo services are included in the adapter installation and are documented in the *TIBCO ActiveMatrix Adapter for Tuxedo Examples Guide*.

Adapter Services

The adapter acts as an agent to export and import data into the Tuxedo system in an event-driven fashion. The adapter offers publication, subscription, request-response, and request-response invocation services.

This section describes how the adapter services interact with the Tuxedo environment to provide message connectivity. [Figure 2](#) shows the adapter services in relation to other components in a Tuxedo system.

Figure 2 TIBCO ActiveMatrix Adapter for Tuxedo Services

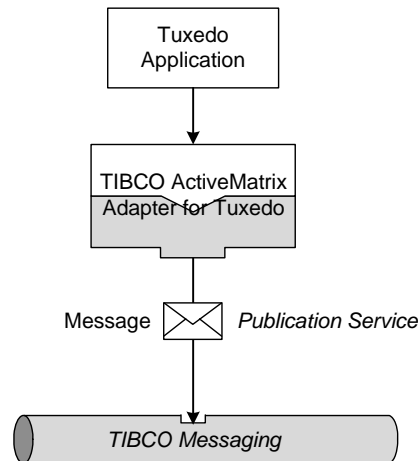


Publication Service

A publication service is used to publish data from Tuxedo to the TIBCO environment. The adapter receives a message from a Tuxedo integrated application and the publication service converts the retrieved data into the TIBCO ActiveEnterprise wire format and publishes the data to the TIBCO environment.

Figure 3 illustrates a publication service work flow.

Figure 3 Typical Publication Service Flow



Publication Service Features

- A publication service allows user defined subject and destination names.
- Multithreading is available. Multithreading allows the adapter to process multiple messages concurrently ensuring better performance.
- Publication service supports the following communication paradigms.
 - Event-based Communication
 - Queue-based Communication
 - Rendezvous Adapter Agent-based
 - JMS Adapter Agent-based

Example Scenarios

Here are the example scenarios for using a publication service:

- Example Scenario for Event-based Publication

A Publication service can publish event data received from the Tuxedo system. For example, a user may put in a request on the Tuxedo client for an operation such as `save` or `delete`. The Tuxedo system processes the event and posts it on the Tuxedo `EventBroker` engine. The adapter's Publication service, which is registered to the Tuxedo `EventBroker` to listen for events, picks up the message and publishes it.

- **Example Scenario for Queue-based Publication**

The Publication service can publish data received from the Tuxedo system. For example, a user may put in a request on the Tuxedo client for an operation like `save` or `delete`. The Tuxedo system puts the data into the queue of Tuxedo, in which the adapter checks the message. After receiving the message, the adapter publishes the message onto the TIBCO environment.

- **Example Scenario for Rendezvous Adapter Agent-based publication**

The Publication service can publish data received from the Tuxedo system. For example, a user may put in a request on the Tuxedo client for an operation like `save` or `delete`. The Tuxedo server program invokes `callPub/callArrayPub` or `callSyncPub/callSyncArrayPub` methods of the Adapter Agent with the required arguments. The Rendezvous Adapter Agent passes the data to the adapter, which publishes the message.

- **Example Scenario for JMS Adapter Agent-based Publication**

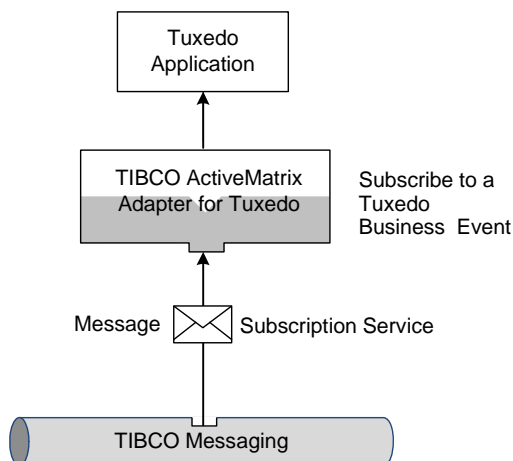
The Publication service can publish data received from the Tuxedo system. For example, a user may put in a request on the Tuxedo client for an operation like `save` or `delete`. The Tuxedo system invokes the agent, which invokes the `callPubJMS` method or the `callArrayPubJMS` method and posts the data into JMS server, from which the adapter receives the message. After receiving the message, the adapter publishes the message onto the TIBCO environment.

Subscription Service

A subscription service is used to subscribe to messages from the TIBCO environment and post them into the Tuxedo application. A subscription service invokes Tuxedo services using the Tuxedo service name and communication paradigm specified in the incoming message.

[Figure 4](#) illustrates a subscription service work flow.

Figure 4 Typical Subscription Service Flow



Subscription Service Features

- Subscription service supports the following Tuxedo communication paradigms:
 - Conversational Communication
 - Asynchronous Communication
 - Queue-based Communication
 - Event-based Communication

When the Queue-based or Conversational communication paradigm is used, a reply is published using the available reply subject.



In the case of Queue-based communication, even if a reply is not required, a reply subject should be specified while configuring the Subscription service. You also need to set the `ReplyQueue` parameter in the `ubbconfig` file. See *TIBCO ActiveMatrix Adapter for Tuxedo Configuration and Deployment Appendix D Message Formats* for details.

When the Conversational communication paradigm is used, the adapter retrieves the specified number of rows as configured in a single conversation, but publishes each row as a separate TIBCO ActiveEnterprise message.

For the sending application (TIBCO ActiveMatrix BusinessWorks) to recognize the last batch of messages from the adapter (so that it can stop

waiting and proceed further), the `IsEnd` boolean field is set by the adapter. This happens only if at least 1 row has been retrieved.



Sequences are not to be used with the Conversational communication paradigm. For the Conversational communication paradigm, only `FML32` buffer type is supported.

- If certified messaging is used, delayed acknowledgement is available. After a message is received, the adapter sends an acknowledgement to the publication service only after the message has been processed successfully by the Tuxedo system.
- The `FML32`, `STRING` and `CArray` buffer types are supported.
- Multithreading is available. Multithreading allows the adapter to process multiple messages concurrently ensuring better performance.
- Multiple connections to the Tuxedo server are supported.

