TIBCO Adapter™ Concepts

June 2011

The Power of Now®

Important Information

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.

Contents

Figures

Tables

Chapter 1 Introduction

This chapter introduces adapters by explaining adapter key features, services, and functionality.

Topics

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.

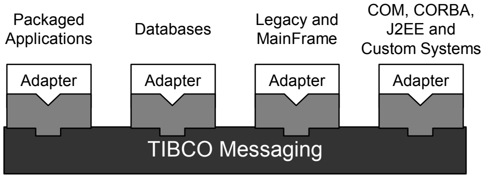
An adapter provides a bridge between an application and your TIBCO integration environment. Using a no-coding approach to integration, TIBCO Adapters enable packaged applications, databases, and other technologies to become active participants in the enterprise information flow, regardless of their data 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.

Figure 1 Adapters provide a bridge for data



Prerequisite Knowledge

Many vendor applications have been integrated into the TIBCO environment using adapters. The most commonly used adapters are described in Appendix A, Adapter Types, on page 67. Before integrating an adapter at your site, you should have working knowledge of the vendor application.

Each TIBCO Adapter documentation set includes a User's Guide and Release Notes. Many adapters also have an Examples Guide. The documentation set describes how to configure an adapter to interoperate with the vendor application, and with TIBCO infrastructure tools.

Adapter Key Terms

At the most basic level, an adapter exposes transactions carried out by the vendor application to the rest of the infrastructure and allows information resulting from transactions in other applications to be entered into the vendor application.

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

- A *run-time adapter* is the adapter component that is used in a production system to handle communication between a vendor application and TIBCO infrastructure.
- A *design-time adapter* is an adapter component that is used at configuration time to introspect the vendor application data schemas and configure the adapter. Not all adapters use design-time adapters. Many adapters have the same functionality built into their palettes.
- A *palette* is an adapter component that contains the screens used to gather input at design-time when configuring an adapter with a service.
- 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 Admininstrator™ product.

Adapter Lifecycle

The following is an overview of the adapter lifecycle. More information is available in Chapter 4, Adapter Lifecycle.

1. Install the vendor application to which the adapter connects before installing the adapter. For many adapters, the adapter and vendor application need not be installed on the same machine.
2. Adapters depend on other software from TIBCO. Before installing an adapter, the TIBCO Runtime Agent™ software must be installed on each computer on which the adapter runs. See Overview on page 22 for details.
3. Create an adapter instance and save it in a project using TIBCO Designer™. A project contains configuration information required for a run-time adapter to interact with the vendor application and other applications.
4. Deploy the adapter. An adapter instance is deployed using TIBCO Administrator.
    a. Using TIBCO Designer, create an Enterprise Archive (EAR) file, which contains information about the adapter instances and processes you wish to deploy.
    b. Using TIBCO Administrator, upload the EAR, then deploy the adapter on the machine(s) of your choice. You can set runtime options before deployment.
    c. Using TIBCO Administrator, start and stop the adapter.
    d. Monitor the adapter using the built-in monitoring tools provided by TIBCO Administrator.

Adapter Services Examples

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
- Subscription Service
- Request-Response Service
- Request-Response Invocation Service

The diagrams in this section use shaded icons to indicate TIBCO components and non shaded icons to indicate other components.

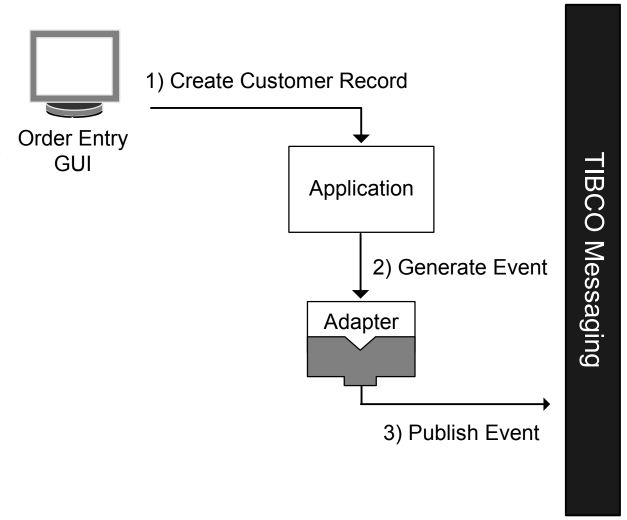
Publication Service

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.

As shown in the next diagram, when an application receives a request to create a customer record, the application notifies the adapter about the request and the adapter publishes the event.

Figure 2   Adapter publication service



The following steps correspond to steps listed in the diagram:

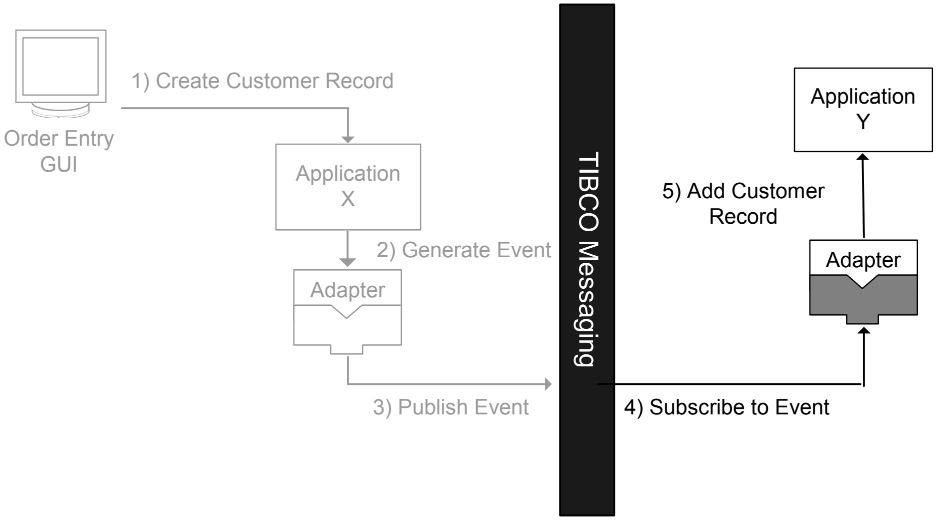
1. Using a graphical user interface, a new customer record is created in the application.
2. The application sends the new customer record to the adapter.
3. The adapter publishes the event using TIBCO messaging.

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.

Figure 3 Adapter subscription service



The following steps correspond to the new steps listed in the diagram:

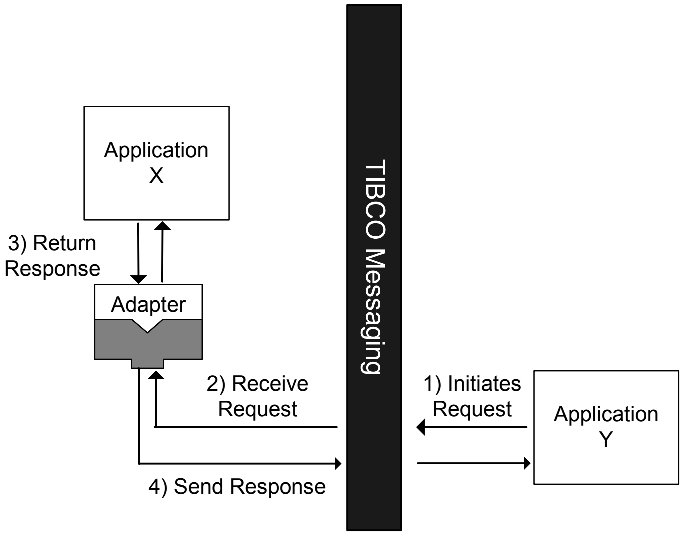
4. The adapter is configured to listen to customer record creation events as they are published.
5. Upon reception of a customer creation event, the adapter updates the application to which it is connected.

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.

In the next 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.

Figure 4 Adapters retrieve data



The following steps correspond to steps listed in the diagram:

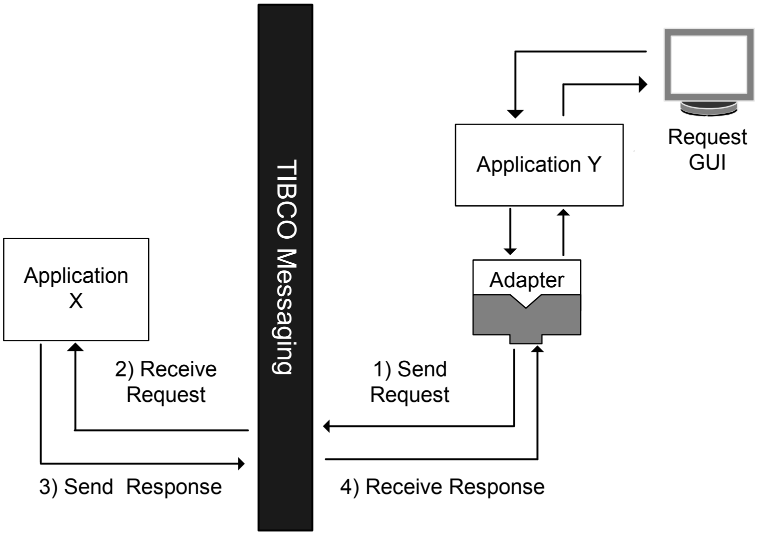
1. An application sends a message request to the TIBCO environment.
2. The adapter, acting as a request-response server receives the request and forwards the request to the application.
3. The application parses the request and returns a response message to the adapter.
4. The adapter sends the response message back to the TIBCO infrastructure where the response is available to the requesting application.

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.

The next diagram shows 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.

Figure 5 Adapters act as proxies



Adapter Services Summary

The next table summarizes the services introduced in this section.

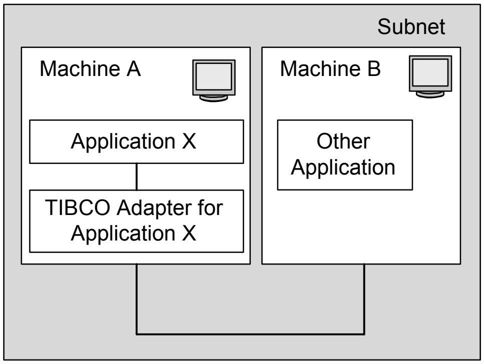
Table 1 Adapter Services Summary

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:

