

TIBCO® Adapter SDK

Concepts

*Software Release 5.8
November 2011*

TIBCO provides the two-second advantage™



Important Information

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

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

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

TIBCO, The Power of Now, TIBCO Administrator, TIBCO Designer, TIBCO Enterprise, TIBCO Integration Manager, TIBCO Rendezvous, and TIBCO Repository are either registered trademarks or trademarks of TIBCO Software Inc. in the United States and/or other countries.

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

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

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

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

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

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

Copyright © 1998-2011 TIBCO Software Inc. ALL RIGHTS RESERVED.

TIBCO Software Inc. Confidential Information

Contents

Figures	v
Preface	vii
Changes from the Previous Release of this Guide	viii
Related Documentation	ix
TIBCO Adapter SDK Documentation	ix
Other TIBCO Product Documentation	ix
Typographical Conventions	x
Connecting with TIBCO Resources	xiii
How to Join TIBCOCommunity	xiii
How to Access All TIBCO Documentation	xiii
How to Contact TIBCO Support	xiii
Chapter 1 Introduction	1
Overview	2
TIBCO Adapter SDK Features	3
TIBCO Adapters and TIBCO Adapter SDK	4
Adapter SDK Class Library	4
Benefits of Using TIBCO Adapter SDK	4
TIBCO Adapter SDK and TIBCO ActiveEnterprise	5
TIBCO Adapters and Other TIBCO Applications	8
Data Transformation Facilities	8
TIBCO ActiveMatrix BusinessWorks Adapter Palette	9
Adapter Interactions	10
Publish/Subscribe Interactions	10
Request/Response Interactions	11
Chapter 2 Designing and Implementing Custom Adapters	13
Overview	14
Requirements Definition	15
Adapter Configuration	16
Basic Adapter Configuration	16
Schema Configuration	17
Global Variables	18

- Saving Adapter Configuration 19
- Adapter SDK Programming..... 20
 - Programming Features 20
 - Program Elements 20
 - Data at Runtime..... 21
- Tracing and Tracking..... 23
 - Tracing..... 23
 - Tracking..... 24
- Internationalization 25
 - SDK-Internal C++ Unicode Type Conversion 25
 - Specifying the Wire Format Encoding 25
- Index 27**

Figures

Figure 1 Adapter Data Flow 2

Figure 2 TIBCO ActiveEnterprise Core Components and Integration Applications 5

Figure 3 An Adapter Subscriber Activity Receiving a Message from an Adapter Publisher. 9

Figure 4 Publication Service 10

Figure 5 Subscription Service 10

Figure 6 Request-Response Interaction. 11

Figure 7 Metadata Creation and Usage 18

Figure 8 Data at Runtime 22

Preface

TIBCO Adapter SDK (Software Development Kit) is a class library that facilitates adapter development. All adapters implemented using TIBCO Adapter SDK have the same external interface and consistently plug in to the overall TIBCO ActiveEnterprise product suite.

Topics

- [Changes from the Previous Release of this Guide, page viii](#)
- [Related Documentation, page ix](#)
- [Typographical Conventions, page x](#)
- [Connecting with TIBCO Resources, page xiii](#)

Changes from the Previous Release of this Guide

There are no changes from the previous release of this guide.

Related Documentation

This section lists documentation resources you may find useful.

TIBCO Adapter SDK Documentation

The following documents form the TIBCO Adapter SDK documentation set:

- *TIBCO Adapter SDK Concepts* Read this manual before reading any other book in the documentation set to familiarize yourself with the product and its use.
- *TIBCO Adapter SDK Installation* Read this manual for instructions on site preparation and installation.
- *TIBCO Adapter SDK Programmer's Guide* Read this manual for details on implementing a custom adapter. This manual also discusses configuration and programming, and provides example code fragments.
- *TIBCO Adapter SDK Status Codes* A reference for the message codes used by TIBCO Adapter SDK.
- *TIBCO API Reference* Provides online documentation for the exposed interfaces, classes, and methods of the TIBCO Adapter C++ and Java APIs.
- *TIBCO Adapter SDK Release Notes* Read the release notes for a list of new and changed features. This document also contains lists of closed 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 ActiveMatrix BusinessWorks™
- TIBCO ActiveEnterprise™
- TIBCO Designer™
- TIBCO Administrator™
- TIBCO Rendezvous®
- TIBCO Enterprise Message Service™
- TIBCO Hawk®
- TIBCO Runtime Agent™

Typographical Conventions

The following typographical conventions are used in this manual.

