TIBCO ActiveMatrix® Adapter for IBM i

Concepts

Software Release 6.0
March 2010

Important Information

Contents

Figures

Tables

Preface

TIBCO ActiveMatrix Adapter for IBM i allows one-way (publish or subscribe) or two-way (request-response) message exchange between TIBCO applications and applications running on IBM System i machines.

This manual explains the basic concepts and supported features of the adapter.

Topics

Related Documentation

This section lists documentation resources you may find useful.

TIBCO ActiveMatrix Adapter for IBM i Documentation

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

- *TIBCO ActiveMatrix Adapter for IBM i Concepts* Read this manual before reading any other book in the documentation set to familiarize yourself with the product and its uses.
- *TIBCO ActiveMatrix Adapter for IBM i Installation* Read this manual for instructions on site preparation and installation.
- *TIBCO ActiveMatrix Adapter for IBM i Configuration and Deployment* Read this manual for instructions on creating, configuring, and deploying standalone adapter projects.
- *TIBCO ActiveMatrix Adapter for IBM i Examples* Read this manual to work through the examples provided with the adapter.
- *TIBCO ActiveMatrix Adapter for IBM i Release Notes* Read the release notes for a list of new and changed features. This document also contains lists of closed and known issues for this release.

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

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

Other TIBCO Product Documentation

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

- TIBCO ActiveMatrix BusinessWorks™
- TIBCO Adapter™ SDK
- TIBCO Administrator™
- TIBCO Designer™
- TIBCO Enterprise Message Service™
- TIBCO Hawk®
- TIBCO Rendezvous®
- TIBCO Runtime Agent™
- TIBCO ActiveEnterprise™

Third-Party Documentation

You may also find it useful to read the IBM documentation on the following web sites:

- http://www-03.ibm.com/systems/i/software/toolbox/
- http://jt400.sourceforge.net/

Typographical Conventions

The following typographical conventions are used in this manual.

Table 1 General Typographical Conventions

Convention	Use
TIBCO_HOME *ENV_HOME* *ADAS400_HOME*	Many TIBCO products must be installed within the same home directory. This directory is referenced in documentation as *TIBCO_HOME*. The value of *TIBCO_HOME* depends on the operating system. For example, on Windows systems, the default value is `C:\tibco`. Other TIBCO products are installed into an installation environment. Incompatible products and multiple instances of the same product are installed into different installation environments. The directory into which such products are installed is referenced in documentation as *ENV_HOME*. The value of *ENV_HOME* depends on the operating system. For example, on Windows systems the default value is `C:\tibco`. TIBCO ActiveMatrix Adapter for IBM i installs into a directory within *TIBCO_HOME*. This directory is referenced in documentation as *ADAS400_HOME*. The value of *ADAS400_HOME* depends on the operating system. For example on Windows systems, the default value is `C:\tibco\adapter\adas400\6.0`.
`code font`	Code font identifies commands, code examples, filenames, pathnames, and output displayed in a command window. For example: Use `MyCommand` to start the foo process.
`bold code font`	Bold code font is used in the following ways: • In procedures, to indicate what a user types. For example: Type **`admin`**. • In large code samples, to indicate the parts of the sample that are of particular interest. • In command syntax, to indicate the default parameter for a command. For example, if no parameter is specified, `MyCommand` is enabled: `MyCommand [`**`enable`**` | disable]`

Table 1 General Typographical Conventions (Cont'd)

Convention	Use
italic font	Italic font is used in the following ways: • To indicate a document title. For example: See *TIBCO ActiveMatrix BusinessWorks Concepts*. • To introduce new terms. For example: A portal page may contain several portlets. *Portlets* are mini-applications that run in a portal. • To indicate a variable in a command or code syntax that you must replace. For example: `MyCommand` *PathName*
Key combinations	Key name separated by a plus sign indicate keys pressed simultaneously. For example: Ctrl+C. Key names separated by a comma and space indicate keys pressed one after the other. For example: Esc, Ctrl+Q.
	The note icon indicates information that is of special interest or importance, for example, an additional action required only in certain circumstances.
	The tip icon indicates an idea that could be useful, for example, a way to apply the information provided in the current section to achieve a specific result.
	The warning icon indicates the potential for a damaging situation, for example, data loss or corruption if certain steps are taken or not taken.

Table 2 Syntax Typographical Conventions

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

Table 2 Syntax Typographical Conventions (Cont'd)

Convention	Use

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

How to Contact TIBCO Support

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

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

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

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

  https://support.tibco.com

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

Chapter 1 Introduction

This chapter introduces basic concepts of adapters.

Topics

Adapter Overview

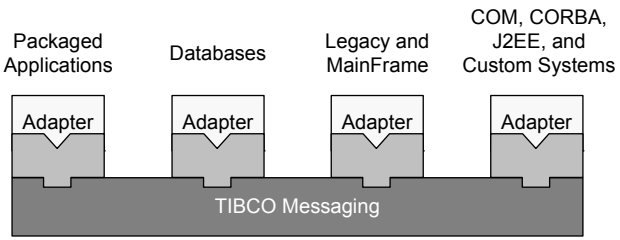
To deploy the best solution for each aspect of a business, it is usually needed to purchase applications from different application vendors. However, vendors typically have their own ways to format and expose data. Therefore, integrating the various applications across an enterprise poses significant challenges.

An adapter provides a bridge between an application and the TIBCO integration environment. Using a no-coding approach, TIBCO adapters enable packaged applications, databases, and other technologies to participate actively in the enterprise information flow, regardless of their data formats or communication protocols. Integration of new applications does not require programming 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 Figure 1, adapters allow for the exchange of data among different technologies.

Figure 1 Adapters Provide a Bridge for Data



- Adapters are available for off-the-shelf applications from leading vendors. Each adapter integrates with one or more interfaces exposed by the vendor application.
- Database adapters enable the database of an enterprise to initiate 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 adapters and the rest business applications and databases of a company.
- Adapters can also enable integration with component or object development models and other messaging technologies.

Adapter Components

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

Standalone

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

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

Adapter Service Engine

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

Adapter service engine projects are created using TIBCO Business Studio and deployed using TIBCO ActiveMatrix Administrator. The adapter component is wired with other composite elements in the ActiveMatrix SOA Project.

Existing standalone adapter configurations can also be deployed as services.

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

Adapter Key Terms

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

- A *palette* is a standalone adapter component that contains the screens used to gather input at design-time when configuring an adapter with a service. The palette is accessed through TIBCO Designer.
- A *project* is a collection of configured adapter resources and it contains configuration information for one or more adapter instances.

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

Adapter Services

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

General Adapter Services

This section lists four kinds of services which can be found in most of TIBCO adapter products. Not all adapters provide all these services and some adapters may provide services not listed here. See Adapter Services on page 31 for information about services available on TIBCO ActiveMatrix Adapter for IBM i.

Publication Service

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

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

Subscription Service

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

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

Request-Response Service

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

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

Request-Response Invocation Service

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

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

Adapter Services Summary

Table 3 summarizes the services introduced in this section.