Example Scenario

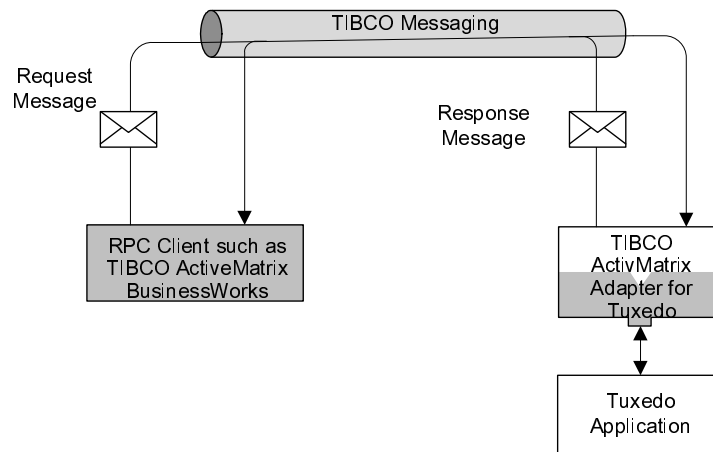
An external system publishes a message in the TIBCO environment, which is mapped to the adapter schema. The adapter configuration subscribes to this message and invokes the specified Tuxedo service using the specified communication paradigm.

Request-Response Service

A request-response service is used to serve the requests coming from external applications through the TIBCO environment. The adapter supports request-response scenarios by acting as an RPC server. When the adapter receives a request, it takes the raw requested data, converts it into formatted Tuxedo schema, and sends it to Tuxedo using a designated Tuxedo interface.

[Figure 5](#) illustrates a request-response service work flow.

Figure 5 Typical Request-Response Service Flow



Request-Response Service Features

- The adapter request-response service supports the Tuxedo synchronous and asynchronous communication paradigms.
- The `FML32`, `String` and `CArray` buffer types are supported.
- Multithreading is available for synchronous communication. Multithreading allows the adapter to process multiple messages concurrently ensuring better performance.
- Multiple connections to the Tuxedo server are supported.

Example Scenario

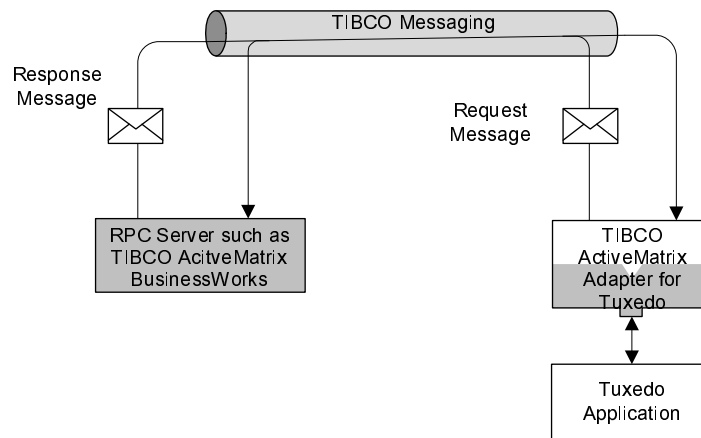
An external system publishes a message in the TIBCO environment, which is mapped to the adapter's request schema. The adapter configuration receives this message and invokes the specified Tuxedo service using the synchronous communication paradigm. The data returned by the service is mapped to the adapter's reply schema and returned to the requesting external system via the TIBCO environment.

Request-Response Invocation Service

A Request-Response Invocation Service is used to request data from external applications through the TIBCO environment. The adapter receives a request from the Tuxedo application and sends the request through the TIBCO environment. When a response is returned from the TIBCO environment, the adapter sends the response back to the application.

Figure 6 illustrates a request-response service work flow.

Figure 6 Typical Request-Response Invocation Service Flow



Request-Response Invocation Service Features

- A request-response service uses the Rendezvous Adapter Agent to communicate with the adapter.
- The FML32, String and CArray buffer types are supported.
- Multithreading is available. Multithreading allows the adapter to process multiple messages concurrently ensuring better performance.

Example Scenario

The Tuxedo system may be configured to query for customer information from an external system. The Tuxedo system invokes the adapter configuration, which sends out a request and receives a reply from the external system. The adapter then sends the reply to the Tuxedo system.

This chapter introduces the required and optional TIBCO infrastructure tools that work with an adapter.

Topics

- [TIBCO Runtime Agent, page 32](#)
- [TIBCO Designer, page 33](#)
- [TIBCO Administrator, page 34](#)
- [TIBCO ActiveMatrix BusinessWorks, page 37](#)
- [TIBCO Hawk, page 38](#)
- [TIBCO Business Studio, page 40](#)
- [TIBCO ActiveMatrix Administrator, page 44](#)

TIBCO Runtime Agent

The TIBCO Run-time Agent (TRA) provides basic connectivity between the adapter and other TIBCO infrastructure tools. The TRA is required on any machine on which an adapter is installed. The TRA runs on each machine on which an adapter runs and executes scripts, sends alerts, and performs recovery as specified.

The TRA has two main functions:

- Supplies an agent that runs in the background on each machine.
 - The agent is responsible for starting and stopping processes that run on a machine according to the deployment information.
 - The agent monitors the machine. That information is then visible via the TIBCO Administrator GUI.
- Supplies the run-time environment, that is, all shared libraries including third-party libraries required by the adapter.

TIBCO Domain Utility

The TRA contains the TIBCO Domain Utility, which is used to manage the components available on a TIBCO administration domain. The utility allows you to:

- Add or remove a machine to a TIBCO administration domain.
- Add or remove the TIBCO Enterprise Message Service server plug-in to a TIBCO administration domain.
- Change TIBCO Rendezvous parameters. Changing TIBCO Rendezvous parameters is an advanced option performed only by users familiar with TIBCO Rendezvous. If you perform this task, you must perform it on each machine in the TIBCO administration domain, then restart the TIBCO Administration Server.
- Change TIBCO administration domain credentials. Changing domain credentials is an advanced option. You must perform it on the machine where the TIBCO Administration Server is installed.
- Remove a secondary TIBCO Administration Server.
- Enable TIBCO administration domain and security management on a machine where TIBCO Administrator has been installed.
- Migrate previous TIBCO Administrator installations.