Table 1 General Typographical Conventions

Convention	Use
<i>ENV_NAME</i> <i>TIBCO_HOME</i> <i>SDK_HOME</i>	<p>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.</p> <p>An installation environment consists of the following properties:</p> <ul style="list-style-type: none">• Name Identifies the installation environment. This name is referenced in documentation as <i>ENV_NAME</i>. On Microsoft Windows, the name is appended to the name of Windows services created by the installer and is a component of the path to the product shortcut in the Windows Start > All Programs menu.• Path The folder into which the product is installed. This folder is referenced in documentation as <i>TIBCO_HOME</i>. <p>TIBCO Adapter SDK is installed into a directory within a <i>TIBCO_HOME</i>. This directory is referenced in documentation as <i>SDK_HOME</i>. The default value of <i>SDK_HOME</i> depends on the operating system. For example, on Windows systems, the default value is C:\tibco\adapter\SDK\version_number.</p>
code font	<p>Code font identifies commands, code examples, filenames, pathnames, and output displayed in a command window. For example:</p> <p>Use MyCommand to start the foo process.</p>
bold code font	<p>Bold code font is used in the following ways:</p> <ul style="list-style-type: none">• In procedures, to indicate what a user types. For example: Type admin.• In large code samples, to indicate the parts of the sample that are of particular interest.• In command syntax, to indicate the default parameter for a command. For example, if no parameter is specified, MyCommand is enabled: MyCommand [enable disable]

Table 1 General Typographical Conventions (Cont'd)




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 PathName</code>
Key combinations	<p>Key names separated by a plus sign indicate keys pressed simultaneously. For example: Ctrl+C.</p> <p>Key names separated by a comma and space indicate keys pressed one after the other. For example: Esc, Ctrl+Q.</p>
	The note icon indicates information that is of special interest or importance, for example, an additional action required only in certain circumstances.
	The tip icon indicates an idea that could be useful, for example, a way to apply the information provided in the current section to achieve a specific result.
	The warning icon indicates the potential for a damaging situation, for example, data loss or corruption if certain steps are taken or not taken.

Table 2 Syntax Typographical Conventions

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

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 param1 and param2, or the pair param3 and param4.</p> <pre>MyCommand {param1 param2} {param3 param4}</pre> <p>In the next example, the command requires two parameters. The first parameter can be either param1 or param2 and the second can be either param3 or param4:</p> <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 param1. You can optionally include param2 as the second parameter. And the last parameter is either param3 or param4.</p> <pre>MyCommand param1 [param2] {param3 param4}</pre>

Connecting with TIBCO Resources

How to Join TIBCOCommunity

TIBCOCommunity is an online destination for TIBCO customers, partners, and resident experts, a place to share and access the collective experience of the TIBCO community. TIBCOCommunity offers forums, blogs, and access to a variety of resources. To register, go to <http://www.tibcocommunity.com>.

How to Access All TIBCO Documentation

After you join TIBCOCommunity, you can access the documentation for all supported product versions here:

<http://docs.tibco.com/TibcoDoc>

How to Contact TIBCO Support

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

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

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

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

<https://support.tibco.com>

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

Chapter 1 **Introduction**

This chapter gives an overall introduction of the TIBCO Adapter SDK.

Topics

- [Overview, page 2](#)
- [TIBCO Adapter SDK Features, page 3](#)
- [TIBCO Adapters and TIBCO Adapter SDK, page 4](#)
- [TIBCO Adapter SDK and TIBCO ActiveEnterprise, page 5](#)
- [TIBCO Adapters and Other TIBCO Applications, page 8](#)
- [Adapter Interactions, page 10](#)

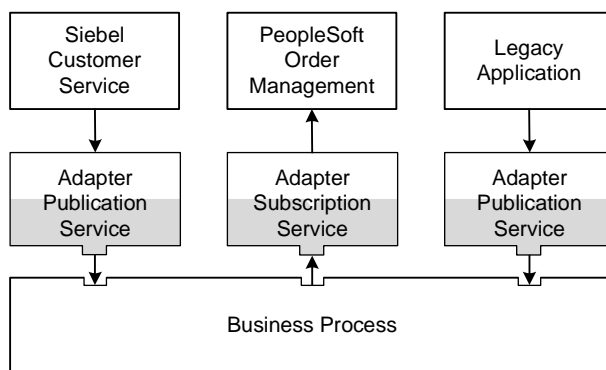
Overview

TIBCO Software Inc. offers TIBCO Adapter applications as part of the TIBCO ActiveEnterprise application integration suite.

In the TIBCO ActiveEnterprise model, adapters make it possible to communicate and update business information that originates from diverse sources and resides on diverse host systems within an organization.

TIBCO ActiveEnterprise includes adapters to popular packaged applications (for example, Siebel, SAP R/3), databases, mainframes, and more.

Figure 1 Adapter Data Flow



TIBCO Adapter SDK (Software Development Kit) is the standard toolkit to build TIBCO Adapter applications. All TIBCO adapters are developed using TIBCO Adapter SDK.

The SDK is also available for developers to create a custom adapter for an application to which no existing TIBCO Adapter products apply.

Supported Platforms

TIBCO Adapter SDK is supported in C++ and Java on Windows platforms, Solaris, HP-UX, AIX, LINUX, and other UNIX platforms. The Java API is also supported on several IBM System i platforms.

For the list of supported operating systems, see the readme file.

TIBCO Adapter SDK Features

TIBCO Adapter SDK provides everything necessary to develop standard TIBCO adapters for custom applications, including the foundation for common adapter functionality, such as sending and receiving information, configuration, management, and monitoring.