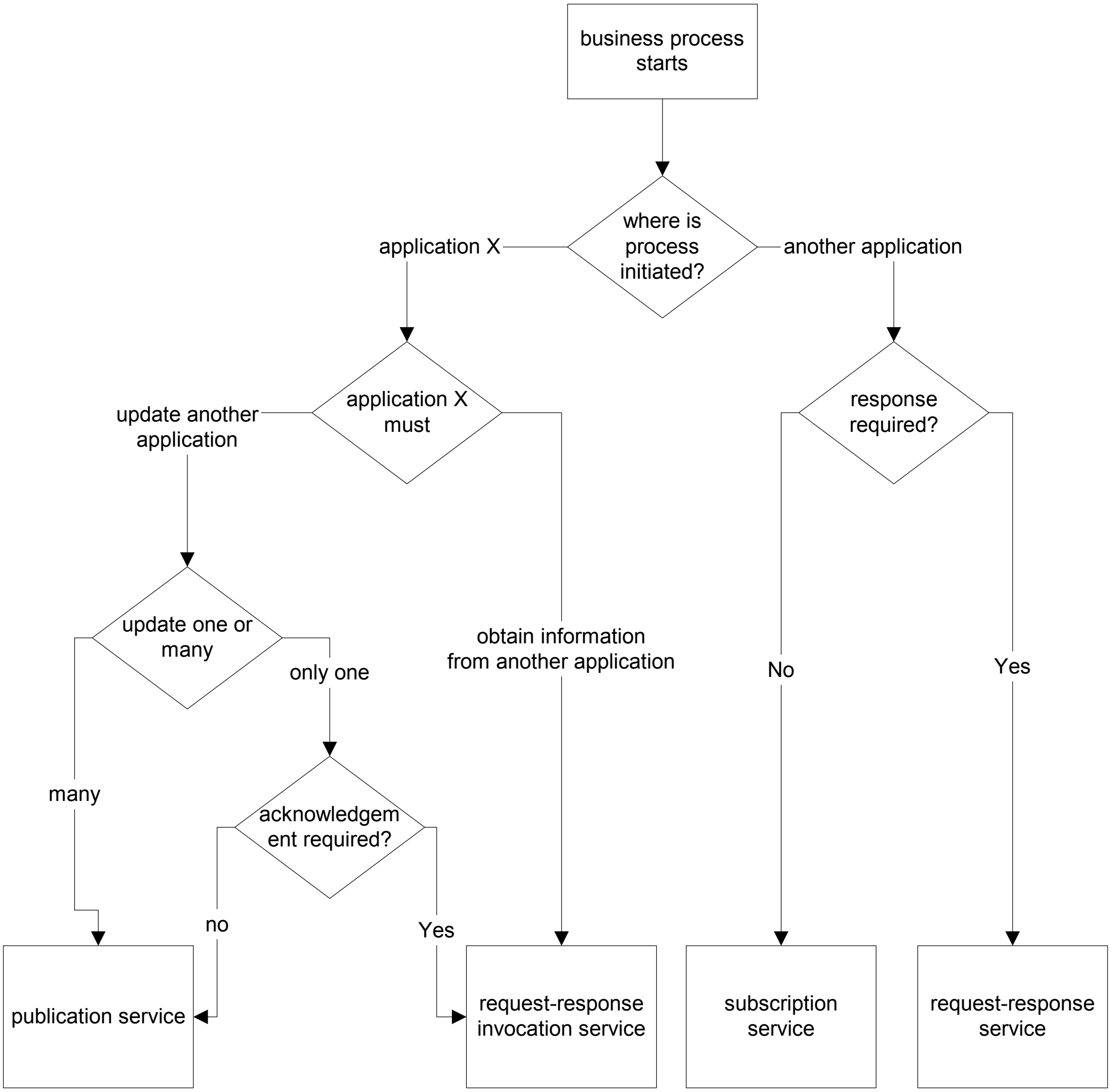


In this scenario, data must be exchanged between the application X and another application. The other application could be a customer management system, such as PeopleSoft, or another TIBCO application, such as TIBCO ActiveMatrix BusinessWorks.

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 Key Features

The following features are available to most adapters.

- Easy-to-use GUI. All adapters use TIBCO Designer for configuring and maintaining each adapter. See TIBCO Designer on page 24 for more information.
- Built-in monitoring and control using TIBCO Hawk. All adapters provide vital business statistics (such as number of messages processed) to the Hawk console and also include Hawk microagents that provide methods to query remotely, set run-time parameters and stop a run-time adapter. See TIBCO Hawk on page 33 for more information.
- Support for multiple message transports. TIBCO Rendezvous or TIBCO Enterprise Message Service can be used to transport messages to and from an adapter. See Message Transports on page 48 for more information.
- Multiple Platform Support. Adapters are usually supported on multiple platforms such as Microsoft Windows, Solaris, and HP-UX. See the readme file for each adapter for details.
- Connection Management. An adapter can detect and replace lost connections to its vendor application. Connection loss is detected upon receipt of a message intended for a particular application. Adapters support both synchronous and asynchronous attempts to replace a lost connection. See Connection Management on page 56 for more information.
- Multi threading. Most adapters maintain a dynamic pool of threads for performing tasks, which allows the adapter to respond to and process multiple events simultaneously. The adapter can track and communicate event status without affecting other processing. See your adapter user's guide for more information.
- Error Handling. All adapters use the same error handling mechanism to report run-time adapter errors. Resolutions for run-time errors are documented in the adapter's user's guide.
- Multi-lingual Support. Most adapters support multi-byte character sets. See Internationalization on page 59 for more information.

Chapter 2 Adapter Components

This chapter introduces adapter components.

Topics

Overview

Each adapter has two main components, an adapter palette and a run-time adapter. In addition, some adapters include a design-time adapter. The adapter palette and design-time adapter are used during configuration, and the run-time adapter is used at production time.

Adapter Palette

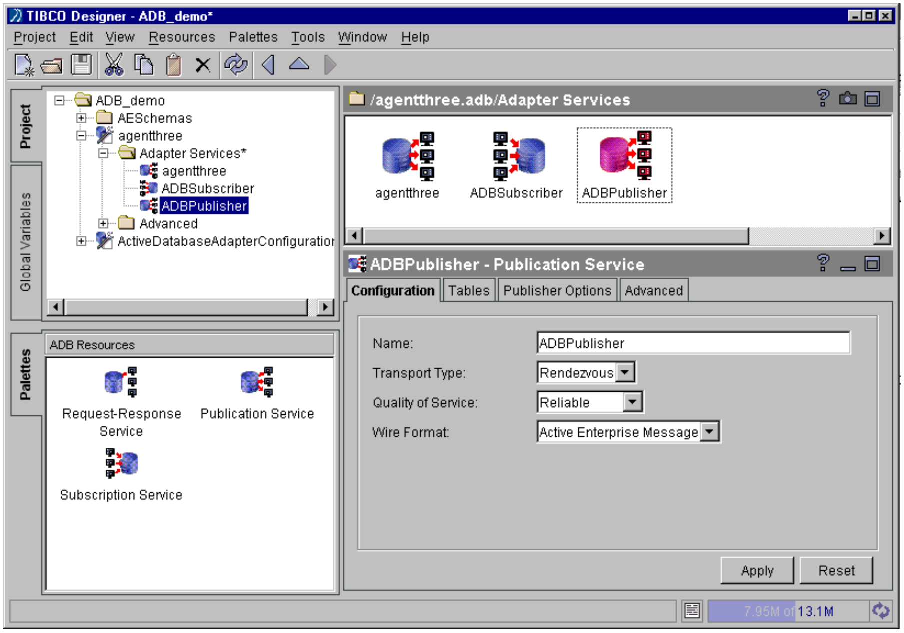
Each adapter includes a palette that is used for configuration. The palette is automatically loaded into TIBCO Designer during adapter installation and available the next time Designer is started. The palette enables you to configure adapter specific options, such as its connection to the vendor application, logging options, and adapter services.

During the design phase, the palette connects to the vendor application and fetches information about connection options and data schemas. You can then graphically select the appropriate items.

For example, during configuration of a TIBCO ActiveMatrix Adapter for Database adapter instance, the palette fetches all pertinent tables in the database. You then choose the tables that the particular service is to send or receive.

The following diagram shows the TIBCO ActiveMatrix Adapter for Database palette and the configuration options for a publication service.

Figure 6 Example adapter palette



Name: ADBPublisher
Transport Type: Rendezvous
Quality of Service: Reliable
Wire Format: Active Enterprise Message

Run-time Adapter

Once the adapter has been configured using TIBCO Designer, it can be deployed. A deployed adapter instance is referred to as a *run-time adapter*. A run-time adapter operates in a production environment, handling communication between a vendor application and other applications that are configured for the TIBCO environment.

See Chapter 4, Adapter Lifecycle, on page 35 for more information.

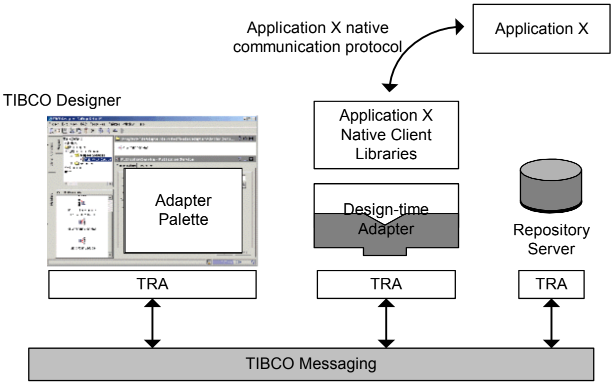
Design-time Adapter

Some adapters use a design-time adapter (DTA) to access a vendor application and return design-time configuration information. The palette is a client of the DTA process. The DTA connects to the vendor application, fetches data schemas and sends them to the palette.

Component Details

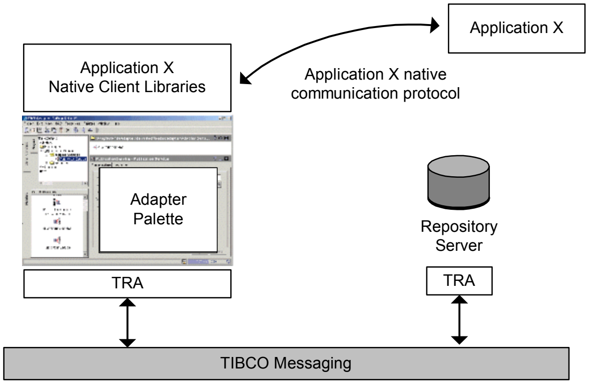
The next diagram shows components used at design-time for adapters that provide a design-time adapter. The DTA uses the native client libraries of the application to access schema from the application. The palette (in TIBCO Designer) communicates with the DTA.

Figure 7 Design-time adapter connection to application



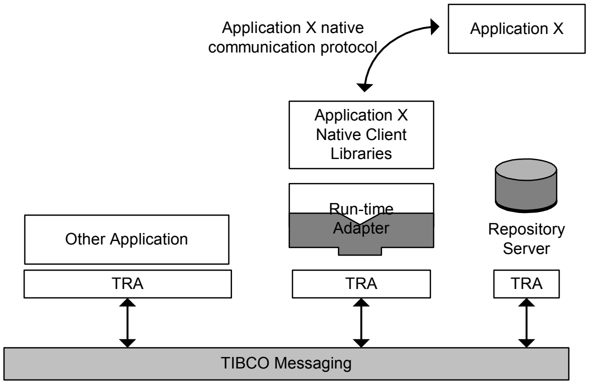
Not all adapters require a design-time adapter. Where possible, the adapter palette communicates directly with the native client libraries of the application as shown in the next diagram.

Figure 8 Adapter palette connection to application


Application X
Application X Native Client Libraries
Application X native communication protocol
Adapter Palette
Repository Server
TRA
TRA
TIBCO Messaging

As shown in the following diagram, the adapter at run-time uses the native client libraries to communicate with the vendor application.

Figure 9 Run-time adapter connection to application



Chapter 3 Adapter Infrastructure Tools

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

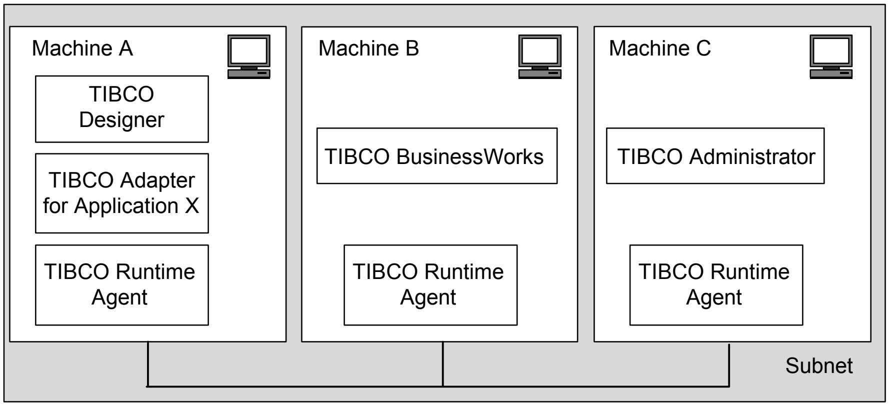
Topics

Overview

All adapters are configured, deployed, and managed using the same infrastructure tools. This section introduces the tools by providing a quick overview of how they interact with the adapter. See Documentation Roadmap on page 64 for references to the documentation sets for each tool.

The next diagram shows an adapter and the infrastructure tools it uses. Infrastructure products can be installed on the same machine or on different machines as shown in the next diagram.