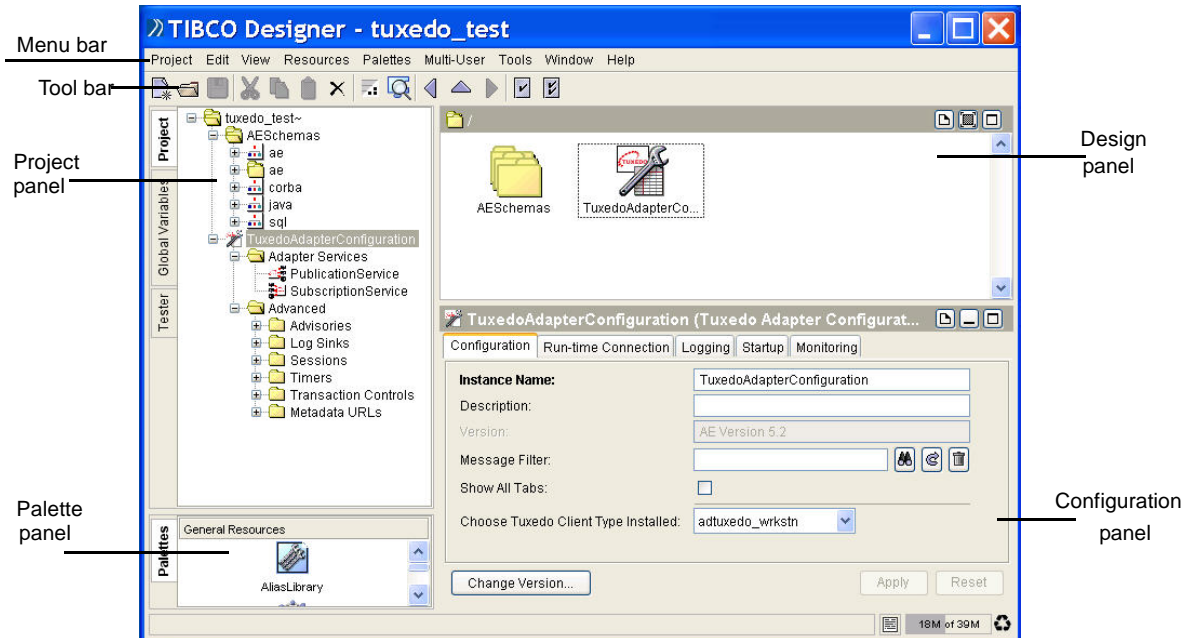
TIBCO Designer

TIBCO Designer provides the design-time environment for configuring a standalone adapter project. Using Designer, you create a project, add adapter services to it with a simple drag-and-drop interface, and specify the configuration information for each adapter service.

Before using TIBCO Designer, make sure you read the *TIBCO Designer*. The documentation can be accessed via the TIBCO Designer **Help > Designer Help** from the menu bar. The next diagram shows the TIBCO Designer interface.

The standalone adapter adds a palette to the TIBCO Designer environment which provides the adapter specific resources

Figure 7 TIBCO Designer main window



TIBCO Administrator

TIBCO Administrator provides user, resource, and application management modules for adapters.

- **User Management.** This module allows you to set permissions for adapter users. You define authentication, users and groups, and assign access control lists to users. This includes security for server-based projects at design-time and for deployed applications at runtime.
- **Resource Management.** This module allows you to monitor machines and all running applications in a TIBCO administration domain. Alerts can be created, for example, to notify an administrator if the number of processes or disk usage exceed a certain number.
- **Application Management.** This module allows you to upload Enterprise Archive (EAR) files, and create, configure, and deploy adapters. This console is also used to start and stop adapters.
- **Load balancing.** An adapter can be served by a primary and secondary TIBCO Administration Server. The primary server allows read and write operations, while the secondary server supports read operations. Load balancing is implemented through the use of the TIBCO Rendezvous distributed queue protocol (RVDQ) and therefore not available for HTTP.

To get the load balancing benefit with HTTP, you must either use an IP redirector or explicitly point to a backup server to be used when a server fails. See your IP Redirector or HTTP Server documentation for information on how to do this.

- **Failure recovery.** You can use a load-balanced TIBCO Administration Server for failure recovery. In a completely trusted environment, you can instead use a database back-end for your server and use checkpoints in the database for failure recovery.

TIBCO Administration Domain

A TIBCO administration domain is installed only if you have also installed the User Management module.

A *TIBCO administration domain* is a collection of users, machines, and components that an administration server manages. There is only one Administration Server for each administration domain. Components within an administration domain can communicate with systems outside of the domain, but the domain is the administrative boundary of your enterprise integration project.

Each TIBCO administration domain contains one or more machines. By default, all machines within an administration domain are expected to be in the same network subnet. You can, however, set up your system to use TIBCO Rendezvous rrvd and can then use the components across subnets. See the *TIBCO Administrator Server Configuration Guide* for details.

Each machine can belong to only one TIBCO administration domain. This is similar to a Microsoft Windows network domain where your machine can also belong to only one network domain.

TIBCO Administration Server

The TIBCO Administrator Server provides a central storage and distribution point for configuration data and schema data needed by an adapter. The server is included in both Administrator editions.

Each administration domain has one and only one TIBCO Administration Server. The *TIBCO Administration Server* is the machine process that handles the stored project and requests to manage the TIBCO administration domain.

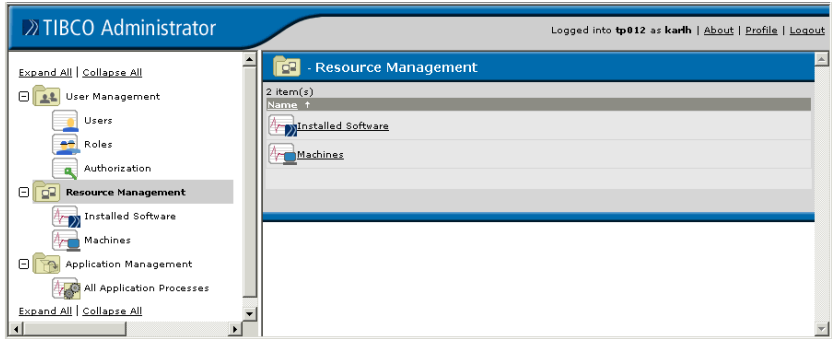
The TIBCO Administrator Server contains its own web server (Apache Tomcat) that can be accessed via the TIBCO Administrator GUI for configuration and monitoring information.