The SDK also enables interoperability with other TIBCO ActiveEnterprise products, including TIBCO ActiveMatrix BusinessWorks for data transformation and process design, TIBCO Hawk and TIBCO Administrator for monitoring and managing of adapter products, and the storage of configuration and metadata information.

SDK features are listed as follows:

- Based on technology that TIBCO, its partners, and its customers have successfully adopted to build a wide range of adapters.
- Provides easy access to standard facilities such as subject-based addressing, class of delivery service, and load balancing.
- Provides metadata specification of business events and callable functions.
- Supports multiple communication models including request/reply and publish/subscribe.
- Simplifies interoperability, management, and maintainability through shared components.
- Supports plug-in transformation modules for localized transformations.
- Supports multiple transports (TIBCO Rendezvous and TIBCO Enterprise Message Service) and wire formats (rvMsg, aeRvMsg, aeXml).

TIBCO Adapters and TIBCO Adapter SDK

A TIBCO Adapter integrates a third-party source or target application with TIBCO ActiveEnterprise applications in an event-driven, decoupled manner.

TIBCO Software Inc. has released a large number of adapter products, which can integrate applications with TIBCO Rendezvous or TIBCO Enterprise Message Service and ultimately with an event-driven enterprise. TIBCO Adapter SDK can also be used to build a custom adapter for the application to be integrated.

Adapter SDK Class Library

All TIBCO adapter products are built on top of the TIBCO Adapter SDK. If no adapter is available for an application to be integrated with an enterprise, the Adapter SDK can be used to build a custom adapter.

The Adapter SDK class library allows developers to implement an adapter for their source or target application. The SDK itself encapsulates much of the required behavior. Developers specify configuration and metadata information and implement certain methods and subclasses. The class libraries are available in C++ and Java. The two APIs are similar. Minor differences are highlighted as appropriate throughout this manual.



Undocumented or unsupported APIs should be avoided, as the use of such may result in compilation errors, runtime errors, or both.

Benefits of Using TIBCO Adapter SDK

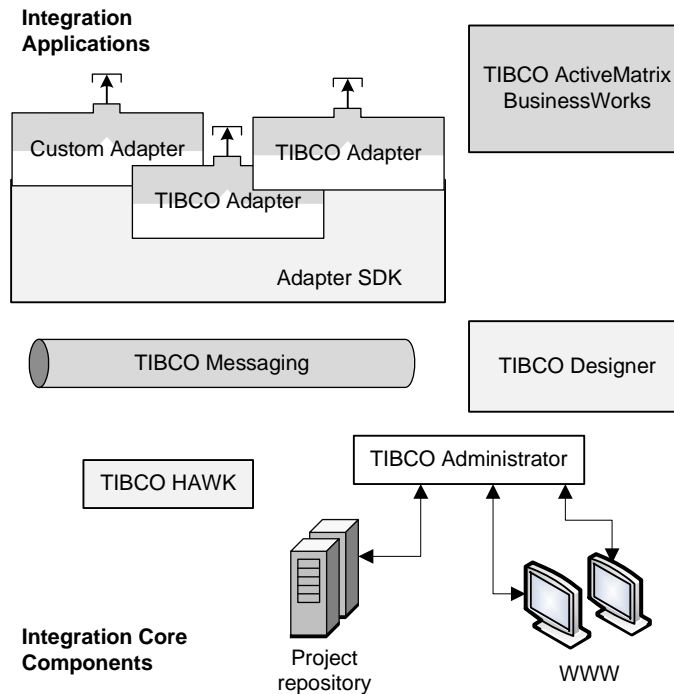
Using the TIBCO Adapter SDK to create a custom adapter has the following advantages:

- Compatibility with other TIBCO adapters and applications configured for the TIBCO environment.
- Common approaches to configure, transform, and route data imported and exported from the adapter.
- Set of reusable components that facilitate development.
- Easy-to-follow development paradigm with few constraints on the development process.
- Integration with TIBCO Hawk or TIBCO Administrator for monitoring.
- Standard exception handling, tracing, and logging mechanism.
- Standard tracking across the TIBCO ActiveEnterprise.

TIBCO Adapter SDK and TIBCO ActiveEnterprise

TIBCO ActiveEnterprise consists of core components, which are used by other TIBCO applications, and integration applications, which are built on top of the core components.

Figure 2 TIBCO ActiveEnterprise Core Components and Integration Applications



Adapter developers use TIBCO core components for adapter development and testing at runtime. TIBCO Adapter SDK integrates with each of them.

TIBCO Designer for Configuration

Adapters can be configured with TIBCO Designer, then implemented with the Adapter SDK. Completed SDK-based adapters can be loaded into TIBCO Designer so end users can perform instance configuration there.

A TIBCO Designer API enables adapter developers to create the appropriate custom interface for end users. Alternatively, end users can use the Generic Adapter Configuration included in TIBCO Designer.

TIBCO Designer provides an easy-to-use graphical user interface for creating integration projects. TIBCO Designer allows you to drag and drop components into a project and then configure each component. The project can then be saved and deployed for use during runtime.