Table 3 Adapter Services Summary

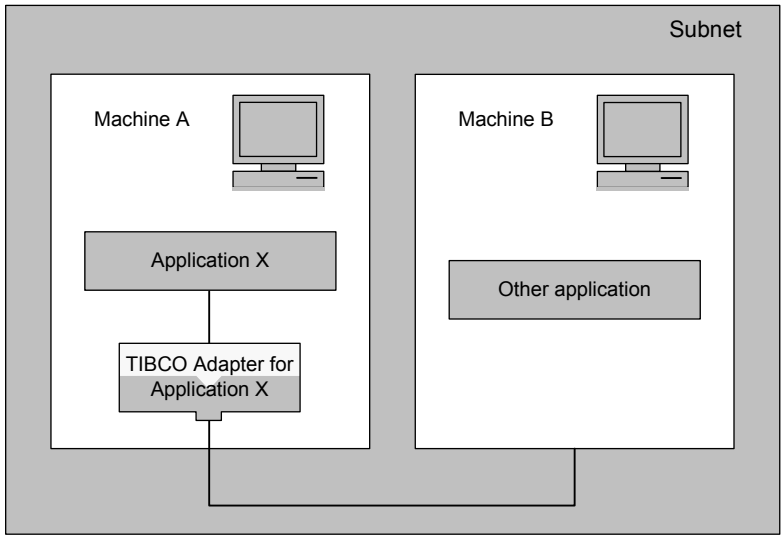
Service	Initiator	Target	Event Mode
Publishing Service (sends to target)	Vendor application	TIBCO infrastructure	Asynchronous
Subscribing Service (gets from initiator)	TIBCO infrastructure	Vendor application	Asynchronous
Request-Response Service (gets from initiator, waits for response then sends response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

Choosing an Adapter Service

A business integration scenario drives the choice of one adapter service or another. This section provides a simple flow chart that helps you to choose the service to use.

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

Figure 2 A Business Integration Scenario



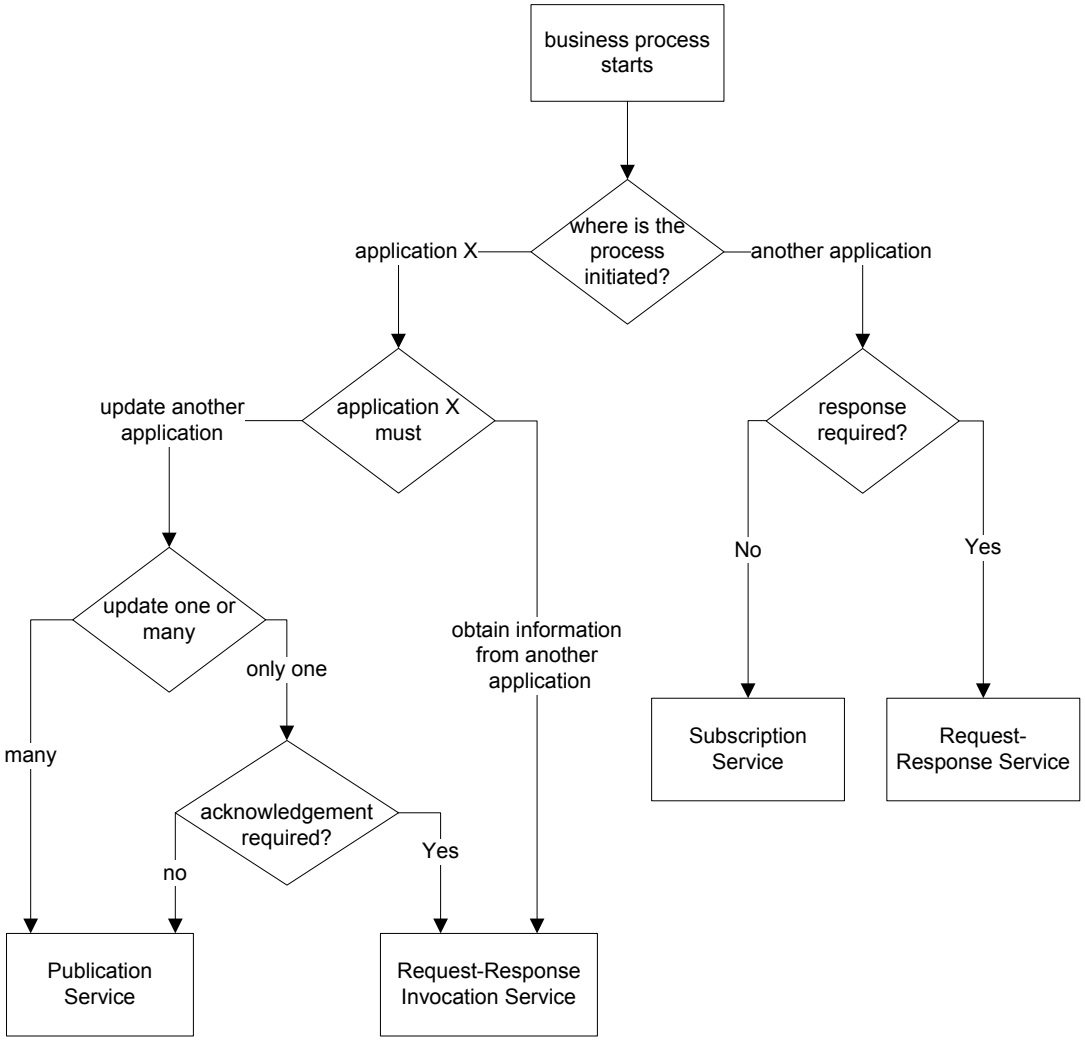
In this scenario, data is exchanged between the application X and another application. The other application could be a customer management system, such as PeopleSoft, or another TIBCO application, such as TIBCO ActiveMatrix BusinessWorks.

To choose the adapter service to use, start by finding out where the scenario begins or what triggers it.

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

This question is the starting point of the decision chart provided in Figure 3.

Figure 3 Choosing an Adapter Service



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 publication service should be used.

Adapter Life Cycle

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

Installation

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

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

Configuration

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

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

Deployment

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

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

Monitoring

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

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

Chapter 2 Adapter Infrastructure Tools

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

Topics

TIBCO Runtime Agent

The TIBCO Runtime 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 through the TIBCO Administrator GUI.
- Supplies the runtime 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. This is an advanced option performed only by users familiar with TIBCO Rendezvous. If you want to 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. This is an advanced option. You must perform it on the machine that hosts the TIBCO Administration Server.
- Remove a secondary TIBCO Administration Server.
- Enable TIBCO administration domain and security management on a machine that hosts TIBCO Administrator.
- Migrate previous TIBCO Administrator installations.