The TIBCO Administration Server supports centralized authentication and authorization. Using the TIBCO Administrator GUI, users with full administrative privileges can define who has access to projects that are managed by the repository server.

TIBCO Administrator GUI

You can access the TIBCO Administration Server using the web-based TIBCO Administrator GUI. The GUI allows you to create users and assign access to projects managed by the Administration Server. You can invoke the GUI from any machine in a TIBCO administration domain. The next diagram shows the GUI.

Figure 8 TIBCO Administrator GUI



TIBCO ActiveMatrix BusinessWorks

TIBCO ActiveMatrix BusinessWorks is a scalable, extensible, and easy to use integration platform that allows you to develop integration projects. TIBCO ActiveMatrix BusinessWorks includes a graphical user interface for defining business processes and an engine that executes the process.

In TIBCO ActiveMatrix BusinessWorks, adapter services are responsible for publishing or subscribing to business data in a decoupled yet reliable manner. The business process receives data from an adapter service and routes data to an adapter service.

TIBCO ActiveMatrix BusinessWorks provides the following activities for use with adapters:

- **Publish to Adapter**—Publishes data from the process to an adapter, which subscribes to data coming from the process and passes the data to the target application.
- **Adapter Subscriber**—Subscribes to incoming data published by the adapter.
- **Invoke an Adapter Request-Response Service**—Communicates (as a client) with an adapter request-response service.
- **Adapter Request-Response Server**—Starts a process based on the receipt of a request from an adapter.
- **Respond to Adapter Request**—Sends a response to an adapter for a previously received request.
- **Wait for Adapter Message**—Waits for the receipt of a message from the publication service of the specified adapter.
- **Wait for Adapter Request**—Waits for the receipt of a request from a request-response invocation service.

See the TIBCO ActiveMatrix BusinessWorks documentation for more information.

TIBCO ActiveMatrix BusinessWorks Service Engine

The product provides an ActiveMatrix container to deploy ActiveMatrix BusinessWorks projects using TIBCO ActiveMatrix Administrator. TIBCO ActiveMatrix BusinessWorks supports service oriented processing with the use of service resources, partners, and partner bindings.

See the TIBCO ActiveMatrix BusinessWorks Service Engine documentation for more information.

TIBCO Hawk

TIBCO Hawk monitors and manages distributed applications and systems throughout the enterprise. System administrators can monitor application parameters, behavior, and loading activities for all nodes in a local or wide-area network and take action when pre-defined conditions occur. In many cases, run-time failures or slowdowns can be repaired automatically within seconds of their discovery, reducing unscheduled outages, and slowdowns of critical business systems.

TIBCO Hawk features include:

- Extensive monitoring capabilities at the operating system and application levels including process data, disk, and CPU utilization, network statistics, log, and system files
- Built-in routines within other TIBCO ActiveEnterprise components allow for proactive management. Problems to be found and fixed before failure can occur.
- Hawk Application Management Interface (AMI) routines can be embedded within custom adapters, allowing active management of those adapters by the Hawk micro-agent
- Distributed micro-agents support autonomous network behavior so local management and problem resolution can continue during an outage
- Fault-tolerance is achieved through the independent operation of Hawk agents, which continue to perform local tasks even in the event of network failure

TIBCO Hawk consists of several components: a console display, a central repository for storage of configuration objects, agents, and microagents whose monitoring duties are defined by the rule bases.

- Agents monitor local conditions and take action or publish alert information that appears in the TIBCO Hawk display.
- Microagents act as an interface to the managed objects and are invoked through their supported methods.

Adapter Microagents

Each adapter includes a standard and custom microagent. The microagents provide:

- Business level statistics—statistics that report the progress of the adapter as it interacts with the vendor application. For example, in a database adapter such

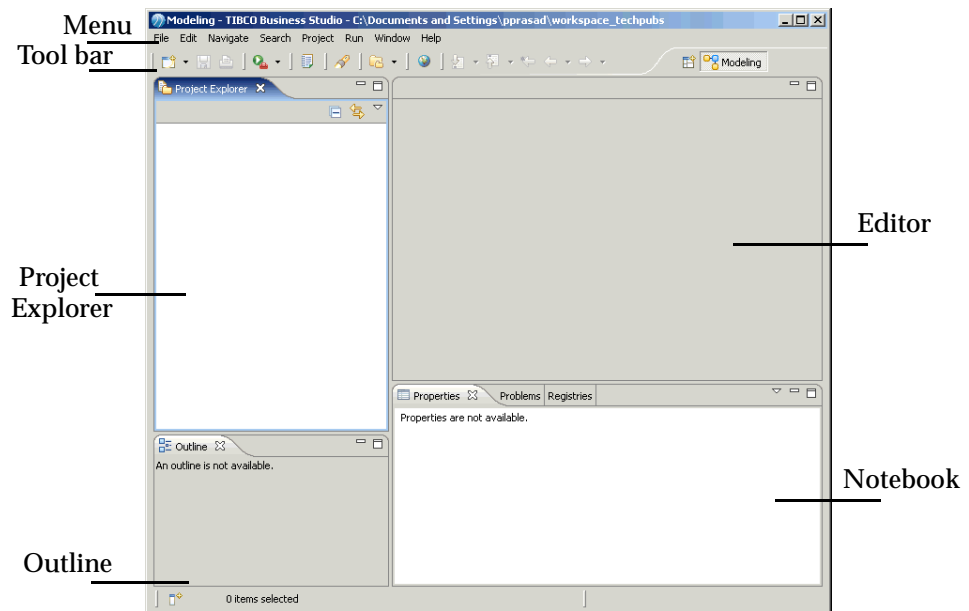
statistics might indicate whether objects were successfully or unsuccessfully inserted, updated, or deleted in the database.

- Queries that return information about the state of the adapter. This can be an important tool for seeing the internals of an adapter and debugging it if something appears wrong. For example, methods can return information about threads, internal queues, or connections to the target system. Using these methods, one might be able to identify certain bottlenecks or gauge how successfully an adapter is scaling with respect to the current environment.
- Updates of the adapter runtime parameters. This includes retrieving the current runtime parameters and setting new runtime parameters without restarting the adapter. An example of this is getting and setting the polling interval. Updating a runtime parameter through the Hawk microagent only affects the setting of the instance that is running.