Adapter developers use TIBCO Designer for adapter configuration, as discussed in the *TIBCO Adapter SDK Programmer's Guide*.

TIBCO Messaging Products

TIBCO Adapter SDK supports the messaging products TIBCO Rendezvous and TIBCO Enterprise Message Service.

- TIBCO Rendezvous allows distributed applications to exchange data across a network. TIBCO Rendezvous can provide reliable and certified services, and support certified queues.
- TIBCO Enterprise Message Service is a high-performance and robust implementation of JMS. The product supports persistent or reliable delivery modes. Standard JMS topics and queues are also available.

TIBCO Adapter SDK allows writing code that does not assume a specific messaging product. The *TIBCO Adapter SDK Programmer's Guide* discusses when the endpoint configuration can be changed to assume a different protocol (and not change SDK-based program) without causing significant performance issues or other problems.

TIBCO Administrator for User, Data, and Domain Management

TIBCO Administrator provides the following features:

- Local or server-based access of project repositories. TIBCO ActiveEnterprise applications store data in project repositories. Multiple formats are available.
- Load balancing and fault tolerance.
- Security, if the application joins the TIBCO administration domain.
- Registration of projects.
- Monitoring and management of runtime projects.

TIBCO Hawk for Monitoring

TIBCO software supports two monitoring and management products, namely TIBCO Hawk and TIBCO Administrator.

TIBCO Hawk is a tool for monitoring and managing distributed applications throughout the enterprise. With TIBCO Hawk, system administrators can monitor application parameters, behavior, and load activities for all nodes in a local or wide-area network, and take action when predefined conditions occur.

TIBCO Adapter SDK includes a number of prespecified microagent methods that can be used to monitor adapter applications.

TIBCO Hawk functionality continues to be added to the TIBCO Administrator product because its browser-based GUI is intuitive to users. For highly distributed applications that require custom monitoring functionality, TIBCO Hawk will continue to be the product of choice.

TIBCO Adapters and Other TIBCO Applications

TIBCO Adapter SDK uses TIBCO messaging as its underlying protocol. Adapters built with SDK are therefore inherently compatible with other TIBCO products.

This section discusses TIBCO products with additional facilities for integration with adapters.

Data Transformation Facilities

A custom adapter works together with an application-independent transformation engine to provide full adapter functionality.

- The adapter's primary functionality is to *retrieve data from* and *export data to* a source or target application such as an ERP (Enterprise Resource Planning) application.
- *Transformation and routing* functions can be performed by TIBCO ActiveMatrix BusinessWorks, which includes a mapper that supports data transformation and a process design component that enables routing behavior configuration.

A custom adapter created with the TIBCO Adapter SDK changes the *format* of the data so it can be accessed by other applications. Though the data model of a source application may be complicated, the adapter retains much of the source information and makes it easy to access the information.

However, a custom adapter does not change the *content* of the data, so two applications might still be incompatible. For example, the name and address of a customer might be stored in five fields in one ERP application and in six fields in another. In this case, content conversion must be performed by a separate tool, for example, the mapper included within TIBCO ActiveMatrix BusinessWorks.

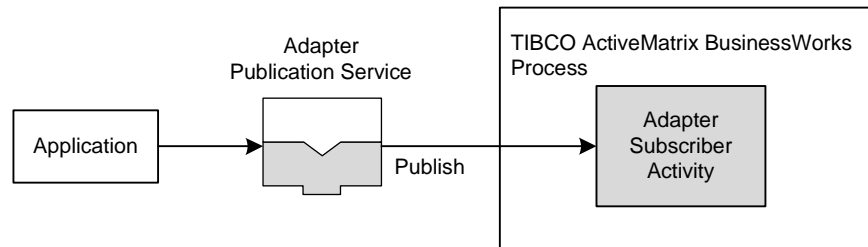
TIBCO ActiveMatrix BusinessWorks Mapper provides a graphic interface for creating and modifying data transformation rules. An XPath Formula Builder, which allows you to drag functions and data elements to create the transformation rules, is available in the GUI.

The adapter pushes data out of a third-party application (for example, SAP R/3). TIBCO ActiveMatrix BusinessWorks can then transform the data to be accessible by other adapters.

TIBCO ActiveMatrix BusinessWorks Adapter Palette

TIBCO ActiveMatrix BusinessWorks is configured using TIBCO Designer. When configuring a TIBCO ActiveMatrix BusinessWorks integration project, TIBCO Designer includes an ActiveEnterprise Adapter palette that contains activities capable of interacting with adapter services. For example, an Adapter Subscriber activity can be configured to receive data from an adapter Publication Service.

Figure 3 An Adapter Subscriber Activity Receiving a Message from an Adapter Publisher



See the TIBCO ActiveMatrix BusinessWorks documentation set for more information.

Adapter Interactions

At the most basic level, a TIBCO Adapter makes data available from a source application or allows information resulting from transactions in a TIBCO application to be entered into a target application. This is often discussed in terms of a services architecture. Services are abstractions that describe how adapters work together with other applications.