Figure 10 Adapter and TIBCO infrastructure tools



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

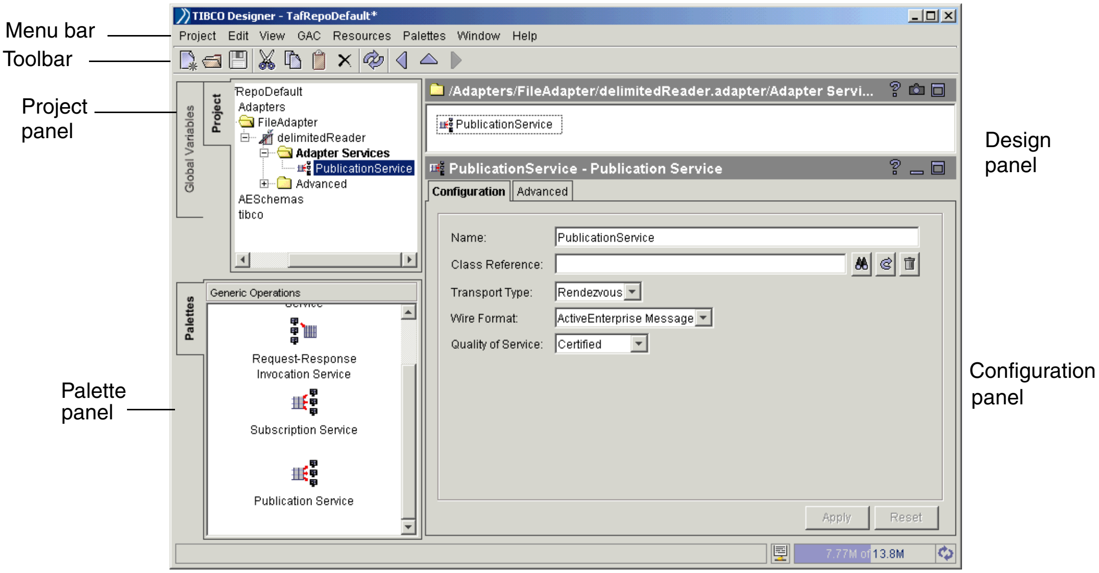
TIBCO Designer provides the design-time environment for configuring an adapter. Using Designer, you create a project, add adapter services to it with a simple drag-and-drop interface, and specify the configuration information for each adapter service.

Before using TIBCO Designer, make sure you read:

- The *TIBCO Designer User's Guide*, which explains how to work with projects, and customize the GUI.
- The TIBCO Adapter User's Guide for the specific adapter you are working with. The guide explains how the adapter interacts with the vendor application, and how to install and configure the adapter.

The guides are available via the TIBCO Designer **Help > Designer Help** from the menu bar. The next diagram shows the TIBCO Designer interface.

Figure 11 TIBCO Designer main window



Name: PublicationService

Class Reference:

Transport Type: Rendezvous

Wire Format: ActiveEnterprise Message

Quality of Service: Certified

Project Panel

Each TIBCO Designer window contains one and only one project, which is represented as the root folder in the panel.

Projects are the key organizational principle for the configuration information you specify. A *project* is a collection of all configured resources. *Resources* are the components of a project. For example, an adapter publication service is a resource. Resources can be complex and contain other resources, much like a folder can contain other folders on your computer file system. Together, these resources make up your integration project. The top-level (root) folder in the project tree panel represents a project. The top-level folder is initially named `Untitled` and is renamed to the name of the project when you save the project for the first time.

Most resources have context-sensitive help available for the configuration of that resource. Right-click on the resource and choose **What Is This?** from the popup menu for more information on configuring the resource.

An adapter project contains the following folders:

- AESchemas Folder

  The *AESchemas* folder is the default location for all TIBCO ActiveEnterprise schema files. Each schema file contains a collection of classes, scalars, associations, unions, and sequences.

- Adapter Services Folder

  The *Adapter Services* folder contains services available to the adapter. Most adapters include publication, subscription, request-response, and request-response invocation services.

- Advanced Folder

  The *Advanced* folder contains resources created by TIBCO Designer while the adapter is configured. For example, each time you add a service to an adapter, a session and endpoint are created and stored in the Advanced folder. Other resources such as advanced logging resources are accessed directly from the folder. Adapter developers typically do not access resources in this folder. Most of the adapter configuration is done by changing resources that are available from the Adapter Services folder.

Palette Panel

Palettes organize resources and allow you to add them into your project. Palettes are available from the palette panel in TIBCO Designer. Resources are visible components in a palette. You select resources in the palette panel and drag-and-drop them into the design panel to add them to your project.

Each adapter you install adds one or more palettes during installation. Which palette is displayed depends on the resource selected in the project tree and on your preferences. In the default view, the current selection in the project tree determines which palettes are displayed in the palette panel.

Design Panel

The design panel displays the current resource selected in the project tree panel. For resources that contain other resources, the contents of the selected resource are shown in the design panel. For example, if you select a folder, its contents are displayed.

Configuration Panel

The configuration panel allows you to specify configuration options for the selected resource. The type and the purpose of the resource determine the contents of the configuration panel. There are usually one or more tabs in the configuration panel that allow you to access the various configuration options. The tabs provide an organization to the options for the resource.

You can click the question mark icon (`?`) in the top right corner of the configuration panel for online help on the current selection.

For each tab, you must click **Apply** after you have specified configuration information before you can select another tab. If you decide you do not want to add the configuration information, click **Reset** before you apply any changes to return to the previous values for each field in the tab.

Projects

A project is a named collection of data, usually schema data, and configuration data that is persistently stored. Each project is opened and saved in multi-file format, which allows the project to be used with a version control system. It allows different developers to collaborate on a project and merge changes as needed.

When a project is ready to be deployed, it can be created or exported in the following formats:

- Enterprise Archive File
- Local Repository
- Server Repository
- ZIP Archive

Enterprise Archive File

An Enterprise Archive file contains information about the adapter instances and processes you wish to deploy. The format is used by TIBCO Administrator. The EAR file is imported into Administrator where you can deploy, start, and manage the adapter instance on the machines of your choice.

Local Repository

A project exported to a local repository is saved in `.dat` format. Projects saved in `.dat` format should only be used for development and testing. The format can be used where data is not shared by more than one adapter. It is possible to have a few local adapters accessing a local project in read-only mode. It is, however, not possible to have more than one local adapter accessing a local project in read and write mode.

Data are loaded at startup for local projects, so a local project has higher memory requirements.

Server Repository

A project exported to a server repository is managed by a TIBCO Administration Server running in a separate process, typically elsewhere on the network. One or more adapters can communicate with a project managed by an Administration Server. Each can support multiple projects.

An Administration Server is identified by a name that must be unique among all administration servers on a network. The server-based mode of operation is scalable and generally recommended for production situations. Server repositories allow multiple simultaneous write operations with locking, automatic updates of clients, and notification.

Data are loaded on demand for server-based projects.

ZIP Archive

A project exported to a ZIP archive is written to the location you specify as a read-only ZIP file. A project exported as a ZIP archive can be imported into TIBCO Designer.

Version Control

TIBCO Designer allows multiple developers to work on the same project and to use file sharing/locking or a version control system so that the same resource is not changed by two developers at the same time. Different users can then add resources to the project and lock the parts of the project they are working on.

TIBCO Designer creates a file that can be shared and locked for each top-level resource, such as an adapter configuration. It does not create a file for each resource. As a result, for example, you can lock an adapter configuration but cannot lock individual adapter services.

When an adapter service is configured, the adapter creates a corresponding set of schema files. A warning is displayed when the files are created advising you to add the files to your version control system. You must add the files to your version control system and ensure they are checked-in, otherwise your project will not be managed correctly by the version control system.

TIBCO Designer also creates folders for folders you create in your project. You can lock each folder as needed.

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.

Additionally, TIBCO Administrator provides the following functionality:

- Access to application repositories. You can choose how the administration server interacts with the application repositories stored as files. You have the following choices:
  - Using Local Application Data

    When using this option, the administration domain uses local application data as the default. This sends the repository instance file of the application to each target machine to which the application is deployed. The application can then run independently of the administration server.
  - Using Server-based Application Data

    When using this option, repository instances for deployed applications are created on the administration server. Adapters send a TIBCO Rendezvous message or HTTP (or HTTPS) request to the TIBCO Administration Server to communicate with the administration server repositories.
- 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 rvrd 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.

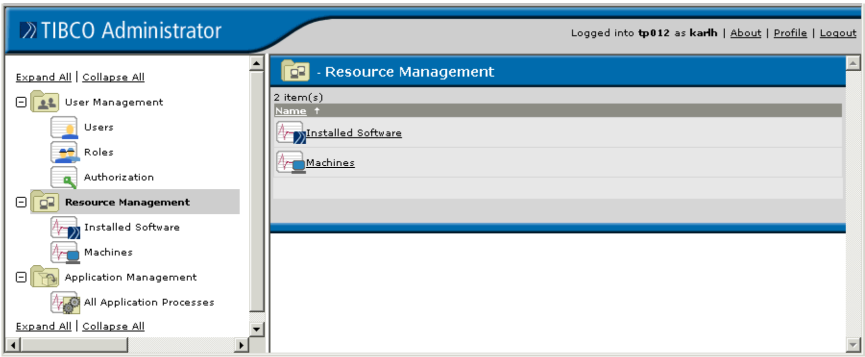
- **Authentication—**The verification of the identity of a person or process.

- **Authorization—**Permission to view or execute. An administrator gives users access rights to the projects. For example, access rights are given to view or to write to projects at design time from TIBCO Designer or to view or to manage modules from the TIBCO Administrator console. The TIBCO Administration Server controls that access.

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 12 TIBCO Administrator GUI



TIBCO ActiveMatrix BusinessWorks

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

In TIBCO ActiveMatrix Adapter for Database, 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 Adapter for Database 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 *TIBCO ActiveMatrix BusinessWorks Concepts* for an introduction to the product.

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.

The microagents are explained in the User's Guide of each adapter.

Chapter 4 Adapter Lifecycle

This chapter explains the lifecycle or sequence of tasks performed when integrating an adapter into a business process.

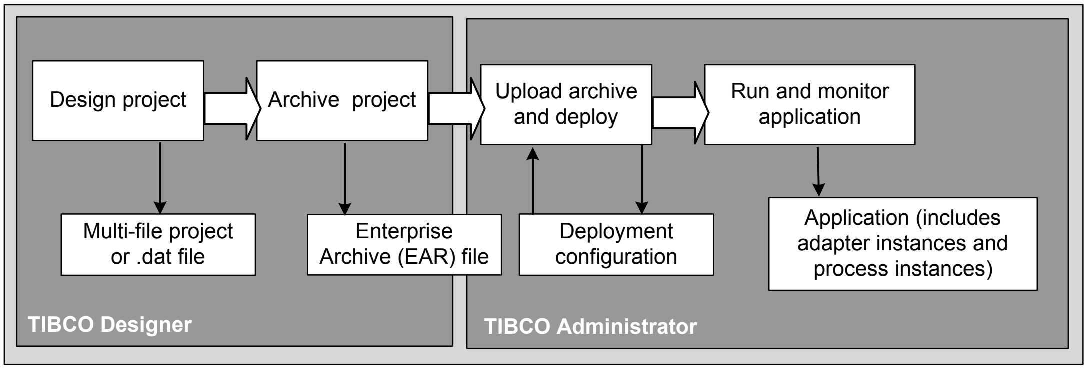
Topics

Overview

Adapters use TIBCO Designer to configure projects and TIBCO Administrator to deploy, run, and monitor projects.

As shown in the next diagram, after a project is created and configured, it is placed into an EAR file and imported into TIBCO Administrator.

Figure 13 TIBCO Application Flow of Information


Design project
Archive project
Upload archive and deploy
Run and monitor application
Multi-file project or .dat file
Enterprise Archive (EAR) file
Deployment configuration
Application (includes adapter instances and process instances)
TIBCO Designer
TIBCO Administrator

Installation

TIBCO Runtime Agent software must be installed before installing an adapter. If the TRA software is not already installed, the adapter installation program warns you, then exits.