TIBCO Business Studio

The TIBCO ActiveMatrix development tools consist of TIBCO Business Studio and a set of ActiveMatrix plug-ins. For introductory information on TIBCO Business Studio, refer to the *Workbench User Guide* in the TIBCO Business Studio online help. To view the online help, select **Help > Help Contents**.

Figure 9 TIBCO Business Studio



The screen contains the following area and views:

- **Menu** See Help > Help Contents > *Workbench User Guide*.
- **Tool bar** See Help > Help Contents > *Workbench User Guide*.
- **Project Explorer** Displays a tree containing all the project resources such as project folders, shared resource definition files, WSDL files, composite files, service assembly files, and so on.
- **Editor** Displays editors for the objects currently being edited. You switch between editors by clicking tabs at the top of the Editor area. The Composite Editor contains a canvas on which you can drop elements and a palette that organizes the elements that you can add to the composite. Other editors allow you to configure shared resources and service assemblies.

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

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

- **Views** Displays under the Editor Area. Contains the following views:
 - **Properties** Displays property sheets for editing composites and composite elements. When you select a composite or composite element in the Composite Editor canvas, this view shows the properties of the selected object in a vertical tabbed notebook.
 - **Problems** Displays validation and other errors.
 - **Registries** Lists UDDI registries and the WSDL files returned from searching a registry.

You open a view by selecting **Window > Show View > View**.

ActiveMatrix Resource Wizard

The starting point for creating all types of ActiveMatrix projects and assets is the ActiveMatrix Resource Wizard. The resource wizard allows you to select wizards to create:

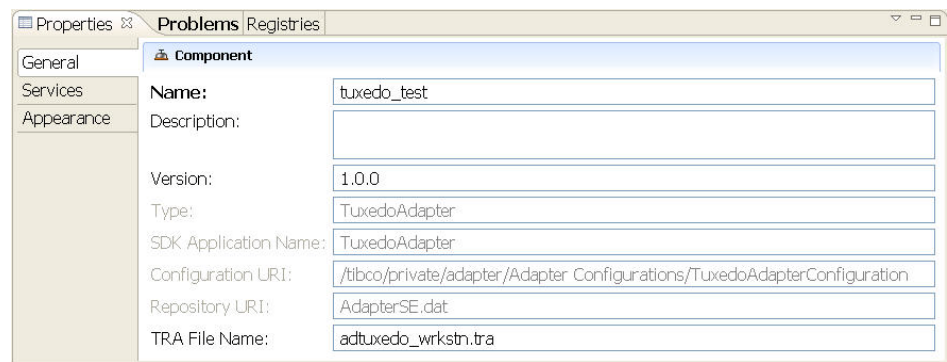
- Adapter service engine projects
- ActiveMatrix sample projects
- ActiveMatrix SOA projects
- Composites
- Mediation flows
- Service assemblies

Composite Element Editors

Composite elements are configured in property sheets accessed through the Properties view.

[Figure 10](#) shows the an example of the Properties view.

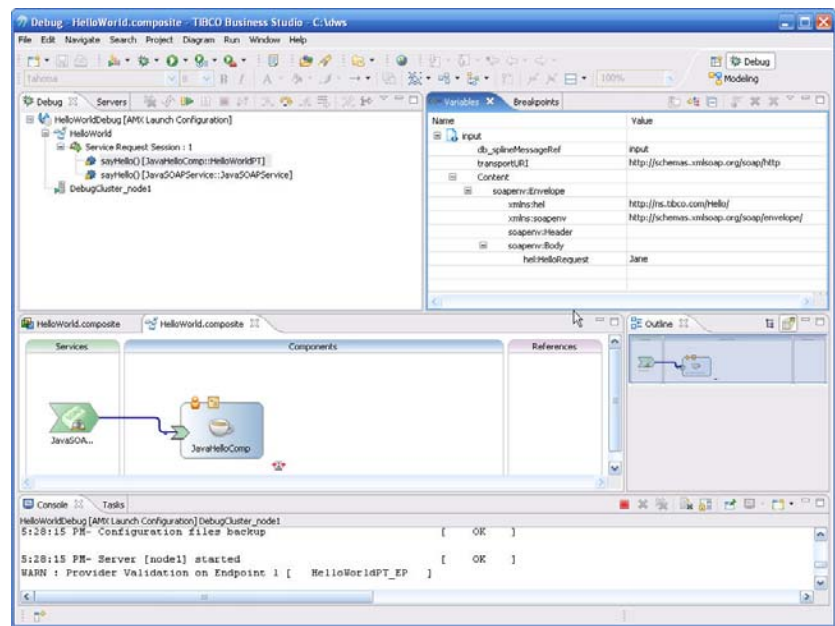
Figure 10 Component Property Sheet



Debugger

The TIBCO Business Studio debugger provides a testing environment for stepping through composite elements and determining the sources of errors. [Figure 11](#) shows the debugger in the process of debugging a sample HelloWorld composite. Breakpoints have been set before and after the Java component executes, and the debugger is stopped at the before breakpoint. In the Variables view on the top-right, the value of the request is being examined.

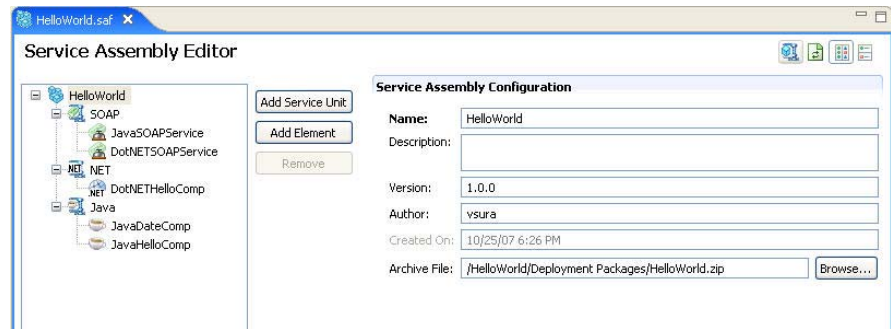
Figure 11 Debugger



Service Assembly Editor

In order to be deployed, composites must be transformed into service units and service assemblies. [Figure 12](#), shows a sample service assembly editor view.