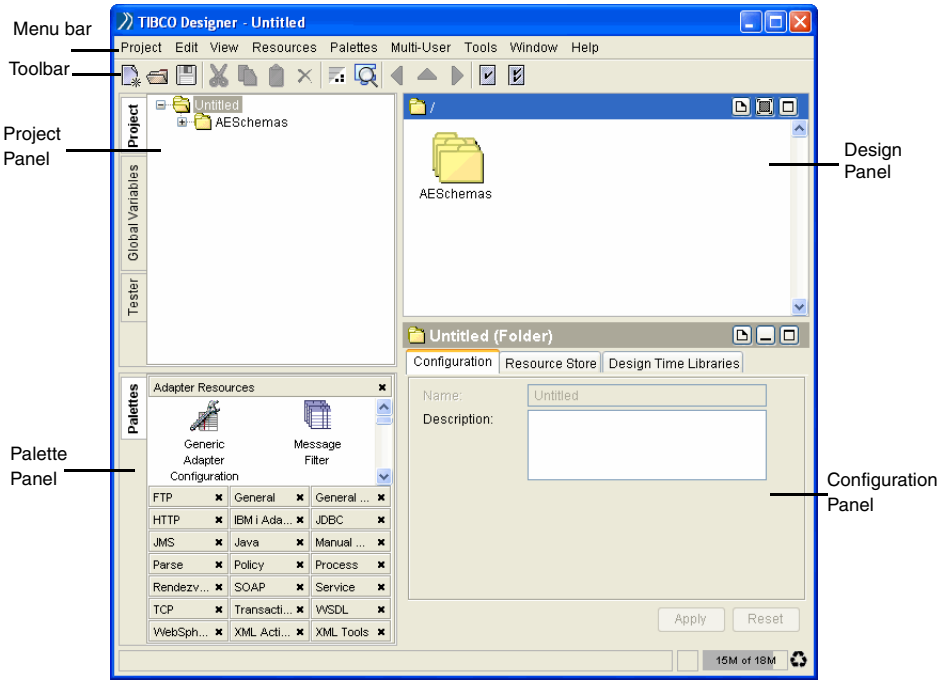
TIBCO Designer

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

Before using TIBCO Designer, ensure that you have read the *TIBCO Designer documentation*. The documentation can be accessed via the TIBCO Designer **Help > Designer Help** from the menu bar. Figure 4 shows the TIBCO Designer interface.

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

Figure 4 TIBCO Designer Main Window



TIBCO Administrator

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

User Management This module allows you to set permissions for adapter users. You can 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 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 level.

Application Management This module allows you to upload Enterprise Archive (EAR) files, and create, configure, and deploy adapters. This module is also used to start and stop adapters.

Load balancing An adapter can be served by a primary and secondary TIBCO Administration Server. The primary server allows read and write operations, while the secondary server supports read operations. Load balancing is implemented through the use of the TIBCO Rendezvous distributed queue protocol (RVDQ) and therefore is 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. Refer to the IP Redirector or HTTP Server documentation for instructions 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 also 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 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. Each machine can belong to only one TIBCO administration domain.

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

TIBCO Administration Server

The TIBCO Administration 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 projects and requests to manage the TIBCO administration domain.

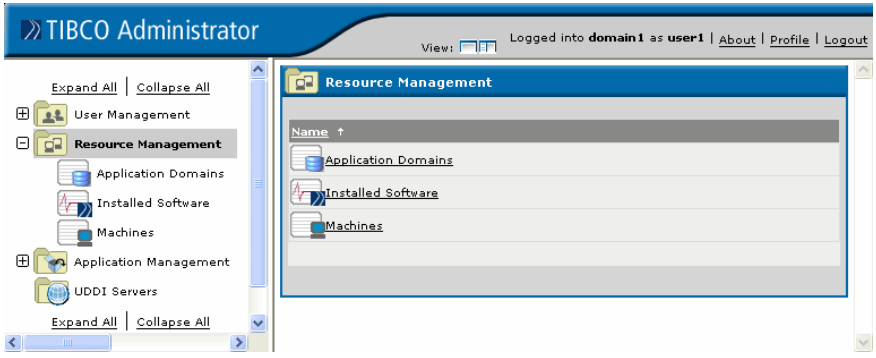
The TIBCO Administrator Server contains its own web server (Apache Tomcat) that can be accessed through the TIBCO Administrator GUI for configuration and monitoring information.

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

TIBCO Administrator GUI

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

Figure 5 TIBCO Administrator GUI



TIBCO ActiveMatrix BusinessWorks

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

In TIBCO ActiveMatrix BusinessWorks, adapter services publish or subscribe to business data in a decoupled yet reliable manner. The business process receives data from, and routes data to, an adapter service.

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

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

See the TIBCO ActiveMatrix BusinessWorks documentation for more information.

TIBCO ActiveMatrix BusinessWorks Service Engine

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

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

TIBCO Hawk

TIBCO Hawk monitors and manages distributed applications and systems throughout the enterprise. System administrators can monitor application parameters, behavior, and loading activities for all nodes in a local or wide-area network and take action when pre-defined conditions occur. In many cases, runtime 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 can be found and fixed before failure occurs.
- Hawk Application Management Interface (AMI) routines can be embedded within custom adapters, allowing active management of those adapters by the Hawk microagents.
- Distributed microagents 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.

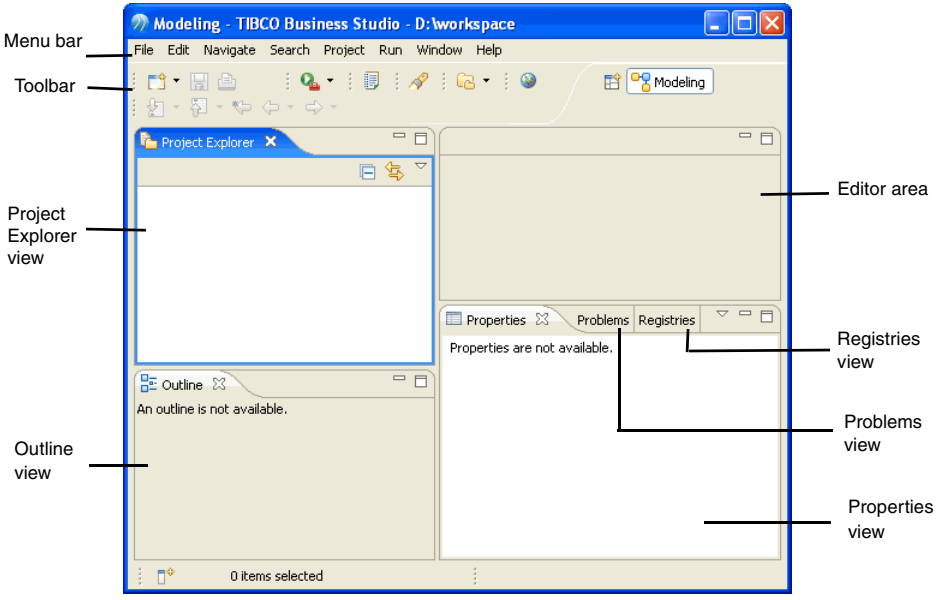
TIBCO Hawk consists of several components: a console display, a central repository for storage of configuration objects, agents, and microagents whose monitoring duties are defined by the rule bases.

- 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.

TIBCO Business Studio

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

Figure 6 TIBCO Business Studio Workbench Window



Following are some important concepts you need to know before starting to work in TIBCO Business Studio:

Perspective A perspective is a group of views and editors in the Workbench window. One or more perspectives can exist in a single Workbench window. Each perspective contains one or more views and editors.

View A view is a visual component within the Workbench. It is typically used to navigate a hierarchy of information (such as the resources in the Workbench), open an editor, or display properties for the active editor.

Editor An editor is also a visual component within the Workbench. It is typically used to edit or browse a resource. The Composite Editor contains a canvas on which you can drop elements and a palette that organizes the elements that you can add to the composite.

Project Explorer view Displays a tree containing all the project resources such as project folders, shared resource definition files, WSDL files, composite files, service assembly files, and so on.

Outline view Provides a overview of the Composite Editor canvas. You can easily navigate from one part of a composite to another. The Outline view also displays a content tree structure that contains the composite element inside the composite. In this view you can delete the contents of the composite. When you select a composite element in the Outline tree, the corresponding artifact in the composite becomes selected.

Properties view Displays property sheets for editing composites and composite elements. When you select a composite or composite element in the Composite Editor canvas, this view shows the properties of the selected object in a vertical tabbed notebook.

Problems view Displays validation and other errors.