You can run the TIBCO Installer in different modes. Each is supported on all platforms.

- GUI mode presents panels that allow you to make choices about product selection, product location, and so on.
- Console mode allows you to invoke the installer and make installation choices from a command prompt or terminal window.
- Silent mode either installs using the default settings or uses a response file that was saved during an earlier installation. You must invoke the installer from the command-line to use silent mode.

See your adapter user's guide for more installation information.

Task A Install Vendor Application

The vendor application should be installed before installing the adapter. The vendor application can be on any machine on your subnet. The machine on which the adapter is installed must have network access to the vendor application.

Task B Install TIBCO Runtime Agent

The TIBCO Runtime Agent must be installed on each machine that hosts an adapter design-time or run-time component before installing the adapter.

Task C Install Adapter(s)

You can install adapter design-time components, run-time components or both on the same machine. A typical environment has the design-time components installed only on machines where adapters are configured. Production machines require only run-time components. TIBCO Runtime Agent must be installed on the machine before installing an adapter design-time or run-time component.

Task D Install TIBCO Administrator

TIBCO Administrator components can be installed on any machine on your subnet as long as it can communicate with the machine that hosts the adapter.

Task E Install other TIBCO Software

Other TIBCO software may be required in your environment. The software can be installed before or after the adapter is installed.

- TIBCO Hawk. Used to monitor and control the run-time adapter.
- TIBCO ActiveMatrix Adapter for Database. An adapter can be included in a BusinessWorks process.
- TIBCO IntegrationManager. An adapter can interoperate with IntegrationManager if complex data mapping is required.
- TIBCO Enterprise Message Service. Must be installed on a machine in your subnet if the JMS transport is used by an adapter.
- TIBCO Rendezvous. TIBCO Rendezvous is installed as part of the TIBCO Runtime Agent installation. You need not install the software separately.

Development

Adapters are configured using TIBCO Designer. This section lists the high-level steps required to configure an adapter. Each of these steps is explained in detail in your adapter user's guide.

Task A Configure the Vendor Application to work with the Adapter

This task is completed after installing the adapter and before configuring an adapter instance for the first time. Each adapter interoperates with its vendor application in different way. See your adapter user's guide for details.

Task B Define an Adapter Project

When starting TIBCO Designer, you create or select a project. A project contains adapter configuration information, such as the service and messaging transport to use, logging options, and other specific settings. A project is opened and saved in muti-file format, which allows a version control system to manage the files associated with the project.

Task C Set Global Variables

By default each project you create in TIBCO Designer includes several global variables. Global variables provide an easy way to set defaults for use throughout your integration project. Default values are predefined for some of the variables. You can define additional variables and, optionally, set their values when configuring your adapter.

When the project is deployed and the configured adapters are run, all occurrences of the global variable name are replaced with the global variable value.

A global variable value set in TIBCO Designer can be overridden at run-time by redefining the value in TIBCO Administrator.

Task D Create an Adapter Instance

An adapter instance is saved in a project and contains all information needed by the run-time adapter. The following tabs are common in most adapter instances. An adapter may have more tabs available along with more options under each tab. These are explained in each adapter's user's guide.

- Configuration tab—Each adapter instance has a unique name in the project. The name is set under this tab. Vendor application-specific options are also set here.

- Logging tab—An adapter can log messages to standard output or to a log file. You can define an adapter instance to use both methods, or one. In addition you can specify that the adapter log messages of type Information, Warning, Error or all three. Advanced logging is also available, which allows you to add specialized log sinks.
- Connection tab—An adapter typically connects to a vendor application or database. Parameters used both at design-time and run-time to access the application or database are set under this tab.
- Monitoring tab—If TIBCO Hawk is used to monitor and control the run-time adapter, and you have written custom microagents, you configure the adapter to use them here.

Task E Configure an Adapter Service

An adapter supports various services: publication, subscription, request-response, and request-response invocation. The services were introduced and explained on Adapter Services Examples on page 6. Each service contains the following tabs. Additional tabs may be available and are specific to each adapter service.

- Configuration tab—The adapter service name, transport type, transport quality of service and wire format to use are specified here.
- Schema tab—The choices available vary by adapter, based on how the adapter interoperates with the vendor application to download schema.
- Encoding tab—Allows you to set encoding options.
- Advanced tab—Allows you to change the transport subject or destination used by the service to send messages. Other advanced fields maybe available and vary by adapter.

After you have configured an adapter instance with at least one service, the project is included in an Enterprise Archive File (EAR). An EAR is imported into TIBCO Administrator when you are ready to deploy.

A project can also be imported or exported in the following formats:

- Local Repository—The project is exported using single-file `.dat` format. This format is convenient if you want to share a single file with other users. It is also included for backward compatibility.
- Server Repository—The project is exported using single-file format and is managed by the administration server. This format also allows you to use the authorization and authentication features of TIBCO Administrator.

Task F Set Security Options

If you have installed TIBCO Administrator with the security module and the TIBCO administration domain, you can use Administrator to set security access to the projects managed by the server. At design-time, only a user who has been authorized with write privileges is allowed to save adapter instances in the project. At run-time, the adapter must use a user account set in Administrator to access the project.

If a local project is used by a run-time adapter, TIBCO Administrator security is not available.

Task G Test the Adapter Configuration

The adapter tester can be used to verify an adapter service after it has been configured. When invoked, all adapter services configured in the project are displayed. You select the adapter service to test, and start and stop the adapter from the tester. The tester window displays adapter output within the tool so you can easily view results.

Deployment

After an integration project has been developed and tested, it is necessary to deploy the run-time components to the machines on which they will ultimately run in a production environment. An adapter instance can be deployed, started and managed using TIBCO Administrator from the Administrator web browser.

Deploy Using TIBCO Administrator

During development, you save your design to a project. When you are ready to deploy your project to a machine, you generate an Enterprise archive file (EAR file) from TIBCO Designer. The EAR file contains information on what you wish to deploy.

Building the archive creates the EAR file, which you can then deploy from TIBCO Administrator. If you make changes to the adapters in the archive, you need to rebuild the archive. Saving the project does not affect the archive.

Task A Generate and Import an Enterprise Archive File

An Enterprise Archive file contains adapter instance configuration information, which is used by a run-time adapter. An Enterprise Archive file is generated using TIBCO Designer and imported into TIBCO Administrator.

Task B Specify Deployment Information

After importing an Enterprise Archive file, the adapter can be deployed. This involves:

- Assigning adapter services to the machines in the administration domain.
- Specifying startup options for each process engine and adapter service.

Task C Specify Monitoring Options

Before starting the adapter you can optionally specify monitoring options, including:

- Specifying alerts or TIBCO Hawk rulebases for each machine.
- Specifying alerts and TIBCO Hawk rulebases for an adapter service.
- Setting log file properties for an adapter service instance.

Task D Start the Adapter

The adapter is started and stopped using the TIBCO Administrator GUI.

Management and Monitoring

There are several ways to monitor a run-time adapter:

- TIBCO Administrator provides the Application Management module, which is used to manage and monitor the machines in your TIBCO administration domain.
- You can use TIBCO Hawk to manage a run-time adapter. Each adapter has built-in microagents that can be accessed using Hawk. The microagents are documented in the adapter user's guide.
- Each adapter maintains a log file where error, warning, and information messages are written.
- Each message sent by an adapter contains a trace message. The message identifies which process has handled the message.
- If a run-time adapter throws an exception, an error is displayed in the console window where the adapter was started. You can use the displayed error code to find the corresponding description and resolution in the adapter's User Guide.

TIBCO Administrator

At run-time, TIBCO Administrator allows you to access remotely all deployments in your administration domain. Depending on your security privileges, you may be able to perform one or more of the following tasks.

- **User Management—**Add users and passwords to the administration domain for authentication, then give each user view or execute privileges for TIBCO Administrator GUI elements or project repositories (data stores).
- **Domain Monitoring and Management—**View the status of machines and components running on machines in the domain. Start and stop components as needed.
- **Deployment monitoring—**Monitor the status of each deployment component (process engine and adapter) and process instance and the status of the machines executing them. View trace files and throughput.
- **Deployment management—**View all running components and stop and restart them as needed.

These aspects of monitoring and management are supported by TIBCO Administrator, which includes the TIBCO Administration Server and the TIBCO Administrator GUI.

TIBCO Hawk

TIBCO Hawk allows monitoring, management, and fault detection. It can be customized according to the user's requirements and be used to monitor any TIBCO product on the network. TIBCO Hawk is especially suited for very large distributed systems where customization is required.

If TIBCO Hawk is installed at a site, Hawk microagents can be used to supplement the monitoring information provided by the standard logging capability. Examples of supplemental information that you can obtain with microagents are:

- Current configuration settings for this adapter instance
- URL of the project used for adapter instance
- Command line arguments used to start the adapter

In a TIBCO Hawk environment, agents on each local computer can perform the monitoring work. A TIBCO Hawk Agent is an autonomous process that uses microagents to monitor local applications and systems activity. Microagents represent managed objects such as operating system subsystems, agent components, log files, event logs, and applications. Each microagent exposes a set of methods to the agent for collecting information and taking actions. This design allows for separation between management data and management rules or policies.

See your adapter user's guide for a list of supported microagents.

Log Files

By default all error, warning and information messages are printed in the console window in which the adapter instance starts and to a default log file. The log file can be located anywhere on the file system.

When a log file name is specified in the Logging tab, the adapter:

- Creates a file with no extension, using the file name specified at design-time.
- Redirects all trace messages generated by this configuration to that file until it reaches the file size specified at design-time.
- When this file size is reached (that is, as soon as the file is greater than or equal to the limit specified at design-time), renames the current file to file.1 and creates a new file with no extension.
- Repeats this process of rolling log files and renaming files each time a new file is generated, until three log files exist.

Information can also be written to the log file in the overwrite mode, rather than the default append mode. In overwrite mode, the adapter instance overwrites the existing log file when it is next started. In append mode, trace information is appended to existing entries in the log file.

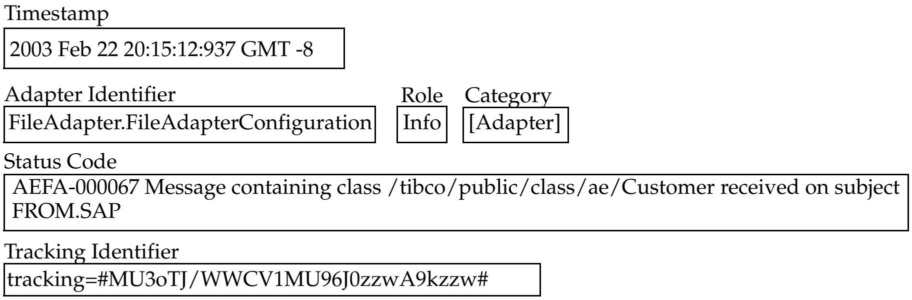
Trace Messages

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

Each trace message can include the following fields:

```
<Timestamp> <Adapter Identifier> <Role> <Category> <Status Code>
<Tracking Identifier>
```

The above fields are explained in your adapter user’s guide. The following diagram shows an example trace message and calls out the fields.



Timestamp

2003 Feb 22 20:15:12:937 GMT -8

Adapter Identifier

FileAdapter.FileAdapterConfiguration

Role

Info

Category

[Adapter]

Status Code

AEFA-000067 Message containing class /tibco/public/class/ae/Customer received on subject FROM.SAP

Tracking Identifier

tracking=#MU3oTJ/WWCV1MU96J0zzwA9kzzw#

Status Codes

An adapter includes a status code in a tracing message. A status code provides information such as connection errors, startup errors, and certain adapter service errors. These messages provide details about the error, are contextual and are applied uniformly across adapters.