Figure 12 Service Assembly Editor



TIBCO ActiveMatrix Administrator

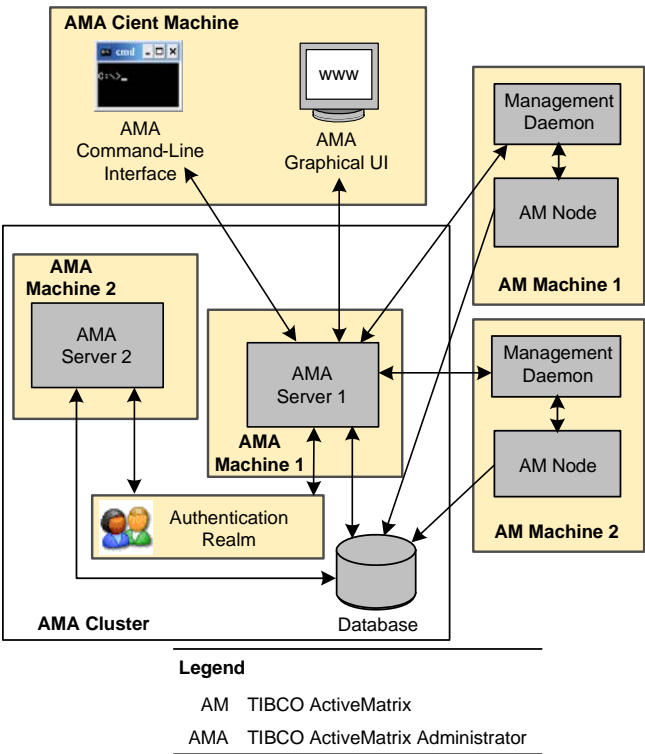
ActiveMatrix provides TIBCO ActiveMatrix Administrator for enterprise, environment, and service management. ActiveMatrix Administrator supports both graphical and command-line interfaces.

The following sections provide an overview of TIBCO ActiveMatrix Administrator. To get a quick introduction to the administration tools in practice, see *TIBCO ActiveMatrix Service Grid Getting Started*. For detailed information about the administration tools, see TIBCO Hawk Administrator’s Guide.

TIBCO ActiveMatrix Administrator Architecture

Figure 13 on page 44 shows ActiveMatrix Administrator components, and the relationship between ActiveMatrix Administrator, other servers, and ActiveMatrix machines and nodes.

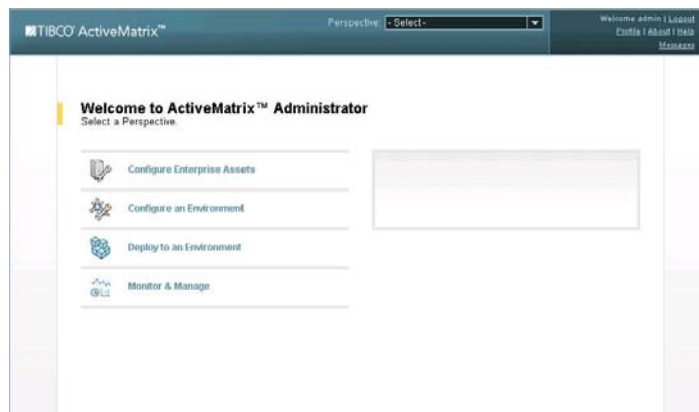
Figure 13 TIBCO ActiveMatrix Administration Architecture



The TIBCO ActiveMatrix Administrator administration architecture consists of the following components:

- **TIBCO ActiveMatrix Administrator Server** Gathers management data from nodes, responds to requests from the ActiveMatrix Administrator graphical and command-line UIs, interacts with the authentication realm server to authenticate users, and interacts with TIBCO Management Daemon to manage nodes.
- **TIBCO ActiveMatrix Administrator Cluster** Groups one or more ActiveMatrix Administrator servers. ActiveMatrix Administrator servers within a cluster share a database and authentication realm and are kept synchronized.
- **ActiveMatrix Database** Stores ActiveMatrix administration data.
- **Authentication Realm** Manages user authentication data. The authentication realm can be provided either by TIBCO Administrator or by another server or a file.
- **ActiveMatrix Administrator Graphical UI** Displays the ActiveMatrix Administrator user interface. [Figure 14 on page 45](#) shows the ActiveMatrix Administrator graphical UI welcome page. In ActiveMatrix Administrator, functionality is divided into perspectives. A *perspective* is a set of controls used to carry out a category of administration tasks.
- **ActiveMatrix Administrator Command-Line Interface** Provides a script-based interface to ActiveMatrix Administrator functions.
- **Management Daemon** Gathers installation information and exposes ActiveMatrix node life cycle operations.

Figure 14 TIBCO ActiveMatrix Administrator



Enterprise and Environment Administration

You administer ActiveMatrix enterprises and environments, shared resource configurations, nodes, containers, and managed resources with ActiveMatrix Administrator graphical and command-line interfaces.

In the graphical interface, enterprise and environment administration is carried out in the Configure Enterprise Assets and Configure an Environment perspectives.

Service Administration

You administer ActiveMatrix services with ActiveMatrix Administrator graphical and command-line interfaces. Service administration consists of deployment tasks and monitoring and management tasks. In the graphical interface, these tasks are carried out in the Deploy to an Environment and Monitor & Manage perspectives.

Service Deployment

The first phase of service administration is deployment. During deployment, the service units within a service assembly are mapped and then deployed into their respective containers, the services provided by the service units are registered with the ActiveMatrix container, and the service endpoints are activated.

The choice of how to distribute services across nodes is determined by the desired level of service performance and availability. Service performance and availability can be enhanced if you deploy a service unit across multiple nodes, which allows Messaging Bus to distribute requests between the service instances.

Load Balanced Services

The Mediation Bus enables load balancing at the container level by defining a container group. A *container group* is defined as a group containing one or more container instances of the same container type. When a service unit is deployed to a container group, a copy of the service unit is deployed into each container in the group.

Highly Available Services

Services deployed on multiple containers are *highly available*; if one container fails, service requests will be handled by one of the remaining containers. No configuration is required to make services highly available. Messaging Bus automatically routes to any available service instance identified in the message exchange.

Load Balanced Services

Requests to services deployed on multiple containers are *load balanced* between the available providers. No configuration is required to load balance between services. Messaging Bus uses a round robin algorithm for routing requests to service instances.