An adapter generally supports publish/subscribe and request/response interactions.

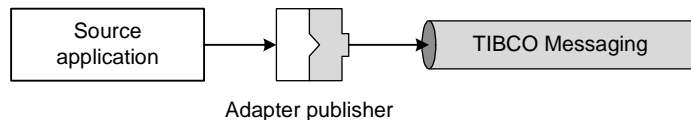
Publish/Subscribe Interactions

Publish/subscribe interactions are driven by events such as the arrival of data or a timer signaling that a specified interval has expired. The following services are available for publish/subscribe interactions.

Publication Service

An adapter *publication service* makes information about source application business events available to other applications enabled for the TIBCO environment. For example, an adapter can publish an event each time an order changes its status in an Oracle database.

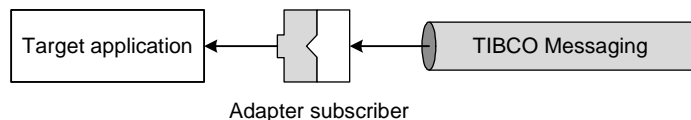
Figure 4 Publication Service



Subscription Service

An adapter *subscription service* gets information about business events from the TIBCO environment and writes the information into a target application. For example, an adapter can subscribe to events that indicate the creation of a new customer and then enter the customer information into a Siebel customer care target application.

Figure 5 Subscription Service



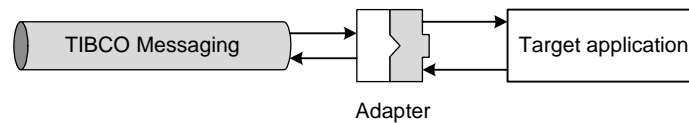
Request/Response Interactions

Demand for data drives request/response (client/server) interactions. The complete interaction consists of two point-to-point messages—a request and a response.

- A client requests data from a server.
- The server computes an individual response and returns it to the client.

Communication flows in both directions, as illustrated in [Figure 6](#).

Figure 6 Request-Response Interaction



Demand-driven computing suits distributed applications that require point-to-point messages. In request/response interactions, data providers coordinate closely with data consumers. A provider does not send data until a consumer requests it.

The server sends replies to the client that requested the data. The client listens until it receives the reply, and then stops listening (unless it expects further installments of information).

The following services are available for request/response interactions.

Request-Response Service

An adapter *request-response service* acts as a server, which retrieves information on demand and accepts instructions to perform transactions in the target application.

Request-Response Invocation Service

An adapter *request-response invocation service* acts as a client, which sends a request for information or actions from the target application to another application.

The application processes the request and sends a response back to the adapter, which then returns the response to the target application.

Chapter 2

Designing and Implementing Custom Adapters

This chapter gives a brief introduction of the adapter development process. See the *TIBCO Adapter SDK Programmer's Guide* for more information.

Topics

- [Overview, page 14](#)
- [Requirements Definition, page 15](#)
- [Adapter Configuration, page 16](#)
- [Adapter SDK Programming, page 20](#)
- [Tracing and Tracking, page 23](#)
- [Internationalization, page 25](#)

Overview

Adapter development consists of several stages:

- **Requirements Definition**—Requirements definitions must be comprehensive, including how the adapter communicates with other applications and what throughput and fault tolerance requirements are expected.
- **Adapter Configuration**—Configuration is performed with the TIBCO Designer. Developers create a Generic Adapter Configuration instance and add the required resources for that instance.

The completed adapter configuration is saved to the project repository. Various formats are available. For more information, see the *TIBCO Designer Adapter Resource Management Guide*. Examples are given in the *TIBCO Adapter SDK Programmer's Guide*.

- **Adapter SDK Programming**—An SDK program uses the information in the adapter configuration and performs specified actions.

The *TIBCO Adapter SDK Programmer's Guide* gives instructions on how to use the available classes. In addition, API Reference documentation for both C++ and Java, and a set of programming examples are present in the TIBCO Adapter SDK package.

As part of the cycle, testing must be included at each stage.

Requirements Definition

The first step in adapter development is requirements gathering and clarification. Here are some points an adapter developer should consider:

- What are the available methods for interfacing with the application? For example, direct API access, protocol access (JDBC), and so on.
- Does the application use basic ASCII characters or special characters? This choice especially affects the runtime application setup.
- What are the metadata that describe the data the adapter uses? These data are specified through the resources in the TIBCO Designer AESchemas folder.
- What are the services the adapter provides? (publication, subscription, remote invocation, and so on).
- Each service encapsulates an endpoint and the corresponding session (transport), that is, TIBCO Rendezvous or TIBCO Enterprise for JMS. What is the message format appropriate for the application? What is the quality of service for that message format?
- What is the protocol for internal communication, TIBCO Rendezvous or TIBCO Enterprise Message Service?
- What is the message format for communicating with the Administration Server, HTTP or TIBCO Rendezvous?
- What is the format of data sent out on the network, aeXml or aeRvMsg?
- What is the appropriate mechanism for exception handling, monitoring, and tracing?
- Will tracking across enterprise applications be included?