Low-level exception messages like System and TIBCO Rendezvous messages have appropriate adapter-level information. Status code descriptions and resolutions are available in the respective adapter user’s guide.

Tracking Identifier

A tracking identifier is "stamped" on each significant trace message by the originating adapter. This allows you to track the progress of a message as it travels among TIBCO applications.

Chapter 5 Adapter Capabilities

This chapter explains key features available for most adapters.

Topics

Message Transports

Adapter messages can be delivered using either the TIBCO Rendezvous transport or the JMS transport. This section introduces the transports and explains the quality of services available under each.

To understand fully the TIBCO Rendezvous features used by adapters, you should read *TIBCO Rendezvous Concepts*. The book is available in the TIBCO Runtime Agent installation area in the *<install-path>*`/tibco/tibrv/doc` folder.

Before configuring an adapter service with the JMS transport, familiarize yourself with JMS concepts, which are explained fully in the *TIBCO Enterprise Message Service User's Guide*. The book is available from the machine on which the TIBCO Enterprise Message Service software is installed in the *<install-path>*`/tibco/ems/doc` directory.

TIBCO Rendezvous

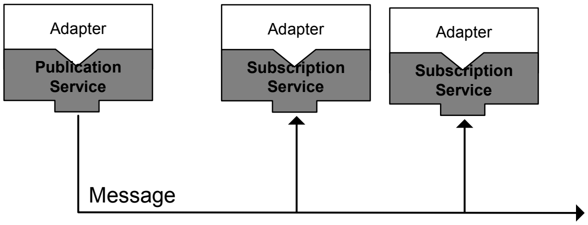
TIBCO Rendezvous integrates *distributed* applications by providing a common, message-based communication mechanism. One main component is the TIBCO Rendezvous daemon, which runs on each participating computer on your network. All information that travels between adapter instances passes through the TIBCO Rendezvous daemon as the information enters and exits host computers. The daemon also passes information between adapters running on the same host. Many hardware and software platforms are supported, so programs on different kinds of computers on a network can communicate seamlessly.

TIBCO Rendezvous software uses subject-based addressing™ technology to direct messages to their destinations. An adapter configured with a publication service sends that information on a well-defined subject name. Each adapter configured with a subscription service listens on a subject name. The TIBCO Rendezvous daemon handles the network messages corresponding to the peer-to-peer communication between the publication adapter and the subscription adapter. This allows adapters to communicate without knowing the details of network addresses or connections. Subject-based addressing conventions define a uniform name space for messages and their destinations.

Publication and Subscription Services Interactions

The publication and subscription services interactions are driven by events (usually the arrival or creation of data). An adapter publication service makes information available for general distribution. Communication is in one direction (publisher to subscribers), and often one-to-many as shown in the next diagram. The complete interaction consists of one multicast message, published once, and received by all subscribers.

Figure 14 Event-driven publication and subscription interaction

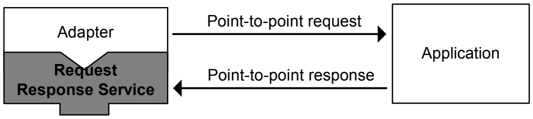


In publication and subscription services interactions, data producers are decoupled from data consumers—they do not coordinate data transmission with each other, except by using the same subject name. The adapter publishes data to the network at large.

Request and Response Services Interactions

Demand for data drives request response interactions. An adapter requests data from a vendor application; the application computes an individual response and returns it to the adapter. Communication flows in both directions, as shown in the next diagram. The complete interaction consists of two point-to-point messages—a request and a response.

Figure 15 Demand-driven request and response interaction



In request interactions, data producers coordinate closely with data consumers. An adapter does not send data until a request is made by a vendor application. Each program sends its message to a specific inbox name within the other program.

The adapter sends its response specifically to the vendor application that requested the data. The requesting application listens until it receives the reply, and then stops listening (unless it expects further installments of information).

Reliable and Certified Message Delivery Modes

If your adapter uses the TIBCO Rendezvous transport, the reliable and certified delivery protocols are available. Table 2 compares the two delivery protocols.

Table 2 Comparing Reliable and Certified Message Delivery (Sheet 1 of 2)

Aspect	Reliable Delivery	Certified Delivery
Protocol Information	TIBCO Rendezvous daemons inform programs when data is lost. No information about the lost data is available.	The library presents advisory messages to inform programs of every significant event related to certified delivery. Advisories identify specific messages by correspondent name, subject name, and sequence number.
Ledger	None.	The certified delivery library records outbound messages in a ledger, either within the program process storage, or in file storage.
Time Limit	Outbound messages are retained for 60 seconds.	The certified delivery library retains outbound messages in the ledger until either delivery is complete or the time limit (set by the adapter) expires.
Effective Range	60 seconds, or when the TIBCO Rendezvous daemon process terminates—whichever is first.	With persistent correspondents, certified delivery can extend beyond program process restart. It is not affected by TIBCO Rendezvous daemon process termination.
Network Bandwidth	Minimal network overhead beyond the message itself.	Additional network overhead to confirm delivery of each certified message.

Table 2 Comparing Reliable and Certified Message Delivery (Sheet 2 of 2)

Aspect	Reliable Delivery	Certified Delivery
File Storage	No file storage overhead.	Optional file-based ledgers consume file storage for each message until delivery is complete (or the time limit expires).

Distributed Queue Mode

This mode allows multiple adapters to distribute the load among the distributed queue members. The service is used for adapters configured with the subscription service or request response service to provide process level load balancing.

Note that a distributed queue cannot be used for adapters configured with the publication or request response invocation services. See the *TIBCO Rendezvous Concepts* book for more information.

Wire Formats

The adapter can be configured to send messages in one of the following wire formats.

- TIBCO ActiveEnterprise wire format — the standard, recommended format

  An adapter should generally use the TIBCO ActiveEnterprise wire format to send messages. Messages that use the TIBCO ActiveEnterprise wire format are TIBCO Rendezvous messages with some structure imposed. The structure includes control information that makes it easier for the receiving application to process the message.

- XML Message

  The `XML Message` wire format conforms to specifically constructed and fully compliant XML Schema (XSD) based on the existing definition of the TIBCO ActiveEnterprise schema.

- TIBCO Rendezvous wire format

  An adapter can send messages using the TIBCO Rendezvous wire format. This wire format does not include control information in the message. This wire format should only be used for communicating with applications configured for the TIBCO environment that do not support the TIBCO ActiveEnterprise wire format.

The TIBCO Rendezvous wire format should be chosen to communicate with a native TIBCO Rendezvous application while the TIBCO ActiveEnterprise wire format should be chosen to communicate among TIBCO applications that use TIBCO ActiveEnterprise Operations for their service configurations.

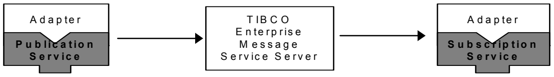
TIBCO Enterprise Message Service

This section provides an overview of the TIBCO Enterprise Message Service product. See the *TIBCO Enterprise Message Service User's Guide* for more information.

JMS is based on creation and delivery of messages. Messages are structured data that an adapter sends to another adapter or vendor application. An adapter publication service produces messages that are consumed by an adapter subscription service. The TIBCO Enterprise Message Service server acts as an intermediary for the message and sends it to the correct destination.

The next diagram shows an adapter publication service producing a message, sending it by way of the server, and an adapter subscription service receiving the message.

Figure 16 JMS Message Delivery



JMS supports publish and subscribe (topics), and point-to-point (queues) interactions.

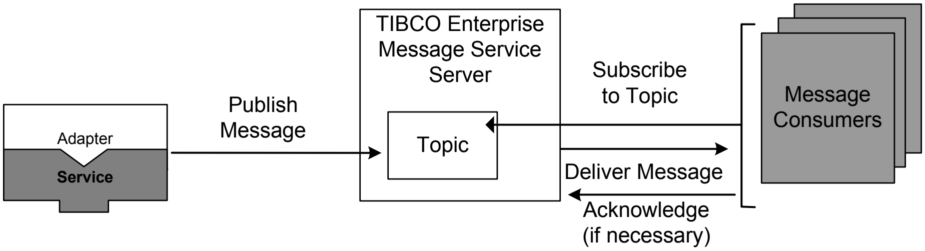
Publish and Subscribe Interactions

In a publish and subscribe message system, producers address messages to a topic. In this model, the producer is known as a *publisher* and the consumer is known as a *subscriber.*

Many publishers can publish to the same topic, and a message from a single publisher can be received by many subscribers. Subscribers subscribe to topics, and all messages published to the topic are received by all subscribers to the topic. This type of message protocol is also known as *broadcast* messaging because messages are sent over the network and received by all interested subscribers, similar to how radio or television signals are broadcast and received.

The next diagram shows publish and subscribe messaging. Each message consumer subscribes to a topic. When a message is published to that topic, all subscribed consumers receive the message.

Figure 17 JMS publish and subscribe messages



There can be a time dependency in publish and subscribe mode. By default, subscribers only receive messages when they are active. If messages are delivered when the subscriber is not available, the subscriber does not receive those messages. The JMS API provides a method to remove part of the timing dependency by allowing subscribers to create durable subscriptions. Messages for durable subscriptions are stored on the server until the message expires or the storage limit is reached. Subscribers can receive messages from a durable subscription even if the subscriber was not available when the message was originally delivered.

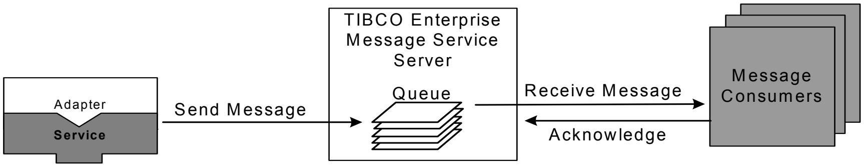
Point-to-Point Interactions

Point-to-point messaging has one producer and one consumer per message. This style of messaging uses a *queue* to store messages until they are received. The message producer sends the message to the queue; the message consumer retrieves messages from the queue and sends acknowledgement that the message was received.

More than one producer can send messages to the same queue, and more than one consumer can retrieve messages from the same queue. The queue can be configured to be exclusive, if desired. If the queue is exclusive, then all queue messages can only be retrieved by the first consumer specified for the queue. Exclusive queues are useful when you wish for only one application to receive messages for a specific queue. If the queue is not exclusive, any number of receivers can retrieve messages from the queue. Non-exclusive queues are useful for balancing the load of incoming messages across multiple receivers. Regardless of whether the queue is exclusive or not, only one consumer can ever retrieve each message that is placed on the queue.

The next diagram shows point-to-point messaging using a non-exclusive queue. Each message consumer receives a message from the queue, acknowledges receipt of the message, and the message is taken off the queue so that no other consumer can receive it.

Figure 18 JMS point-to-point messages



Wire Format

JMS messages use the `XML Message` wire format that conforms to specifically constructed and fully compliant XML Schema (XSD) based on the existing definition of the TIBCO ActiveEnterprise schema.

Connection Type

The following delivery modes can be specified when using the JMS transport:

- Topic, where an adapter service publishes to a topic or subscribes to a topic. Messages are sent over the network and received by all interested subscribers, similar to how radio or television signals are broadcast and received.
- Queue, where an adapter service sends to a queue or receives from a queue. This message protocol is known as point-to-point because messages sent to a queue are consumed by one and only one receiver. A queue retains all messages sent until such time the messages are consumed or expired. Each message has only one receiver, though multiple receivers can connect to the queue.

Delivery Mode

A JMS message is marked as *persistent* or *non persistent*.

- Persistent — In general, a message marked persistent will be available to a JMS client even if the JMS server goes down.
- Non Persistent — A message marked non persistent will not be available to a JMS client if the JMS server goes down.

Multithreading

The adapter communicates with the target application to process requests. Prior to processing the requests, the adapter creates threads to process the request. These threads are created by the adapter during its initialization.