Registries view Lists UDDI registries and the WSDL files returned from searching a registry.

ActiveMatrix Resource Wizard

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

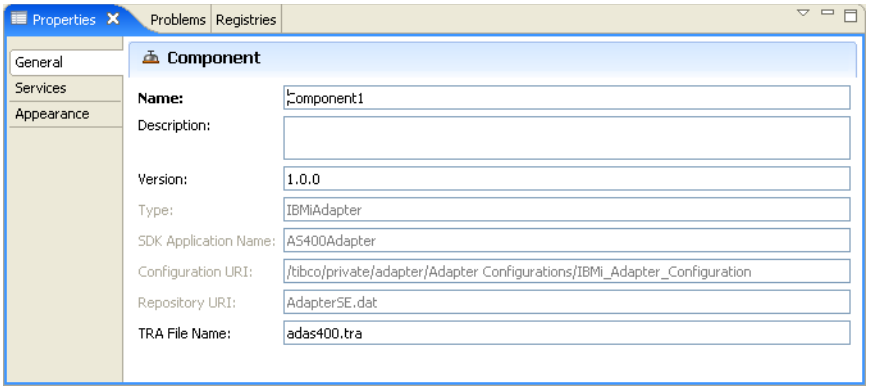
- Adapter service engine projects
- ActiveMatrix sample projects
- ActiveMatrix SOA projects
- Composites
- Mediation flows
- Service assemblies

Composite Element Editors

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

Figure 7 shows an example of the Properties view.

Figure 7 Component Property Sheet

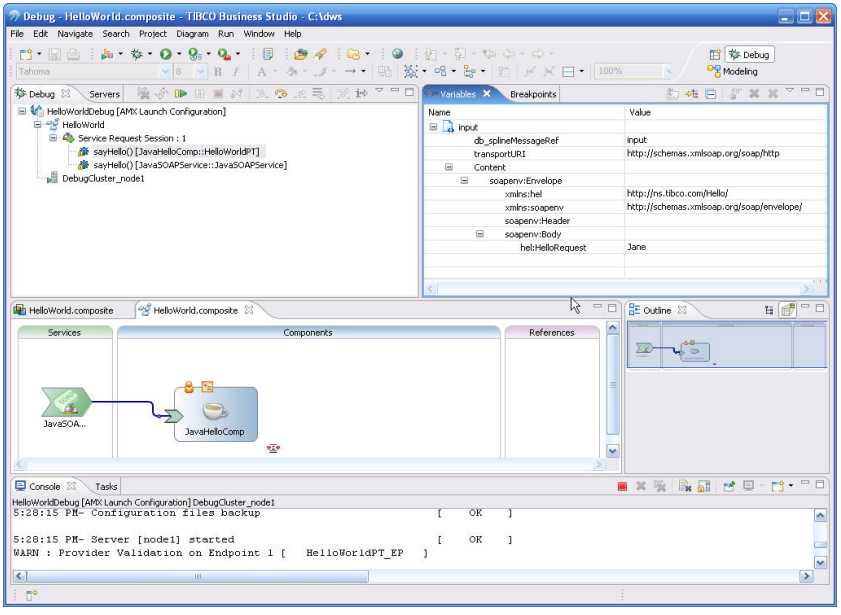


Debugger

The TIBCO Business Studio debugger provides a testing environment to step through composite elements and locate errors.

Figure 8 shows the debugger in the process of debugging a sample HelloWorld composite. When the Java component executes, the debugger stops at pre-defined breakpoints. Values can be checked in the Variables editor.

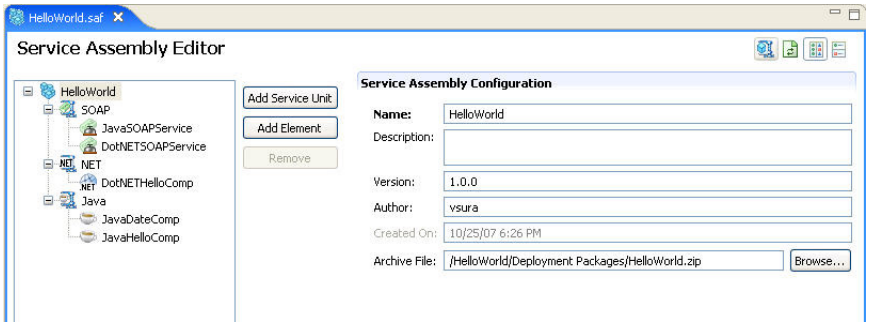
Figure 8 Debugger



Service Assembly Editor

In order to be deployed, composites must be transformed into service units and service assemblies. Figure 9 shows a sample service assembly editor.

Figure 9 Service Assembly Editor



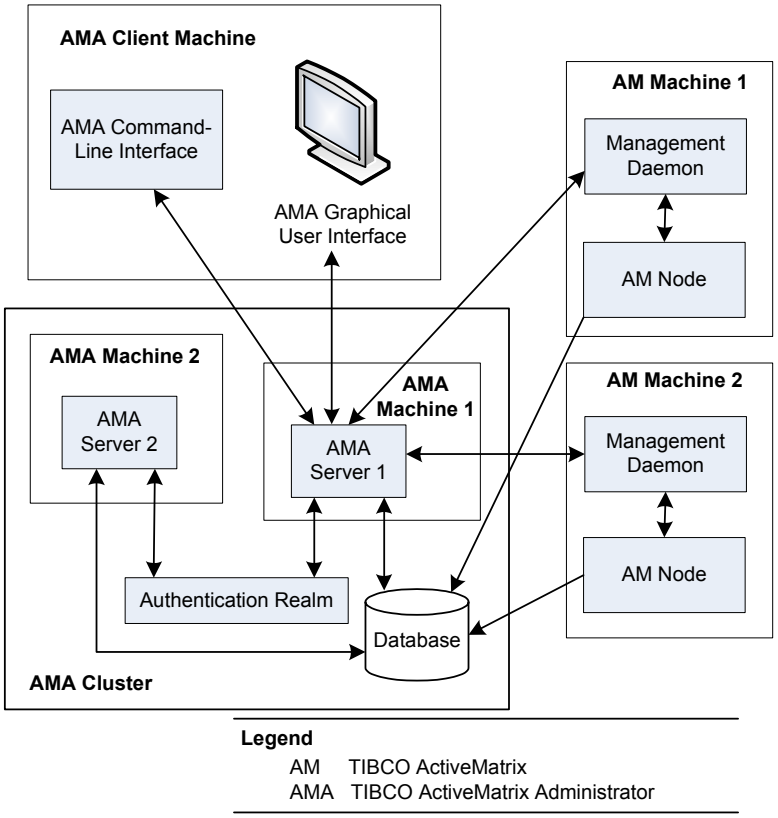
TIBCO ActiveMatrix Administrator

TIBCO ActiveMatrix Administrator can be used for enterprise, environment, and service management. The following sections provide an overview of TIBCO ActiveMatrix Administrator. To get a quick introduction to the use of the administration tools, see *TIBCO ActiveMatrix Service Grid Getting Started*. For detailed information about the administration tools, see *TIBCO Hawk Administrator's Guide*.

TIBCO ActiveMatrix Administrator Architecture

Figure 10 shows TIBCO ActiveMatrix Administrator components, and the relationship between TIBCO ActiveMatrix Administrator, other processes, and ActiveMatrix machines and nodes.