Adapter Configuration

Adapters are configured in TIBCO Designer using the Generic Adapter Configuration resource and its dependent resources.

Adapter configuration consists of the configuration of the adapter itself and the creation and definitions of metadata schemas.

Basic Adapter Configuration

Basic adapter configuration, which includes startup state, session parameters, adapter service definition, tracing information, and so on, is the configuration of the adapter itself.

Services

Adapters can provide Publication Service, Subscription Service, Request-Response Service, and Request-Response Invocation Service.

Sessions and endpoints are part of service configuration.

Pointers to Metadata (loadURL)

A loadURL resource can be configured to point to the location of the metadata.

Tracing

Developers can choose to use default tracing or advanced tracing.

Hawk Monitoring Related Information

Developers can enable the standard or class Microagents, or define custom Microagents.

Custom Configuration

Any configuration that the adapter developers need in addition to the standard configuration is custom configuration. For example, database connection account information.

Transformation Plug-in

A low-level plug-in that allows the transformation of message content before or after it is sent out on the network.

Schema Configuration

The TIBCO Designer AESchemas palette can be used for the schema (metadata) configuration. With the schema palette, developers define the data that the adapter sends or receives and also the parameters and exceptions of the operations a client/server type of adapter uses. Developers can also generate schema in AEXML format and import them into the repository or into TIBCO Designer. For more information, see *TIBCO Designer Palette Reference*.

Examples of how to use TIBCO Designer to perform adapter configuration are included in the *TIBCO Adapter SDK Programmers Guide*. It is recommended that first time Adapter SDK developers work through the examples to familiarize themselves with Designer.

You can also access online help for each resource from the TIBCO Designer GUI.

Metadata (Schemas)

Metadata describes application data. Metadata isolates the data description from the application data and allows data to be exchanged independent of languages and platforms.



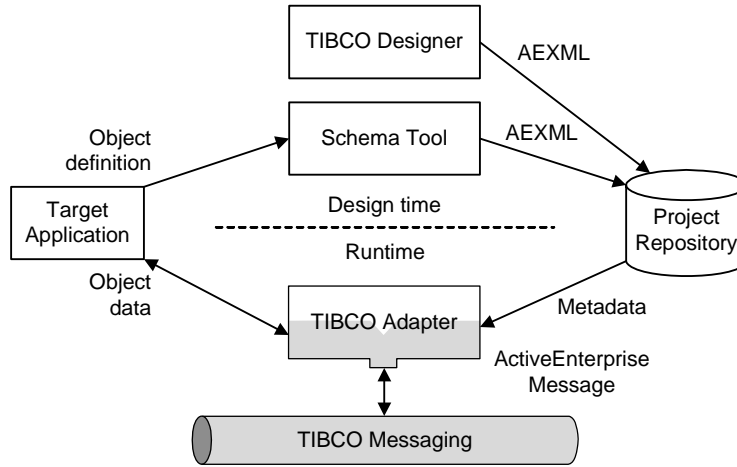
The terms metadata and schema data are used interchangeably.

TIBCO Designer defines metadata classes, attributes, and other items at design time. There are two types of metadata:

- Metadata that define incoming or outgoing data.
These metadata are defined with class, union, or association resources and have scalar-type attributes or complex-type attributes (which can be another class, union, or association).
- Metadata that define operation methods, parameters, and exceptions.
These metadata have a name, a method or methods with a return class (if any), and parameters (their name, class and the direction: in, out, or both), as well as any exceptions the operation raises. These metadata are used by SDK Operations classes.

By default, metadata is loaded upon startup. The SDK stores the metadata information using a class registry and associated classes.

Figure 7 Metadata Creation and Usage



Metadata is used in a similar way to a hardcopy specification document that is shared among developers. Instead of delivering a hardcopy document for each developer and updating and distributing that document periodically, applications access metadata through the adapter instance description stored in a repository. Because metadata is managed by TIBCO Administrator, changes in the data description are immediately picked up by all applications or application components.

Global Variables

By default, a number of global variables are defined for TIBCO Designer projects. These global variables set the values for parameters such as session, subject, and so on.

It is recommended that these variables be kept and used because they provide a flexible way of changing the adapter configuration at runtime without having to load, reconfigure, and save the adapter configuration instance each time.

However, depending on the requirements, it may not be desirable to change an adapter configuration at runtime. In this case, fixed values can be set.

Saving Adapter Configuration

The adapter configuration can be saved in three formats:

- Multi-file project—The preferred format during development. Allows multiple developers to collaborate with a version control system.
- Single-file repository project—A single .dat file.
- Server-based repository project—A server-based .dat file, which can be managed by a TIBCO Administrator administration server.

See *TIBCO Administrator Server Configuration Guide* for more information.

Adapter SDK Programming

TIBCO Adapter SDK programming is discussed in detail in *TIBCO Adapter SDK Programmer's Guide*. Reference documentation for C++ and Java API is also included in the Adapter SDK documentation set.

Programming Features

This section gives an overview of SDK features that facilitate the development of custom adapter programs.