Multithreading is the ability of the adapter to manage multiple requests from and to the target application without having multiple adapter instances running. Based on the load, as and when requests arrive; they are scheduled among the available threads. Each thread has a unique identity. The adapter keeps track of the status of a request processed by each thread.

To receive requests from a target application, dispatcher threads are created by the adapter. Each adapter creates dispatcher threads in a different way (this is explained in detail in the following section). To send requests to a target application, SDK dispatcher threads are used by the adapter.

Based on the requirements of a given implementation, the adapter can be configured to run in a multithreaded mode or with multiple adapter instances (RCVMQ). The merits and demerits of both are listed in the following sections.

Running the Adapter in Multithreading Mode

- Merit—When properly configured according to the memory and processing power of a machine, multithreading leads to better memory management and improved performance.
- Demerit—If a machine has insufficient memory or the adapter has too many threads configured, overall performance may be adversely affected.

Running the Adapter in RVCMQ Mode

- Merit—Running multiple adapter instances prevents loss of messages. If one instance terminates, the other instances take-over and handle the messages thereby providing fault tolerance functionality. This mode also provides the load balancing functionality, as multiple adapter instances share the load.
- Demerit—If multiple instances of the adapter are running, more memory is used. Introducing the intermediary scheduler increases the message latency, which affects the throughput in a synchronous scenario.

Connection Management

This section explains the following connection management concepts:

- Connection Pooling
- Reconnection Mechanism
- Connection Monitoring

Connection Pooling

Connection pooling is an efficient way of sharing of connections between different resources through creation, recycling, and release of connections.

The adapter uses connection pooling to enhance the performance while communicating with a target application. Opening and maintaining a connection for each user or each request made to a target application is expensive in terms of both time and memory consumed while opening every new connection.

This is where the connection pool comes in. The adapter maintains a pool of already-established connections to the target application; so a thread can simply request a connection from the pool when it needs to, use it, and then release it back to the pool.

After a connection is created, it is placed in the pool and it is re-used so that a new connection does not have to be established for each new task. If all the connections are being used, a new request for a connection will have to wait until a connection in the pool becomes available. Connection pooling also cuts down on the amount of time a thread must wait because it does not have to incur the delay inherent in establishing a connection to the target application.

Reconnection Mechanism

Sometimes, the adapter might lose the connection with a target application due to various reasons such as network problems or target application connectivity issues.

In such a scenario, the adapter uses the reconnection mechanism to restore dynamically the connection to the target application. The connection pool keeps track of the number of connections it established to the target application during its operation. After the network problems or target application connectivity issues are resolved, the adapter attempts to re-open the same number of connections to the target application prior to connection loss.

Each adapter provides reconnection configuration options that can be set using the TIBCO Designer.

An adapter service is suspended when the connection to the vendor application is broken. Upon reconnect, the adapter service resumes. No manual intervention is required. Here is how each type of service handles the reconnection scenario:

- Publication Service — The publication service maintains a log that includes a timestamp attribute for each message sent. Before a message is sent, the service records information about the message in its log file. If the service loses its connection, upon reconnect, the service checks the log for the last message timestamp and proceeds by sending the next message, based on its timestamp relative to the last message in the log.
- Subscription and Request-Response Services — The subscription and request-response invocation services update a vendor application with new or additional information present in the received message. If the connection is down, the service queues its messages and polls the vendor application until the connection comes back up. When the connection is established, updates on the vendor application are performed based on the queued messages.
- Request Response Service — The request-response service provides updates or retrieval operations against the vendor application. In the event of a connection loss, a connection error message is immediately sent back in the response message, which allows the sending process to become aware of the connection problem. If the connection is down when a request is sent, the adapter attempts to reestablish the connection (for future requests) without delaying the response.

Connection Monitoring

The connection state can be monitored using TIBCO Hawk. An adapter sends an unsolicited notification to Hawk whenever adapter services are suspended. These unsolicited notifications are visible from all custom microagents defined for the adapter.

If the adapter:

- starts before the vendor application is started, the adapter displays an error message and stops.
- is configured with multiple services and one connects to the vendor application and another service does not, the services that connect will operate normally. The services that do not connect will start the reconnection procedure.
- cannot connect to the vendor application, trace messages that track reconnection attempts are logged to the adapter’s log file.

Polling Mechanism

Polling is a mechanism used by adapters to retrieve periodically data from a vendor application that stores data in a permanent store like a database. The settings for the polling interval and polling size can affect adapter performance. Setting too small a polling size or too large a polling interval can slow your adapter's performance if the vendor application itself is relatively fast. Conversely, setting the polling interval to be too small can create too much of a load on the vendor application.

Definitions

- Event data — Changes to application data that are stored as events in a permanent store. An example of a store is a database. The event data may be in the form of database records or reference keys that are used to retrieve actual data by invoking application API(s).
- Event store or Publishing Table — The permanent store containing event data. The event store is also known in many adapters as a Publishing Table, or a Staging table. The event store may either be populated by setting triggers or registering callbacks within the application or by an adapter agent that is notified by the application of event data. The changes to application data are written (as actual data or reference to data) into the event store by plug-ins or triggers setup in the application instance during adapter configuration.
- Publication Service — The adapter component that polls event data from the event store and publishes them to the TIBCO environment on the configured subject for that event type.
- Poller— Component of the publication service that polls for the event data from the event store.

How the Polling Mechanism Works

The publication service in the adapter retrieves data from the store and publishes the desired information about the event in the TIBCO environment. For example, when an application updates a table in a database monitored by an adapter such as TIBCO ActiveMatrix Adapter for Database, the adapter extracts data from the changed rows from database tables and publishes them on appropriate subjects. An adapter polls (checks/selects) the publishing tables periodically for any new rows to be published. This method is most efficient when the publishing tables change frequently and a limited number of database operations is preferred.

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.

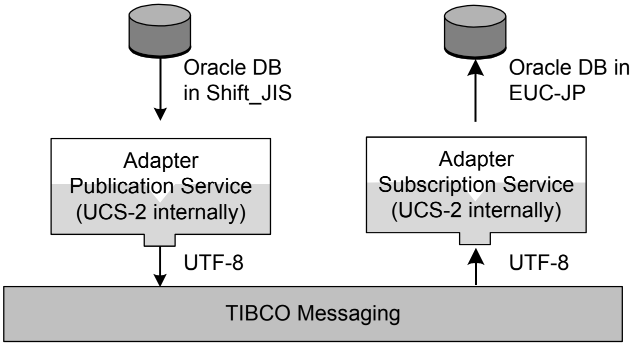
How TIBCO Adapters Support Unicode

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. This is shown in the next diagram where an example of a TIBCO ActiveMatrix Adapter for Database conversion between database character strings and Unicode strings is given.

Figure 19 Unicode Conversion Example



Shift_JIS data retrieved from database by the adapter is converted to UTF-8, which is published using a TIBCO messaging transport. Another adapter, configured with a subscription service, converts the received message from UTF-8 into EUC-JP (another Japanese encoding) and inserts the data into the EUC-JP database to which it is connected.

By using UTF-8 as the TIBCO messaging encoding, the two adapters connecting to databases of different encoding can exchange data without data loss. Also, the systems where the two adapters are running need not be in the same locale. In the above scenario, as an example, the adapter publication service can run on a Microsoft Windows platform with Shift_JIS as the locale encoding, while the adapter subscription service can run on a Solaris platform with EUC-JP as the locale encoding.

Using UTF-8 as the TIBCO messaging encoding doesn't disallow a native encoding to be used by the adapter to communicate with its vendor application (SAP, Siebel, database, etc). The encoding for communicating with the vendor application depends on the nature of the data exposed by the application. See your adapter User's Guide for more information.

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 Adapter for Database 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 <*install-path*>`/tibco/administrator/`*n.n*`/bin/`).

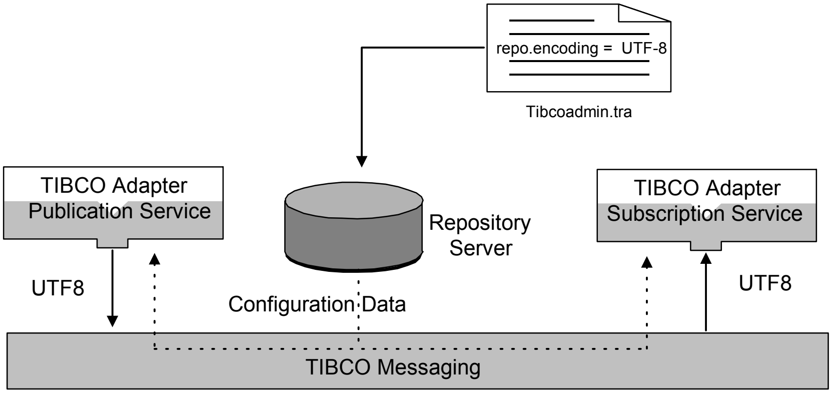
The encoding is set when using the TIBCO Domain Utility to create the domain or by editing the `repo.encoding` property in the `.tra` configuration file.

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 <*install-path*>`/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 20 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 Appendix B, Encoding Tables, on page 77 for a list of additional encodings supported by most adapters.

Documentation Roadmap

The adapter interoperates with other TIBCO products. This section provides a roadmap to the books that describe the TIBCO infrastructure software. The books can be installed with the product and are available from the products root directory in the doc directory. Note that only manuals relevant to adapters are listed in the table. Each product may have additional manuals.

Table 3 Documentation roadmap

Product	Documentation Set	Location on Disk
TIBCO Designer	*TIBCO Designer User's Guide* *TIBCO Designer Palette Reference* *TIBCO Designer Release Notes*	<*install-path*>`\tibco\designer\`*n.n*`\doc`
TIBCO Administrator	*TIBCO Administrator User's Guide* *TIBCO Administrator Server Configuration Guide* *TIBCO Administrator Installation Guide* *TIBCO Administrator Release Notes*	<*install-path*>`\tibco\administrator\`*n.n*`\doc`
TIBCO Enterprise Message Service	*TIBCO Enterprise Message Service User's Guide* *TIBCO Enterprise Message Service Installation* *TIBCO Enterprise Message Service Application Integration* *TIBCO Enterprise Message Service Release Notes*	<*install-path*>`\tibco\ems\doc`

Table 3 Documentation roadmap

Product	Documentation Set	Location on Disk
TIBCO Rendezvous	*TIBCO Rendezvous Concepts* *TIBCO Rendezvous Installation* *TIBCO Rendezvous Administration* *TIBCO Rendezvous Configuration Tools*	*<install-path>*`\tibco\tibrv\doc`
TIBCO ActiveMatrix BusinessWorks	*TIBCO ActiveMatrix BusinessWorks Concepts* *TIBCO ActiveMatrix BusinessWorks Quick Start* *TIBCO ActiveMatrix BusinessWorks Process Design Guide* *TIBCO ActiveMatrix BusinessWorks Palette Reference* *TIBCO ActiveMatrix BusinessWorks Installation* *TIBCO ActiveMatrix BusinessWorks Release Notes*	*<install-path>*`\tibco\bw\`*n.n*`\doc`
TIBCO IntegrationManager	*TIBCO IntegrationManager Concepts* *TIBCO IntegrationManager Administrator's Guide* *TIBCO IntegrationManager Process Design Guide* *TIBCO IntegrationManager Release Notes*	*<install-path>*`\tibco\im\`*n.n*`\doc`
TIBCO Hawk	*TIBCO Hawk Installation and Configuration* *TIBCO Hawk Administrator's Guide* *TIBCO Hawk Release Notes*	*<install-path>*`\tibco\hk\`*n.n*`\doc`

Table 3 Documentation roadmap

Product	Documentation Set	Location on Disk
TIBCO Runtime Agent	*TIBCO Runtime Agent Installation Guide* *TIBCO Runtime Agent Domain Utility User's Guide* *TIBCO Runtime Agent Scripting Deployment User's Guide* *TIBCO Runtime Agent Installing Into a Cluster* *TIBCO Runtime Agent Upgrading to Release 5.4* *TIBCO Runtime Agent Release Notes*	*<install-path>*`\tibco\tra\`*n.n*`\doc\tra`
TIBCO Adapter SDK	*TIBCO Adapter SDK Concepts* *TIBCO Adapter SDK Message Codes* *TIBCO Adapter SDK Programmer's Guide* *TIBCO Adapter SDK Release Notes*	*<install-path>*`\tibco\tra\`*n.n*`\doc\sdk`
TIBCO Adapters	Each TIBCO Adapter typically provides the following: *User's Guide* *Examples Guide* *Releases Notes* Some adapters provide a separate installation guide, but most installation instructions are found in the adapter user's guide.	*<install-path>*`\tibco\adapters\`*<adapter-name>*`\`*n.n*`\doc`

Appendix A Adapter Types

TIBCO Adapters are available for a wide range of off-the-shelf applications, databases and files, network technologies, and mainframe applications. Custom adapters can be developed using the TIBCO Adapter SDK. Adapter SDK is a development platform for building adapters that automatically interoperate with TIBCO ActiveEnterprise applications.