Figure 10 TIBCO ActiveMatrix Administrator



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

TIBCO ActiveMatrix Administrator Server Gathers management data from nodes, responds to requests from the ActiveMatrix Administrator graphical and command-line UIs, interacts with the authentication realm server to authenticate users, and interacts withTIBCO Management Daemon to manage nodes.

TIBCO ActiveMatrix Administrator Cluster Groups one or more ActiveMatrix Administrator servers together. ActiveMatrix Administrator servers within a cluster share a database and authentication realm and are kept synchronized.

ActiveMatrix Database Stores ActiveMatrix administration data.

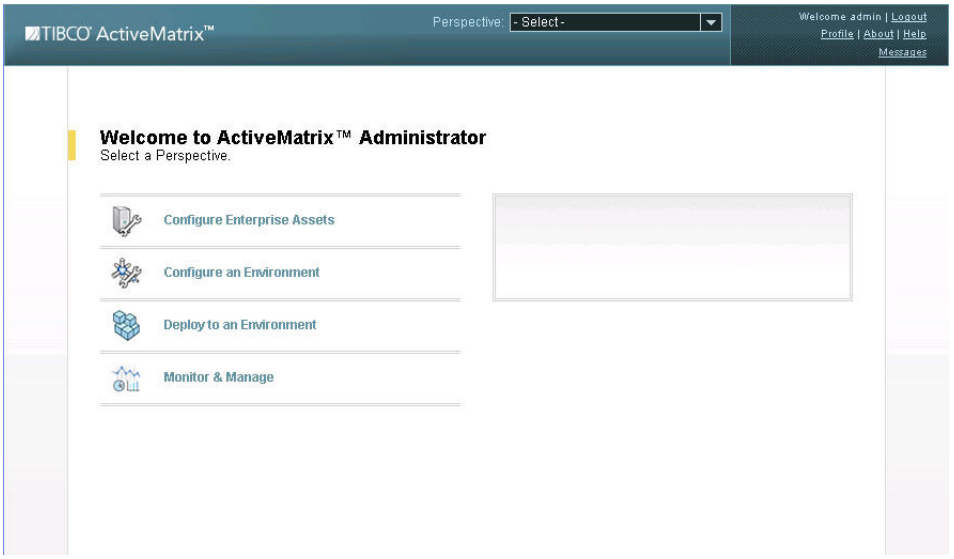
Authentication Realm Manages user authentication data. The authentication realm can be provided either by TIBCO Administrator or by another server or a file.

ActiveMatrix Administrator Graphical UI Displays the ActiveMatrix Administrator user interface. In ActiveMatrix Administrator, functionality is divided into perspectives. A *perspective* is a set of controls used to carry out a category of administration tasks. Figure 11 shows the ActiveMatrix Administrator graphical UI welcome page.

ActiveMatrix Administrator Command-Line Interface Provides a script-based interface to ActiveMatrix Administrator functions.

Management Daemon Gathers installation information and displays ActiveMatrix node life cycle operations.

Figure 11 TIBCO ActiveMatrix Administrator



Enterprise and Environment Administration

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

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

Service Administration

Service administration consists of deployment tasks and monitoring and management tasks. In the graphical interface, these tasks are carried out in the Deploy to an Environment and Monitor & Manage perspectives.

Service Deployment

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

The distribution of services across nodes is determined by the required level of service performance and availability. You can deploy a service unit across multiple nodes to improve service performance and availability. In this way, the Messaging Bus distributes requests among service instances.

Load Balanced Services

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

Highly Available Services

Services deployed on multiple containers are *highly available*. If one container fails, service requests will be handled by one of the other containers. Messaging Bus automatically routes to an available service instance identified in the message exchange.

Load Balanced Services

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

Service Monitoring and Management

TIBCO ActiveMatrix Administrator not only allows you to configure and deploy services, but also lets you monitor and manage the deployed services. Monitoring the system performance needs to be carried out on a day-to-day basis. The Monitor & Manage perspective of TIBCO ActiveMatrix Administrator keeps track of system health with low overhead.

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

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

Chapter 3 TIBCO ActiveMatrix Adapter for IBM i

This chapter explains the features and services of TIBCO ActiveMatrix Adapter for IBM i.

Unless explicitly stated otherwise, all features mentioned in this chapter are supported by both standalone adapter and adapter service engine.

Topics

Overview

TIBCO ActiveMatrix Adapter for IBM i allows one-way (publish or subscribe) or two-way (request-response) message exchange between TIBCO applications and applications running on an IBM System i machine.

The adapter interoperates with the IBM System i platform by:

- Invoking programs on the IBM System i machine
- Enabling applications running on the IBM System i machine to send outbound messages
- Writing messages to a data queue
- Retrieving spooled files list
- Converting a spooled file to PDF format

The adapter is compliant with the TIBCO ActiveEnterprise environment, and can be used in conjunction with adapters and products that are compliant with this environment, such as TIBCO ActiveMatrix BusinessWorks.

This book uses the terms *inbound* and *outbound* in the following way:

- Inbound means that data or messages are sent *to* the IBM System i machine.
- Outbound means that data or messages are sent *from* the IBM System i machine to outside applications.

Components

The adapter has two main components, the adapter palette and the runtime adapter.

- The adapter palette provides an adapter-specific GUI that seamlessly integrates with TIBCO Designer. It allows you to configure the adapter and stores the configuration in persistent storage. (TIBCO Designer is required to complete this task, and is installed as part of the TIBCO Runtime Agent installation. TIBCO Runtime Agent must be installed before installing the adapter).
- The runtime adapter uses this configuration to publish events through Publication Service, post data to TIBCO applications through Subscription Service, or allow exchange of data between TIBCO applications and applications on the IBM System i machine through Request-Response Service or Request-Response Invocation Service.

Adapter Features

The following features are supported by the TIBCO ActiveMatrix Adapter for IBM i.

Configuration Features

Easy-to-Use GUI

The adapter provides the adapter palette that seamlessly integrates with TIBCO Designer. This easy-to-use interface allows you to quickly configure adapter-specific features. You can use it to enter, delete, and modify configuration information.

Flexible Adapter Services

A single adapter instance can contain any number of Publication Service, Subscription Service, Request-Response Service, and Request-Response Invocation Service.

Mapping and Translation Support

The adapter performs mapping of functions and translation of data formats between external applications configured for the TIBCO environment and the corresponding application running on the IBM System i machine. You can use TIBCO ActiveMatrix BusinessWorks with the adapter when data mapping is required.

Robust and Reliable Data Transmission

The following features of the adapter make it a robust integration solution and provide reliable transmission of data through the network:

- **Connection Pool** The adapter has an extensive connection pooling mechanism, which provides an efficient way to manage connections and share them across different service requests. With connection pooling support, the adapter can allocate, recycle, reuse and release connections to the IBM System i machine.

  Connection pooling provides superior performance and reduces the number of idle connections.

If connections get invalidated or lost due to network outages or non-availability of the IBM System i machine, the connection pooling mechanism automatically re-creates new connections when the outage is over or when the system is available again. If a connection has been idle for a long time, the connection pooling mechanism closes it. The next time that access to that connection is required, the connection is automatically re-created.

- **Multi-threading** The adapter contains a simple and efficient multi-threading model which allows it to receive and post events and execute program calls concurrently. This multi-threading ability increases the performance of the adapter.

  You can configure the adapter to run in single threaded or multi-threaded mode. The number of threads can be increased or decreased for concurrent and simultaneous processing of both outbound and inbound messages. Inbound and Outbound dispatchers determine the number of threads the adapter will have at runtime. The number of Inbound dispatchers equals the total number of outbound or publication services configured in the adapter instance. Each outbound message or real-time event is processed in parallel in a separate thread.