Abstraction

The SDK abstracts transport details and many other things a developer usually needs to manage when writing applications that communicate with each other across networks.

Default Behaviors

Default behaviors for multiple interaction scenarios are provided, such as default message acknowledgement behavior. These behaviors can be customized when needed.

Toolkit Items

Tracing, tracking, TIBCO Hawk methods, NT Service wrapper, and other classes or small packaged modules are available.

Separations

Configuration is separated from programming, application data is separated from metadata, and so on. These kinds of separation allow easy customizations. For example, the configuration data can be changed without changing the program. The metadata can also be changed to affect the data the adapter sends or receives.

Program Elements

This section briefly describes the main elements of an adapter program.

- The *application manager* handles the initialization and shutdown of an adapter and provides the glue logic that holds all components together. As part of the initialization, it loads the configuration data (sessions, endpoints, and so on)

and metadata into memory. It also sets up tracing and TIBCO Hawk management. Applications can customize initialization and shutdown.

- When configuring a service in TIBCO Designer, an endpoint and a session are created.



The SDK encapsulates endpoints and sessions but not services.

- *Endpoints* send or receive the data. They are the publishers, subscribers, clients, and servers in the custom adapter. They are configured to use a specific message format, which packages the data going over the network.
- Each endpoint is associated with a *session*. Sessions (or the corresponding services) can be configured to use TIBCO Rendezvous or TIBCO Enterprise for JMS. Different quality of service options are available. A session is used to communicate with the source or target application.
- The program is executed based on an event model using event sources (which are the endpoints that receive or send data), events, and event listeners. Event listeners are created programmatically to correspond to endpoints.
- Data in the SDK are encapsulated by two classes:
 - An `MTree` class represents hierarchical data in the wire format used by an endpoint. On the network, data are always represented as `MTrees`.
 - An `MInstance` class represents data constrained by metadata. In addition to `MInstance`, the SDK supports other classes such as `MSequence` or `MAssociation`.

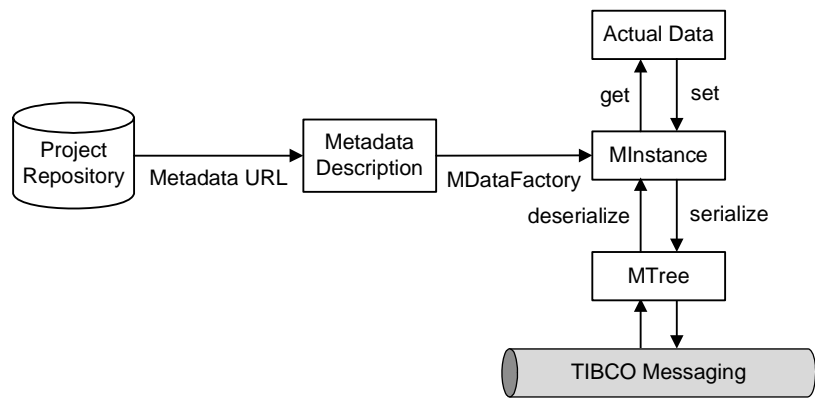
Metadata can also be specified in TIBCO Designer. Upon startup, the application manager creates appropriate metadata description classes that can then be accessed when creating instances.

Data at Runtime

At runtime the adapter creates instances (`MInstance`) as defined by the metadata and populates the `MInstance` attributes with object data from the target application for publishing or responding to request-response invocation.

On the receiving side, a subscriber adapter converts the incoming message back to an instance through de-serialization, which also validates the message against expected metadata definitions.

Figure 8 Data at Runtime



Tracing and Tracking

TIBCO Adapter SDK offers various ways to send information from a running custom adapter.

Tracing

Upon startup of an adapter application, the Adapter SDK creates a trace class that allows custom adapters to implement the following tracing functions:

- Single trace message to multiple target sinks: Files, network (TIBCO Rendezvous messages), standard IO, and TIBCO Hawk. Users need only change the configuration of the adapter instance to change the destination of the trace messages.
- Multiple roles per sink (info, debug, error, and warn). Users can attach any role to any sink in the adapter configuration. Adapter developers can also define custom roles.

TIBCO Adapter SDK implements the ActiveEnterprise trace format used by ActiveEnterprise applications. Developers create a separate XML file using a predefined format in which trace messages are stored. Each time the custom adapter needs to send out a trace message, it uses a message symbol. The SDK accesses the message corresponding to the message symbol in the XML file and sends it to the appropriate destination.



Storing error messages in a properties file or separate error message class allows you to easily localize the adapter to a specific language.

The Adapter SDK includes the `genAeErrors` utility, which parses the XML file containing message descriptions and generates appropriate code for inclusion in the custom adapter.

See *TIBCO Adapter SDK Programmer's Guide* for more information on configuring and using tracing facilities. See also the description of `MMessageBundle` and `MTrace` in the *TIBCO Adapter SDK API References*.

Tracking

In contrast to tracing, tracking information is included with the data. This enables the tracking of the progression of data as it moves from one ActiveEnterprise application to another.



Use tracking sparingly; it can result in large messages.