TIBCO Adapters provide shared XML metadata definitions, standard error handling and reporting, and the same GUIs for configuration, deployment, and monitoring.

Topics

Application Adapters

Application adapters for a wide range of applications from leading vendors are supported. Each adapter integrates with at least one, and usually several, of the interfaces exposed by the applications.

TIBCO Adapters are available for many applications and platforms. Note that more adapters are available than listed here, including adapters for Broadvision, Infranet, Metasolv, JDE, Remedy, and Tuxedo.

Note that this appendix provides a short summary of each adapter. See your adapter user's guide for more information.

TIBCO ActiveMatrix Adapter for SAP

The adapter enables bi-directional, real-time, and asynchronous communication with SAP R/3. The adapter supports the following SAP R/3 integration interfaces:

- **BAPI (Business Application Programming Interface)** — This interface provides an object-oriented view of SAP's business objects and enables easy integration of non-SAP applications in the TIBCO environment with SAP's Business Integration Framework. This interface is best used in concert with the adapter for performing realtime queries and updates to and from the SAP R/3 system. For example, creation of a sales order in SAP R/3 from a legacy order management system in the TIBCO environment. The BAPI is SAP's recommended integration interface over RFC (Remote Function Call).
- **RFC interface** — This interface is the most popular SAP R/3 access technology. It allows non-SAP R/3 and SAP R/3 applications to execute SAP Advanced Business Application Programming (ABAP) functions that have been RFC-enabled and forms the basis of ALE/IDoc and BAPI interfaces. The RFC interface is functionally handled similar to the BAPI by the adapter and is typically used when a BAPI is not provided by SAP to perform the same task. RFCs are also the quickest way to integrate highly customized scenarios.
- **ALE/IDoc Interface** — ALE/IDocs or Intermediate Documents are SAP defined document containers for business transactional data, such as purchase or sales orders and order invoices. SAP's middleware to integrate different SAP R/3 systems together to begin with, ALE helps achieve cross-business functionality and facilitates the event driven, asynchronous exchange of information directly out of business transactions. This interface is typically used with the adapter for non-realtime and asynchronous transfer of business data to and from the SAP R/3 system, for example, a batch transfer of sales orders from SAP R/3 to a data warehouse.

TIBCO Adapter for Siebel

The adapter exports or imports data into a Siebel system and other external systems in real time. A Business Event Specification is defined to describe how data is imported into and exported from the Siebel system. The Business Event Specification is configured according to the Siebel Business Object model.

Siebel eBusiness applications are customized primarily through creating and modifying object definitions. Programming is supported through the following facilities in Siebel applications:

- **Siebel VB and Siebel eScript** — enables developers to write event procedures, known as scripts, that are associated with certain object types and events. These scripts can be used to attach additional validation logic to a business component or interface with external applications by invoking methods of a Microsoft COM object. Siebel VB is available only on Microsoft Windows.
- **Java Data Bean Interface** — provides an open interface to Siebel applications, supporting integration between Siebel applications and external applications. For example, an application can use the Siebel Object Interfaces for importing data into Siebel from an external data source and exporting data from Siebel to external applications in an event-driven manner. Using the Siebel Object Interface, the adapter can access Siebel data using the Siebel Java Bean that is independent of the underlying database platform, while at the same time preserving the Siebel business rules built into the Business Component Objects.
- **Siebel Business Service** — simplifies the task of moving data and converting data formats between Siebel and external applications.

  Siebel business services can be written in Siebel VB or Siebel eScript. They can then be accessed by Siebel VB or Siebel eScript code and called from the Siebel workflow process.

  Through the use of Siebel VB and Siebel eScript, the adapter can be invoked to send or request data to or from external applications in the outbound scenarios.

  Using the custom Business Service 'EAI TIBCO HTTP Agent' provided with the adapter installation, publication requests can be forwarded to the Siebel Adapter Publication Services and work seamlessly with Siebel Workflow process and other Siebel Business Services.

TIBCO Adapter for PeopleSoft 8

The adapter is a bidirectional gateway between a PeopleSoft 8 application and the TIBCO environment. The adapter can integrate with any PeopleSoft 8 component including PeopleSoft's CRM module and the PeopleTools 8.4 application.

TIBCO Adapter for ClarifyCRM

The adapter exports data from and imports data into a ClarifyCRM system in an event-driven fashion. For example, data is exported by the adapter when a user saves data to the ClarifyCRM database. When the adapter receives a message containing data to import, the adapter inserts the data into the ClarifyCRM database.

TIBCO Adapter for Teradata

The adapter uses CLI (Call Level Interface) to integrate with Teradata applications. CLI is a collection of callable service routines that provide an interface to the Teradata RDBMS. The Teradata RDBMS communicates with the external world using CLI APIs; this allows the adapter and Teradata RDBMS to be deployed on remote machines.

The adapter also uses JDBC to integrate with Teradata applications. JDBC is a specification for an application programming interface (API). This API allows platform independent Java applications to access database management systems using Structured Query Language (SQL).

TIBCO Adapter for Remedy

The Remedy Action Request System (AR System) provides an application foundation for IT Service Management solutions, such as the Remedy Help Desk, Remedy Asset Management and custom-built applications.

The AR System consists of an Action Request server, which interacts with the Remedy database and provides information to client applications, independent of the database that the client applications use.

The adapter is a bi-directional gateway between an AR System and applications configured for the TIBCO environment. All communication between the Remedy database and the adapter passes through the Action Request server except for the Publication service, which interacts directly with the Remedy database to get data from the publishing table. The adapter uses the AR System's APIs to interface with the Action Request server.

TIBCO Adapter for Tuxedo

TIBCO 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 the Subscription, Request-Response and Request-Response Invocation services, the adapter uses the appropriate method (communication paradigms) to invoke a Tuxedo Service, which reads, writes from, or writes into buffers.

For the Publication service, all communication between Tuxedo and the adapter is either through the EventBroker or the Adapter Agent.

TIBCO Adapter for Lotus Notes

TIBCO Adapter for Lotus Notes processes schemas between Lotus Notes applications through TIBCO Rendezvous in real-time. It maintains data integrity in both directions, and integrates other software in the TIBCO environment seamlessly into an enterprise.

The adapter receives messages from external applications that are integrated with TIBCO Rendezvous. Messages are passed to Lotus Notes and documents are published from the Lotus Notes database using TIBCO Rendezvous. Publishing is carried out by agents in the Lotus Notes databases. The Lotus Notes agents can be initiated manually, or scheduled to run automatically.

TIBCO Adapter for J.D. Edwards OneWorld Xe

TIBCO Adapter for J. D. Edwards OneWorld Xe allows one-way (publish or subscribe) or two-way (request-response) message exchange between TIBCO ActiveEnterprise applications and the J. D. Edwards OneWorld Xe enterprise resource planning (ERP) system.

The adapter interoperates with the J. D. Edwards OneWorld Xe system by making business function calls, extracting data using query lists, and publishing real-time events.

Adapters for Database and File Systems

Adapters can connect directly to databases or files systems to query or place data. In addition to the following, adapters are available to transfers files from IBM AS/400 and OS/390 external applications to applications that reside on operating systems other than IBM AS/400 or OS/390.

TIBCO ActiveMatrix Adapter for Database

The adapter monitors a database for changes and sends them as they occur to other databases and applications. It extends publish-subscribe and request-response services to databases, making multiple levels of delivery services available to applications that access data from a database. ODBC and JDBC compliant databases such as Oracle, Sybase, and Microsoft SQL Server are supported. While the adapter does not run on OS/390 and AS/400 systems, it can remotely connect to a DB2 database running on these systems.

TIBCO Adapter for Oracle Applications

TIBCO Adapter for Oracle Applications allows TIBCO ActiveEnterprise applications to communicate with Oracle Applications using publish and subscribe with reliable, certified, and transacted message delivery.

The adapter allows third-party applications to send updates, inserts, and deletes to Oracle Applications. It also allows Oracle Applications to publish to other applications. The adapter includes pre-configured publication and subscription transactions.

TIBCO ActiveMatrix Adapter for Files

The adapter provides a publication and subscription service. The publication service reads text files and publishes their contents. The subscription service subscribes to messages and writes their contents to a text file.

Adapters for Other Technologies

TIBCO Adapters enable integration with component or object development models as well as with other messaging technologies. In fact, integrating applications based on these network technologies into a TIBCO-enabled e-business infrastructure can expand their functionality to include features such as publication and subscription service level messaging, and load-balancing.

TIBCO Adapters for Network Technologies are available for the following technologies. Note that the list is not complete. Adapters are available for CORBA, CICS and other technologies.

TIBCO Adapter for LDAP

The adapter allows applications configured for the TIBCO environment to communicate with LDAP servers without any knowledge of LDAP protocol, LDAP APIs, and LDAP server implementation. Applications can retrieve information from or update LDAP servers, or receive changes occurring on the LDAP servers.

TIBCO Adapter for COM

The adapter allows you to use ordinary COM interfaces to send messages to or receive messages from applications configured for the TIBCO environment. You do not need to write complex code to manage a messaging session or to marshal your data into messages. Instead, you develop a COM interface and then use a graphical administration tool to set certain configuration parameters.

TIBCO Adapter for CORBA

TIBCO Adapter for CORBA enables the integration of CORBA and TIBCO ActiveEnterprise applications. The adapter provides a standalone process which interoperates between your CORBA applications and TIBCO ActiveEnterprise. There is no need to re-compile your existing CORBA or TIBCO ActiveEnterprise applications. Information about your CORBA and TIBCO ActiveEnterprise applications is entered into a repository that contains configuration information for the adapter.

TIBCO Adapter for CORBA supports the CORBA Internet Inter-ORB Protocol (IIOP) which allows it to interoperate with CORBA applications built using third-party ORBs.

TIBCO Adapter for EJB

The adapter enables the integration of EJB and applications that operate in the TIBCO environment. This adapter currently supports version 1.3 of the Java 2 Platform Enterprise Environment (J2EE) specification. It also supports the J2EE Connector API, which is part of the J2EE version 1.3 specification.

TIBCO Adapter for MQSeries

The adapter facilitates message flow between MQSeries applications, applications configured for the TIBCO environment, and compatible third-party applications. The adapter takes messages received from applications configured for the TIBCO environment, transforms the message if necessary, and then places this data on an MQSeries queue where it can be accessed by MQSeries applications. Conversely, the adapter extracts data from an MQSeries queue and converts it into a TIBCO wire format for consumption by external applications that subscribe to this data. Request-Response interactions are also supported between TIBCO Rendezvous and MQSeries.

TIBCO Adapter for SWIFT

The adapter enables exchange of data between the SWIFT network and applications configured for the TIBCO environment. It sends and receives SWIFT messages to and from the SWIFT network. It also parses, validates, and unparses SWIFT messages for publishing and subscribing in the TIBCO environment. This can be done in two ways: automatically by running the adapter, or programmatically using the Java API. The API enables SWIFT messages to be used for private messaging without using the SWIFT network.