- **Authentication** The adapter provides a simple authentication system of username and password verification for logging into the IBM System i machine. Authentication is configured at design-time when setting connection parameters.

Internationalization

The adapter provides internationalization support based on the Unicode support provided by the TIBCO Adapter SDK. Support is provided for string data only. Schema, metadata, error messages, and all other non-string information do not support extended character sets.

The adapter design time (palette) is fully localizable. The adapter design time component interface can display any language supported by TIBCO Designer.

Transport Features

TIBCO Rendezvous and TIBCO Enterprise Message Service can be used to transport messages to and from the adapter.

Tracing and Exception Handling Support

The adapter allows external applications to take advantage of the tracing and exception handling provided through TIBCO ActiveEnterprise.

Adapter Services

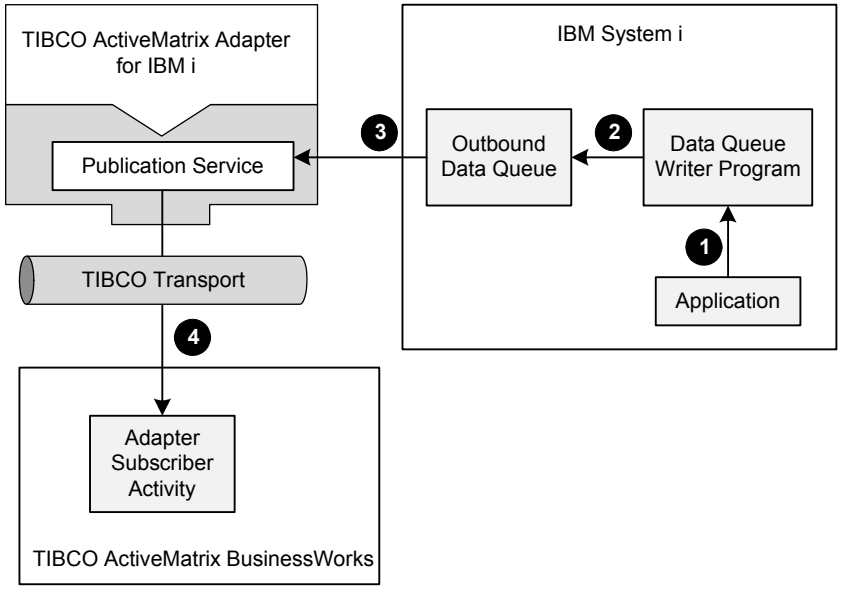
In TIBCO terminology, an adapter offers services to the host application and to the TIBCO environment. A service broadly encapsulates routing rules for messages handled by the service and also custom configuration information.

This section describes the services offered by the adapter.

Publication Service

Publication Service is initiated by an application running on an IBM System i machine. Publication Service recognizes when business events happen in an IBM System i application, and asynchronously sends out the event data in real-time to interested processes running in the TIBCO environment. Figure 12 shows the message flow.

Figure 12 Typical Message Flow of a Publication Service



In the preceding diagram:

1. An application running on IBM System i sets values of message data structure elements and invokes the data queue writer program.
2. The data queue writer program writes the message to outbound data queue.

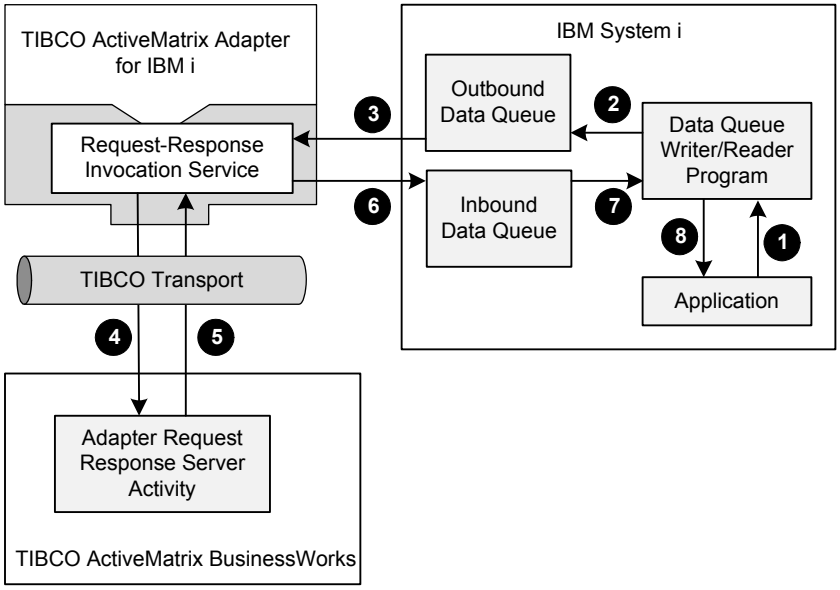
3. The adapter dispatcher component picks up added entry from the outbound data queue and Publication Service publishes the message using TIBCO transport.
4. A TIBCO ActiveMatrix BusinessWorks process that uses an adapter subscriber activity receives the message.

Request-Response Invocation Service

Request-Response Invocation Service is initiated by an application running on an IBM System i machine.

The System i application is the requester or initiator of the service. The adapter acts as a proxy. The System i application synchronously invokes functionality on an application running in the TIBCO environment. Figure 13 shows the message flow.

Figure 13 Typical Message Flow of a Request-Response Invocation Service



In the preceding diagram:

1. An application running on IBM System i sets the values of message data structure elements and invokes the data queue writer/reader program.
2. The data queue writer/reader program writes the message with key information to the outbound data queue. It then waits for a response entry with the same message key in the inbound data queue.

3. The adapter dispatcher component picks up the added entry from the outbound data queue. The Request-Response Invocation Service sends the message through TIBCO transport.
4. A TIBCO ActiveMatrix BusinessWorks process that uses an adapter request-response server activity receives the message.
5. The TIBCO ActiveMatrix BusinessWorks process for outbound messages responds. The adapter service for outbound messages receives the response from the process.
6. The adapter service writes the response to the inbound data queue with the message key. The data queue writer/reader program waiting for a reply receives the response from the inbound data queue.
7. The data queue writer/reader program sets the return parameter values and ends.
8. Control is transferred back to the calling application.

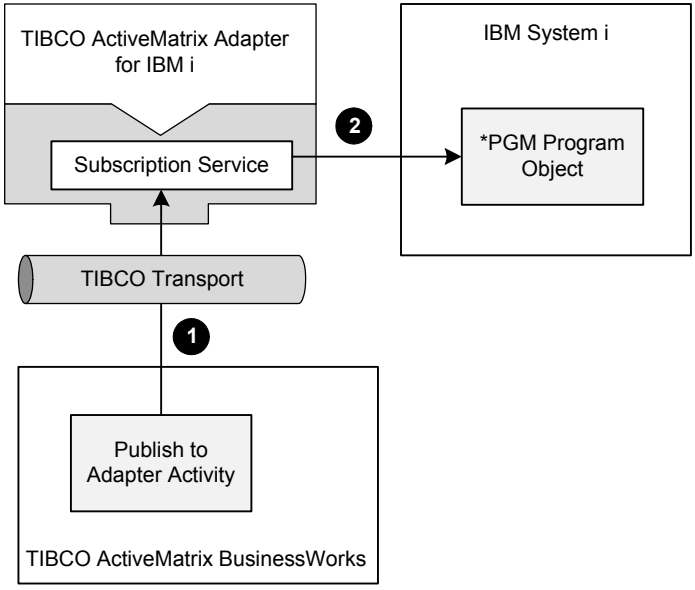
Subscription Service

Subscription Service is initiated by an application running in the TIBCO environment.