For example:

1. Data originates at a source and is picked up by an adapter publisher.
2. The publisher adds tracking information before it sends the data to TIBCO ActiveMatrix BusinessWorks.
3. TIBCO ActiveMatrix BusinessWorks performs data transformation, adds tracking information, and then sends the data to the messaging system.
4. An adapter subscriber picks up the message and enters the data into a target information.

If a problem occurs, you can look into the tracking information to determine when the process stopped working as expected.

Tracking information uses an `MTrackingInfo` object. The source application places appropriate information into the `MTrackingInfo` objects using the `MTrackingInfo::addApplicationInfo()` method. SDK adds this application information to the `MTree`, `MData`, or `MOperation`. As the data is sent through the enterprise, each application adds information about its operations to messages that pass through.

You can use the tracking information within an ActiveEnterprise wire format message to trace where a message originated and where it has been.

Internationalization

Both C++ APIs and Java APIs support Unicode for application data. This allows custom adapters to work with Unicode strings programmatically and to send and receive data between applications that use a variety of supported encodings. The data can be aggregated and serialized, and then sent over the network.

The Java Adapter SDK takes advantage of native Java Unicode support.

The C++ Adapter SDK includes classes to encapsulate Unicode data. Their constructors allow you to specify the encoding for the source data. A complete list of supported encodings can be found in the *TIBCO Adapter SDK Programmer's Guide*.

SDK-Internal C++ Unicode Type Conversion

Internally, the C++ SDK first decides to use one of two native implementations: Latin-1 for single-byte characters or UTF-16 for double-byte characters.

Whether the SDK attempts conversion, and what conversion the SDK attempts depends on the encoding argument presented to the constructor for `MChar` or `MStringData`.

If the encoding presented to the constructor is ASCII, Latin-1, or UTF-16, no conversion is needed. For all other cases, the SDK attempts a best-case conversion. If conversion is required (for example, UTF-16 to Latin-1), a replacement character is used for unmappable characters.

For Unicode conversion, the environment variable `TIB_ICU_DATA` must point to the directory that contains the file named `tibicudata.dat`, which provides a lookup table for conversion. If the SDK cannot find this file, it will throw an exception when you attempt to convert certain types of string encodings.

Specifying the Wire Format Encoding

The wire format encoding for messages affects communications for adapter applications. TIBCO Administrator supports either Latin-1 or UTF-8 as its wire format encoding when the adapter application is using a server-based repository.

If the project uses only ASCII or Latin-1 data, you can set the encoding to be Latin-1. This encoding makes the custom adapter run faster. Otherwise, use UTF-8.

Index

A

- adapter configuration
 - saving [19](#)
- Adapter SDK
 - and TIBCO ActiveEnterprise [5](#)
 - benefits [4](#)
- Adapter SDK programming [20](#)
- adapters
 - and message transformation systems [8](#)
 - configuration [16](#)
 - design [13](#)
 - implementation [13](#)
 - overview [4](#)
- application manager [20](#)

B

- basic adapter configuration [16](#)

C

- changes from the previous release of TIBCO Adapter
 - SDK Concepts [viii](#)
- class library [4](#)
- CONFIG_HOME [x](#)
- configuration [5, 16](#)
 - saving [19](#)
 - schema [17](#)
- custom configuration [16](#)
- customer support [xiii](#)

D

- data at runtime [21](#)
- data management [6](#)
- data producers
 - in request/reply model [11](#)
- designing custom adapters [13](#)
- domain management [6](#)

E

- encoding
 - wire format [25](#)
- endpoints [21](#)
- ENV_NAME [x](#)

G

- global variables [18](#)

H

- Hawk [6](#)

I

- implementing custom adapters [13](#)
- interactions
 - publish/subscribe [10](#)
 - request/response [11](#)
- internationalization [25](#)

introduction [1](#)

M

message transformation [8](#)

message transformation systems [8](#)

messaging products [6, 6](#)

metadata [17](#)

and repository [18](#)

monitoring [6](#)

O

overview [2, 14](#)

P

program elements [20](#)

programming features [20](#)

publication service [10](#)

publish/subscribe interactions [10](#)

R

request/response interactions [11](#)

response [11](#)

runtime data [21](#)

S

saving adapter configurations [19](#)

schema [17](#)

schema configuration [17](#)

SDK_HOME [x](#)

subscribe [10](#)

subscription service [10](#)

support, contacting [xiii](#)

supported platforms [2](#)

T

technical support [xiii](#)

TIBCO ActiveMatrix BusinessWorks

adapter palette [9](#)

TIBCO adapters

and other TIBCO applications [8](#)

TIBCO Administrator [6, 18](#)

TIBCO Designer

configuration [5](#)

default configuration values [18](#)

TIBCO Hawk [6](#)

TIBCO Messaging Products [6, 6](#)

TIBCO_HOME [x](#)

tracing [23](#)

and tracking [23](#)

tracking [24](#)

transformation of messages [8](#)

U

Unicode [25, 25](#)

type conversion [25](#)

V

variables [18](#)

W

wire format encoding [25](#)