Service Monitoring and Management

TIBCO ActiveMatrix Administrator not only allows you to configure and deploy services, but also lets you monitor and manage the deployed services. Monitoring the system performance is not a one time activity but needs to be performed on a day-to-day basis. The Monitor & Manage perspective of TIBCO ActiveMatrix Administrator keeps track of system health without much overhead.

In the Monitor & Manage perspective you can monitor the overall health and performance of the grid infrastructure, applications, and services. You can monitor performance at various levels such as environment, machine, node, container, service assembly, and service unit.

The monitoring subsystem uses content-based metrics to measure the service performance, availability of services, service usage, and the number of successful to faulty service responses. These metrics provide real-time values by fetching data every minute and updating the values of the metrics. The real-time data is then displayed in a web-based dashboard provided with pre-defined views and visual alerts.

Appendix A **Encoding Tables**

This appendix lists the encoding values that can be used for the adapter.

Topics

- [Encoding Tables, page 50](#)

Encoding Tables

The encoding tables include the following table headings:

- Encoding Value in Pick List
This column lists all adapter Encoding field display names that are available from the field's pick list at design time. Note that the column lists potential values for all adapters. Your adapter likely displays a subset of these values.
- IBM CCSID
The IBM CCSID column represents the character code set identifier (CCSID) given to the ISO code page in IBM publications.
- Oracle NLS String
The Oracle National Language Support (NLS) string. Some multi-nation character sets require the LANGUAGE_TERRITORY prefix before the Oracle NLS_LANG value. Check with Oracle for details.
- Description
Description of the encoding value.

The next table list the ISO character sets.

Table 5 ISO Character Sets

Value in Encoding Field Pick List	IBM CCSID	Oracle NLS String	Description
ASCII	IBM-367	US7ASCII	7-bit ASCII
ISO-8859-1	IBM-819	language_territory.WE8ISO8859P1	ISO8859-1 (Latin-1), West European
ISO-8859-2	IBM-912	language_territory.EE8ISO8859P2	ISO8859-2 (Latin-2), East European
ISO-8859-3	IBM-913	language_territory.SE8ISO8859P3	ISO8859-3 (Latin-3), South European
ISO-8859-4	IBM-914	language_territory.NEE8ISO8859P4	ISO8859-4 (Latin-4), North European
ISO-8859-5	IBM-915	language_territory.CE8ISO8859P5	ISO8859-5, Cyrillic

Table 5 ISO Character Sets

Value in Encoding Field Pick List	IBM CCSID	Oracle NLS String	Description
ISO-8859-6	IBM-1089	ARABIC_UNITED ARAB EMIRATES.AR8ISO 8859P6	ISO8859-6, Arabic
ISO-8859-7	IBM-813	GREEK_GREECE.E L8ISO8859P7	ISO8859-7, Greece
ISO-8859-8	IBM-916	HEBREW_ISRAEL.I W8ISO8859P8	ISO8859-8, Hebrew
ISO-8859-9	IBM-920	TURKISH_TURKEY .WE8ISO8859P9	ISO8859-9 (Latin-5), Turkey
ISO-8859-13	IBM-921	N/A	ISO8859-13, Baltic
ISO-8859-15	IBM-923	N/A	ISO8859-15 (Latin-9), Latin-1 with Euro
Shift_JIS (CP943)	IBM-943	JAPANESE_JAPAN. JA16SJIS	Japanese Shift-JIS, CP943
KSC_5601	IBM-949	KOREAN_KOREA. KO16KSC5601	Korean KSC-5601
Big5	IBM-1370	TRADITIONAL CHINESE_TAIWA N.ZHT16BIG5	Traditional Chinese Big5 (with Euro Sign)
GBK	IBM-1386	SIMPLIFIED CHINESE_CHINA. ZHS16CGB231280	Simplified Chinese GBK, super set of GB2312-80
EUC-JP	IBM-954	JAPANESE_JAPAN. JA16EUC	Japanese EUC
EUC-KR	IBM-970	KOREAN_KOREA. KO16KSC5601	Korean EUC
EUC-CN	IBM-1383	SIMPLIFIED CHINESE_CHINA. ZHS16CGB231280	Simplified Chinese EUC, compatible with GB2312-80

Table 5 ISO Character Sets

Value in Encoding Field Pick List	IBM CCSID	Oracle NLS String	Description
Shift_JIS (TIBCO)	N/A	JAPANESE_JAPAN. JA16SJIS	Variant of IBM-943, flavoring some MS-932 conversions
Shift_JIS(932)	IBM-932	JAPANESE_JAPAN. JA16SJIS	Shift_JIS encoding in MS-932 flavour
UTF8	IBM-1208	AMERICAN_AMER ICA.UTF8	Unicode Transformation Format-8
UTF16_BigEndian	N/A	N/A	UTF16 (UCS-2) Big Endian
UTF16_Little Endian	N/A	N/A	UTF16 (UCS-2) Little Endian

The next table lists the EBCDIC character sets.

Table 6 EBCDIC Character Sets

Encoding Value	IBM CCSID	Oracle NLS String	Description
IBM-37	IBM-37	N/A	EBCDIC United States
IBM-273	IBM-273	N/A	EBCDIC Germany, Austria
IBM-277	IBM-277	N/A	EBCDIC Denmark, Norway
IBM-278	IBM-278	N/A	EBCDIC Finland, Sweden
IBM-280	IBM-280	N/A	EBCDIC Italy
IBM-284	IBM-284	N/A	EBCDIC Spain
IBM-285	IBM-285	N/A	EBCDIC UK Ireland
IBM-297	IBM-297	N/A	EBCDIC France
IBM-420	IBM-420	N/A	EBCDIC Arabic
IBM-424	IBM-424	N/A	EBCDIC Hebrew