Program Call

Subscription Service listens for events occurring in the TIBCO environment and asynchronously performs an action on an System i application.

Figure 14 Typical Message Flow of a Subscription Service—Program Call



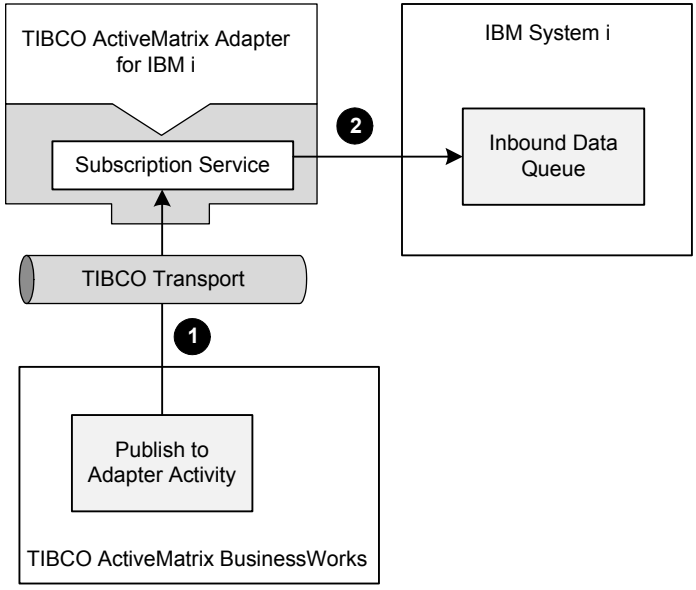
In the preceding diagram:

1. A process for invoking a program starts in TIBCO ActiveMatrix BusinessWorks.
2. Subscription Service receives the request from the process and asynchronously invokes an application running on an IBM System i machine.

Data Queue Write

Subscription Service for data queue write listens for events occurring in the TIBCO environment and asynchronously writes messages to a data queue.

Figure 15 Typical Message Flow of a Subscription Service—Data Queue Write



In the preceding diagram:

1. A process for writing to a data queue starts in TIBCO ActiveMatrix BusinessWorks.
2. Subscription Service receives the request from the process and asynchronously writes the message to a data queue on the IBM System i machine.

Request-Response Service

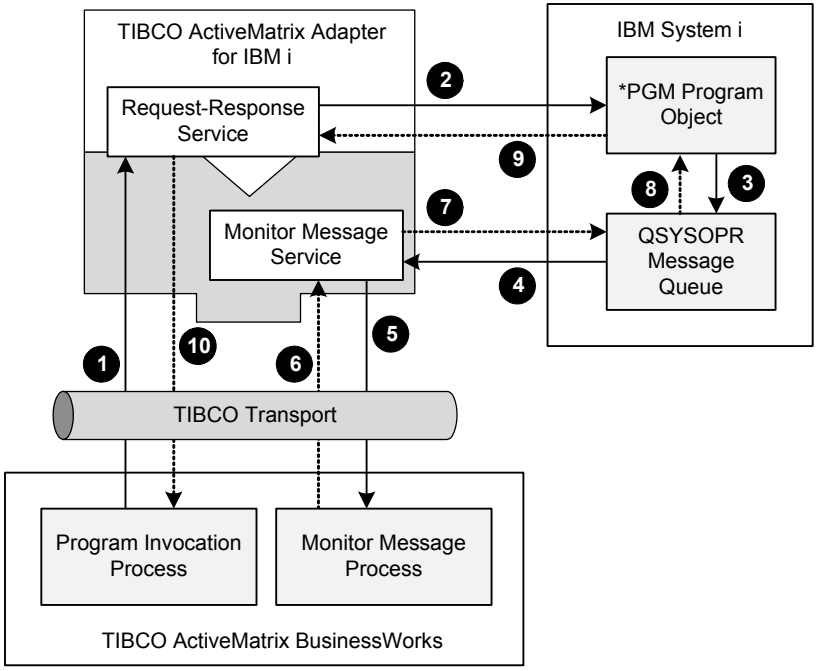
Request-Response Service is initiated by an application running in the TIBCO environment.

Program Call

The adapter can be used to synchronously retrieve data from or execute transactions within an IBM System i application for a TIBCO application.

After the action is performed in the IBM System i application, the adapter service sends a response back to the TIBCO environment 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 objects.

Figure 16 Typical Message Flow of a Request-Response Service—Program Call



In the preceding diagram:

1. A process for invoking a program starts in TIBCO ActiveMatrix BusinessWorks.
2. The adapter receives the request from the process and invokes a program on the IBM System i machine.
3. The program tries to add a record to a table that is full. An error message is generated by the operating system in the QSYSOPR message queue.
4. The adapter service for monitoring messages receives the error message from the QSYSOPR message queue.
5. The adapter publishes the error message through TIBCO transport. A process for monitoring the message in TIBCO ActiveMatrix BusinessWorks receives it and replies to the QSYSOPR message.
6. The adapter service for monitoring messages receives a response from the TIBCO ActiveMatrix BusinessWorks process.
7. The adapter service for monitoring messages replies to the QSYSOPR message.

8. The operating system receives the QSYSOPR message response and notifies program execution.
9. Program execution receives an error that is received by adapter program invocation service.
10. The adapter program invocation service sends the error back to the TIBCO ActiveMatrix BusinessWorks program invocation process.

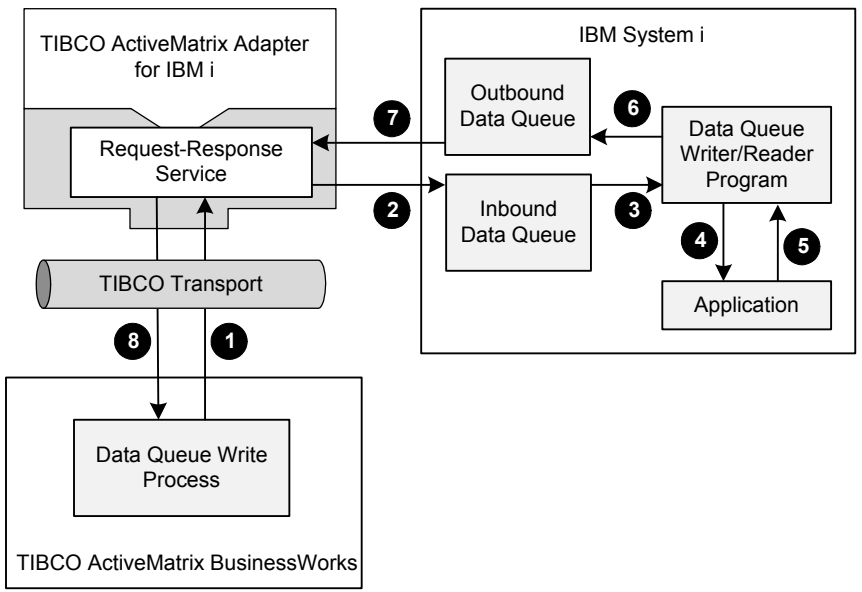
The step 3 through step 8 are not always performed.