Custom Adapters

To write a custom adapter program, developers use the TIBCO Adapter Software Development Kit (SDK). The Adapter SDK is a class library available in C++ and Java on multiple platforms. The Adapter SDK also includes several example programs for commonly found behavior.

TIBCO Adapter SDK provides the foundation for common adapter functionality, such as sending and receiving information, metadata management, configuration, and tracing. The SDK includes classes that allow access to the data saved during configuration in the project file. The program then uses these publishers, subscribers, sinks, etc.

Using the Adapter SDK allows custom adapters interoperability with other TIBCO ActiveEnterprise products such as other packaged adapter products and TIBCO IntegrationManager.

Custom adapters are configured using TIBCO Designer (the same tool used to configure other adapters). The Generic Adapter Configuration palette allows configuration of the sessions and endpoints, that is, publishers, subscribers, clients, or servers in TIBCO Designer. Configuration includes specifying the messaging protocol to be used, and specifying which session type should handle which endpoint. Both standard and custom logging behavior can also be configured. Adapter metadata determines which data the adapter sends or receives. For this configuration, TIBCO Designer supports schema resources.

Appendix B Encoding Tables

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

Topics

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.

- Set 1

  Your adapter may support more encoding values than given in the adapter's Encoding field pick list. The values in this column represent the underlying encoding used by the run-time adapter. You can type an encoding value listed in this column in the adapter's Encoding field. The palette does not verify whether the value you provide is supported. If the value is not supported, the adapter displays an error at runtime.

  Set 1 column values apply to the following adapters written in C++:

  — TIBCO ActiveMatrix Adapter for Database

  — TIBCO ActiveMatrix Adapter for Files

  — TIBCO Adapter for MQSeries

  — TIBCO Adapter for Oracle Applications

  — TIBCO ActiveMatrix Adapter for SAP

- Set 2

  Your adapter may support more encoding values than given in the adapter's Encoding field pick list. The values in this column represent the underlying encoding used by the run-time adapter. You can type an encoding value listed in this column in the adapter's Encoding field. The palette does not verify whether the value you provide is supported. If the value is not supported, the adapter throws an error at runtime.

  Set 2 column values apply to the following adapters written in Java:

  — TIBCO Adapter for ClarifyCRM

  — TIBCO Adapter for PeopleSoft 8

  — TIBCO Adapter for Siebel

- 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 4 ISO Character Sets

Value in Encoding Field Pick List	Set 1	Set 2	IBM CCSID	Oracle NLS String	Description
ASCII	ascii	ASCII	IBM-367	US7ASCII	7-bit ASCII
ISO-8859-1	LATIN_1	ISO8859_1	IBM-819	language_territory. WE8ISO8859P1	ISO8859-1 (Latin-1), West European
ISO-8859-2	ibm-912	ISO8859_2	IBM-912	language_territory.EE8ISO8859P2	ISO8859-2 (Latin-2), East European
ISO-8859-3	ibm-913	ISO8859_3	IBM-913	language_territory.SE8ISO8859P3	ISO8859-3 (Latin-3), South European
ISO-8859-4	ibm-914	ISO8859_4	IBM-914	language_territory. NEE8ISO8859P4	ISO8859-4 (Latin-4), North European
ISO-8859-5	ibm-915	ISO8859_5	IBM-915	language_territory.CL8ISO8859P5	ISO8859-5, Cyrillic
ISO-8859-6	ibm-1089	ISO8859_6	IBM-1089	ARABIC_UNITED ARAB EMIRATES.AR8ISO8859P6	ISO8859-6, Arabic
ISO-8859-7	ibm-813	ISO8859_7	IBM-813	GREEK_GREECE.EL8ISO8859P7	ISO8859-7, Greece
ISO-8859-8	ibm-916	ISO8859_8	IBM-916	HEBREW_ISRAEL.IW8ISO8859P8	ISO8859-8, Hebrew

Table 4 ISO Character Sets

Value in Encoding Field Pick List	Set 1	Set 2	IBM CCSID	Oracle NLS String	Description
ISO-8859-9	ibm-920	ISO8859_9	IBM-920	TURKISH_TURKEY .WE8ISO8859P9	ISO8859-9 (Latin-5), Turkey
ISO-8859-13	ibm-921	ISO8859_13	IBM-921	N/A	ISO8859-13, Baltic
ISO-8859-15	ibm-923	ISO8859_15 _FDIS	IBM-923	N/A	ISO8859-15 (Latin-9), Latin-1 with Euro
Shift_JIS (CP943)	ibm-943	CP943	IBM-943	JAPANESE_JAPAN. JA16SJIS	Japanese Shift-JIS, CP943
KSC_5601	ibm-949	KSC5601	IBM-949	KOREAN_KOREA. KO16KSC5601	Korean KSC-5601
Big5	ibm-1370	Big5	IBM-1370	TRADITIONAL CHINESE_TAIWA N.ZHT16BIG5	Traditional Chinese Big5 (with Euro Sign)
GBK	ibm-1386	GBK	IBM-1386	SIMPLIFIED CHINESE_CHINA. ZHS16CGB231280	Simplified Chinese GBK, super set of GB2312-80
EUC-JP	tibx-eucJP	EUC_JP	IBM-954	JAPANESE_JAPAN. JA16EUC	Japanese EUC
EUC-KR	ibm-970	EUC_KR	IBM-970	KOREAN_KOREA. KO16KSC5601	Korean EUC
EUC-CN	ibm-1383	EUC_CN	IBM-1383	SIMPLIFIED CHINESE_CHINA. ZHS16CGB231280	Simplified Chinese EUC, compatible with GB2312-80
Shift_JIS (TIBCO)	tibx-943	Shift_JIS	N/A	JAPANESE_JAPAN. JA16SJIS	Variant of IBM-943, flavoring some MS-932 conversions
Shift_JIS(932)	tibx-932	MS932	IBM-932	JAPANESE_JAPAN. JA16SJIS	Shift_JIS encoding in MS-932 flavour

Table 4 ISO Character Sets

Value in Encoding Field Pick List	Set 1	Set 2	IBM CCSID	Oracle NLS String	Description
UTF8	UTF8	UTF8	IBM-1208	AMERICAN_AMERICA.UTF8	Unicode Transformation Format-8
UTF16_BigEndian	UTF16_BigEndian	UnicodeBig	N/A	N/A	UTF16 (UCS-2) Big Endian
UTF16_LittleEndian	UTF16_LittleEndian	UnicodeLittle	N/A	N/A	UTF16 (UCS-2) Little Endian

The next table lists the EBCDIC character sets.

Table 5 EBCDIC Character Sets

Encoding Value	Set 1	Set 2	IBM CCSID	Oracle NLS String	Description
IBM-37	ibm-37	CP037	IBM-37	N/A	EBCDIC United States
IBM-273	ibm-273	CP273	IBM-273	N/A	EBCDIC Germany, Austria
IBM-277	ibm-277	CP277	IBM-277	N/A	EBCDIC Denmark, Norway
IBM-278	ibm-278	CP278	IBM-278	N/A	EBCDIC Finland, Sweden
IBM-280	ibm-280	CP280	IBM-280	N/A	EBCDIC Italy
IBM-284	ibm-284	CP284	IBM-284	N/A	EBCDIC Spain
IBM-285	ibm-285	CP285	IBM-285	N/A	EBCDIC UK Ireland
IBM-297	ibm-297	CP297	IBM-297	N/A	EBCDIC France
IBM-420	ibm-420	CP420	IBM-420	N/A	EBCDIC Arabic
IBM-424	ibm-424	CP424	IBM-424	N/A	EBCDIC Hebrew

Table 5 EBCDIC Character Sets

Encoding Value	Set 1	Set 2	IBM CCSID	Oracle NLS String	Description
IBM-500	ibm-500	CP500	IBM-500	N/A	EBCDIC Latin-1 (International)
IBM-1148	ibm-1148	CP1148	IBM-1148	N/A	EBCDIC Latin-1 (with Euro Sign)
IBM-1047	ibm-1047	N/A	IBM-1047	N/A	EBCDIC Open System Latin-1
IBM-1153	ibm-1153	N/A	IBM-1153	N/A	EBCDIC Latin-2 (with Euro Sign)
IBM-1154	ibm-1154	N/A	IBM-1154	N/A	EBCDIC Cyrillic
IBM-1156	ibm-1156	N/A	IBM-1156	N/A	EBCDIC Baltic
IBM-1157	ibm-1157	N/A	IBM-1157	N/A	EBCDIC Estonia
IBM-290	ibm-290	N/A[a]	IBM-290	N/A	EBCDIC Japanese Katakana (SBCS).
IBM-1159	ibm-1159	N/A[b]	IBM-1159	N/A	EBCDIC Traditional Chinese (SBCS)
IBM-300	ibm-16684	N/A[c]	IBM-300	N/A	EBCDIC Japanese (DBCS)
IBM-834	ibm-834	N/A[d]	BM-834	N/A	EBCDIC Korean (DBCS)
IBM-835	ibm-835	N/A[e]	IBM-835	N/A	EBCDIC Traditional Chinese (DBCS)
IBM-837	ibm-935	N/A	IBM-837	N/A	Forward compatible with IBM-935
IBM-930	ibm-930	CP930	IBM-930	N/A	EBCDIC Japanese (MBCS)
IBM-933	ibm-933	CP933	IBM-933	N/A	EBCDIC Korean (MBCS)

Table 5 EBCDIC Character Sets

Encoding Value	Set 1	Set 2	IBM CCSID	Oracle NLS String	Description
IBM-935	ibm-935	CP935	IBM-935	N/A	EBCDIC Simplified Chinese (MBCS)
IBM-937	ibm-937	CP937	IBM-937	N/A	EBCDIC Traditional Chinese (MBCS)
IBM-1390	ibm-1390	N/A	IBM-1390	N/A	EBCDIC Japanese (MBCS) with Euro
IBM-1364	ibm-1364	N/A	IBM-1364	N/A	EBCDIC Korean (MBCS) with Euro
IBM-1371	ibm-1371	N/A	IBM-1371	N/A	EBCDIC Traditional Chinese (MBCS) with Euro

a. Forward compatible with CP930

b. Forward compatible with CP937

c. Forward compatible with CP930

d. Forward compatible with CP933

e. Forward compatible with CP937

The next table lists the Microsoft Windows character sets

Table 6 Microsoft Windows Character Sets

Encoding Value	Set 1	Set 2	IBM CCSID	Oracle NLS Encoding	Description
Windows Latin-1	ibm-1252	CP1252	IBM-1252	language_territory.WE8MSWIN1252	Windows Latin-1 with Euro Sign
Windows Latin-2	ibm-1250	CP1250	IBM-1250	language_territory.EE8MSWIN1250	Windows Latin-2 with Euro Sign
Windows Cyrillic	ibm-1251	CP1251	IBM-1251	language_territory.CL8MSWIN1251	Windows Cyrillic with Euro Sign
Windows Greek	ibm-1253	CP1253	IBM-1253	GREEK_GREECE.EL8MSWIN1253	Windows Greek with Euro Sign

Table 6 Microsoft Windows Character Sets

Encoding Value	Set 1	Set 2	IBM CCSID	Oracle NLS Encoding	Description
Windows Turkish	ibm-1254	CP1254	IBM-1254	TURKISH_TURKEY.TR8MSWIN1254	Windows Turkish with Euro Sign
Windows Hebrew	ibm-1255	CP1255	IBM-1255	HEBREW_ISRAEL.IW8MSWIN1255	Windows Hebrew with Euro Sign
Windows Arabic	ibm-1256	CP1256	IBM-1256	ARABIC_UNITED ARAB EMIRATES.AR8MSWIN1256	Windows Arabic with Euro Sign
Windows Baltic	ibm-1257	CP1257	IBM-1257	language_territory.BLT8MSWIN1257	Windows Baltic with Euro Sign