Table 6 EBCDIC Character Sets

Encoding Value	IBM CCSID	Oracle NLS String	Description
IBM-500	IBM-500	N/A	EBCDIC Latin-1 (International)
IBM-1148	IBM-1148	N/A	EBCDIC Latin-1 (with Euro Sign)
IBM-1047	IBM-1047	N/A	EBCDIC Open System Latin-1
IBM-1153	IBM-1153	N/A	EBCDIC Latin-2 (with Euro Sign)
IBM-1154	IBM-1154	N/A	EBCDIC Cyrillic
IBM-1156	IBM-1156	N/A	EBCDIC Baltic
IBM-1157	IBM-1157	N/A	EBCDIC Estonia
IBM-290	IBM-290	N/A	EBCDIC Japanese Katakana (SBCS).
IBM-1159	IBM-1159	N/A	EBCDIC Traditional Chinese (SBCS)
IBM-300	IBM-300	N/A	EBCDIC Japanese (DBCS)
IBM-834	BM-834	N/A	EBCDIC Korean (DBCS)
IBM-835	IBM-835	N/A	EBCDIC Traditional Chinese (DBCS)
IBM-837	IBM-837	N/A	Forward compatible with IBM-935
IBM-930	IBM-930	N/A	EBCDIC Japanese (MBCS)
IBM-933	IBM-933	N/A	EBCDIC Korean (MBCS)
IBM-935	IBM-935	N/A	EBCDIC Simplified Chinese (MBCS)
IBM-937	IBM-937	N/A	EBCDIC Traditional Chinese (MBCS)
IBM-1390	IBM-1390	N/A	EBCDIC Japanese (MBCS) with Euro

Table 6 EBCDIC Character Sets

Encoding Value	IBM CCSID	Oracle NLS String	Description
IBM-1364	IBM-1364	N/A	EBCDIC Korean (MBCS) with Euro
IBM-1371	IBM-1371	N/A	EBCDIC Traditional Chinese (MBCS) with Euro

The next table lists the Microsoft Windows character sets

Table 7 Microsoft Windows Character Sets

Encoding Value	IBM CCSID	Oracle NLS Encoding	Description
Windows Latin-1	IBM-1252	language_territory.WE8MSWIN1252	Windows Latin-1 with Euro Sign
Windows Latin-2	IBM-1250	language_territory.EE8MSWIN1250	Windows Latin-2 with Euro Sign
Windows Cyrillic	IBM-1251	language_territory.CL8MSWIN1251	Windows Cyrillic with Euro Sign
Windows Greek	IBM-1253	GREEK_GREECE.EEL8MSWIN1253	Windows Greek with Euro Sign
Windows Turkish	IBM-1254	TURKISH_TURKEY.TR8MSWIN1254	Windows Turkish with Euro Sign
Windows Hebrew	IBM-1255	HEBREW_ISRAEL.IW8MSWIN1255	Windows Hebrew with Euro Sign
Windows Arabic	IBM-1256	ARABIC_UNITED_ARAB_EMIRATES.AR8MSWIN1256	Windows Arabic with Euro Sign
Windows Baltic	IBM-1257	language_territory.BLT8MSWIN1257	Windows Baltic with Euro Sign

Index

A

- ActiveMatrix Administrator cluster [45](#)
- ActiveMatrix Administrator Command-Line Interface [45](#)
- ActiveMatrix Administrator Graphical UI [45](#)
- ActiveMatrix Administrator server [45](#)
- ActiveMatrix Database [45](#)
- ActiveMatrix resource wizard [41](#)
- Adapter
 - service engine [3](#)
 - standalone [3](#)
- Adapter Agent [19](#)
- Adapter Infrastructure Tools
 - TIBCO ActiveMatrix Administrator [44](#)
 - architecture [44](#)
 - Enterprise and Environment Administration [46](#)
 - Load Balanced Services [46](#)
 - TIBCO ActiveMatrix BusinessWorks [37](#)
 - TIBCO ActiveMatrix BusinessWorks Service Engine [37](#)
 - TIBCO Administrator [34](#)
 - TIBCO Administrator
 - Administration Domain [34](#)
 - Administration Server [35](#)
 - Administrator GUI [35](#)
 - Application Management [34](#)
 - Failure recovery [34](#)
 - Load balancing [34](#)
 - Resource Management [34](#)
 - User Management [34](#)
 - TIBCO Busindess Studio [40](#)
 - TIBCO Designer [33](#)
 - TIBCO Hawk [38](#)
 - TIBCO Runtime Agent [32, 32](#)
 - TIBCO Domain Utility [32](#)

- Adapter Life Cycle
 - Configuration [9](#)
 - Deployment [9](#)
 - Installation [9](#)
 - Monitoring [9](#)
- adapter service
 - Supported Tuxedo Communication Paradigms [18](#)
- Adapter Services [24](#)
- Adapter Terms
 - .ear file [4](#)
 - adapter container [4](#)
 - business objects [4](#)
 - Message Exchange Pattern (MEP) [4](#)
 - palette [4](#)
 - project [4](#)
 - service assembly [4](#)
 - SOA [4](#)
 - WSDL [4](#)
- AMX_HOME [xii](#)
- ATMI [16](#)
- authentication realm [45](#)

B

- BEA Tuxedo [16](#)

C

- CArray [16](#)
- Communication Paradigms [17](#)
- composite element editors [41](#)

Connection Paradigms in Tuxedo [22](#)
 Asynchronous Communication [17, 22](#)
 Conversational Communication [17, 22](#)
 Event-based Communication [17](#)
 Queue-based Communication [17, 22](#)
 Synchronous Communication [17, 22](#)
 customer support [xv](#)

E

Encoding Tables [50](#)
 ENV_HOME [xii](#)
 EventBroker [17, 19](#)
 Example Scenario for event-based Publication [25](#)
 Example Scenario for RV agent-based publication [26](#)

F

FML32 Buffer Type [16, 22](#)

H

highly available services [46](#)

I

Internationalization [10](#)

J

JMS Adapter Agent [20](#)

L

load balanced services [47](#)

M

Management Daemon [45](#)

P

Publication Service [18, 18, 25, 25](#)
 Publication Service typical flow diagram [25](#)

R

Rendezvous [19](#)
 Rendezvous Adapter Agent [19](#)
 Request-Response Invocation Service [30](#)
 Request-Response Service [28](#)
 RPC server used by the adapter [28](#)
 RV-based Adapter Agent [19](#)

S

service administration [46](#)
 service and binding definition editor [41](#)
 service assembly deployment [46](#)
 service assembly editor [43](#)
 service deployment [46](#)
 service monitoring and management [47](#)
 STRING Buffer [16](#)
 Subscriber
 Multi-threading [22](#)
 Subscription Service [26](#)
 support, contacting [xv](#)

T

technical support [xv](#)
TIBCO Resource Wizard [41](#)
TIBCO_HOME [xii](#)
TRA [32](#)
Tuxedo [16](#)
Tuxedo Client [17](#)
Tuxedo Queue [21](#)
Tuxedo Server [17](#)