For example, if a call to an IBM System i application is likely to generate a message in a message queue that needs a user reply (error or otherwise), the user should create a `Monitor Message` type Request-Response Invocation Service and associate it with a Request-Response Service or a Subscription Service. At runtime, if invocation of IBM System i application generates a message to a message queue, steps 3 through 8 will be performed.

Data Queue Write

The adapter can be used to synchronously write a message to a data queue and optionally read a reply from another data queue.

Figure 17 Typical Message Flow of a Request-Response Service—Data Queue Write



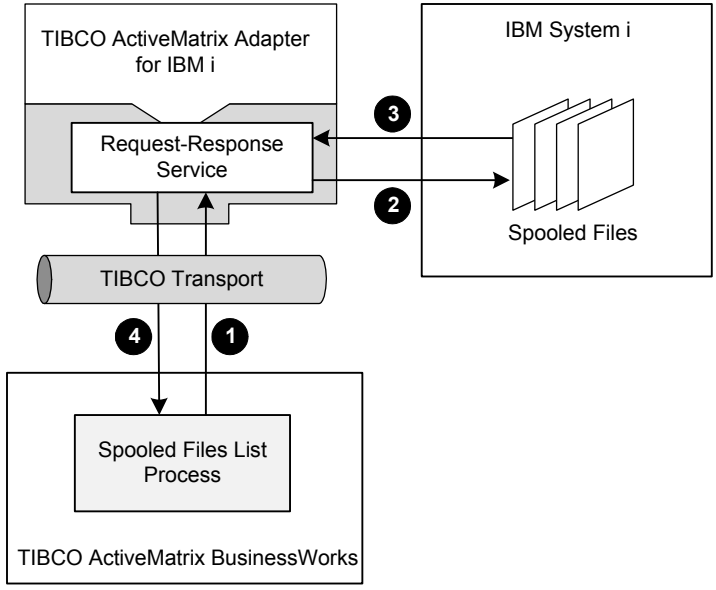
In the preceding diagram:

1. A process of Data Queue Write starts in TIBCO ActiveMatrix BusinessWorks.
2. The adapter receives the request from the process and writes a message to the inbound data queue on IBM System i machine.
3. The data queue reader/writer program that is invoked by an application running on IBM System i machine reads the message from the inbound data queue.
4. The data queue reader/writer program passes the message to the invoking application for processing.
5. The application that invoked the data queue reader/writer program processes the message and sends a reply.
6. The data queue reader/writer program receives the message and writes it to the outbound data queue.
7. The adapter request-response service waiting for a reply message in outbound data queue, picks up the reply message from outbound data queue.
8. The reply message is sent back to the TIBCO ActiveMatrix BusinessWorks process for further processing.

Spooled Files List

The adapter can retrieve a list of spooled files from IBM System i machine based on user-specified selection criteria.

Figure 18 Typical Message Flow of a Request-Response Service—Spooled Files List



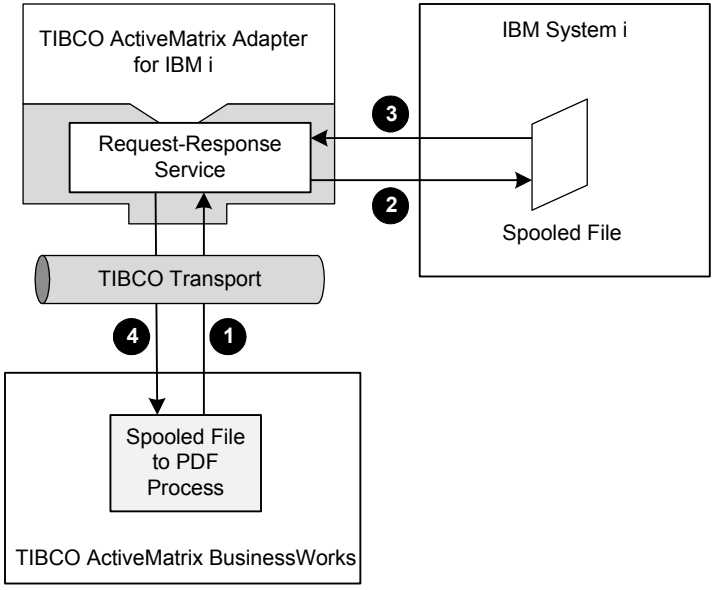
In the preceding diagram:

1. A process of Spooled Files List starts in TIBCO ActiveMatrix BusinessWorks.
2. The adapter receives a request from the process. The adapter sets the selection criteria for spooled files list and sends a request to the IBM System i machine.
3. A list of spooled files is returned from the IBM System i machine.
4. The adapter Request-Response Service sends the list of spooled files back to the TIBCO ActiveMatrix BusinessWorks process.

Request-Response Service — Spooled File to PDF

The adapter can be used to retrieve a spooled file from IBM System i machine and convert it to PDF format.

Figure 19 Typical Message Flow of a Request-Response Service—Spooled File to PDF



In the preceding diagram:

1. A process for retrieving a spooled file in PDF format starts in TIBCO ActiveMatrix BusinessWorks.
2. The adapter receives a request from the process. The adapter sends the request to IBM System i machine for selected spooled file.
3. The selected spooled file is returned from the IBM System i machine.
4. The Request-Response Service converts the spooled file into PDF format and sends the PDF file in binary format back to the TIBCO ActiveMatrix BusinessWorks process.

Chapter 4 Adapter and IBM i

This chapter introduces concepts and terms that are used in the IBM i environment, as well as explains the interaction between the adapter and IBM i.

Topics

IBM System i Overview

IBM System i platform is a midrange server offering by IBM. This machine has IBM proprietary operating system named IBM i. The database engine is integrated with operating system. The database is called DB2 UDB for iSeries or simply DB2/400.

Basic Concepts

Integrated File System

The file system used on System i (iSeries) is called Integrated File System (IFS). IFS comprises of many different sub-file systems.

Object and Library

The native file system used by iSeries is `QSYS.LIB`. Each entity in this file system is an *object*.

Each object has an object type that is signified by a character string starting with '*' character. For example, an executable program object is signified by a character string `*PGM` while a data queue object is signified by a character string `*DTAQ`.

There is a special type of object called a *library* (signified by `*LIB`). Library objects are similar to directories in Microsoft Windows and UNIX systems. A library groups object of other types together. However, unlike a directory, a library cannot have another library as its child.

Objects such as files (tables), programs, and other native System i type objects can exist only inside a library. An object in a library is addressed as *library_name/object_name*.

Job and Library List

Each unit of work that is processed by the IBM System i is called a job. A job is uniquely identified by a job number assigned by the system when the job is started.

Each job has a library list associated with it. This is similar to `PATH` variable in Windows or `LD_LIBRARY_PATH` variable in UNIX. When an object is not explicitly qualified with a library name, the object is searched in the library list for the job. The first occurrence of the object in library list is used. A library list is signified by string `*LIBL`.

Data Queue

Data queues are a type of object on the IBM System i. Applications can send messages to the data queues, and the messages wait in data queues until be retrieved by other applications.

Appendix A Encoding Tables

This appendix lists the encoding values that can be used for TIBCO ActiveMatrix Adapter for IBM i.

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.
- Encoding for Runtime—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 runtime 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.
- 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.

Table 4 lists the ISO character sets.

Table 4 ISO Character Sets

Value in Encoding Field Pick List	Encoding for Runtime	IBM CCSID	Oracle NLS String	Description
ASCII	ASCII	IBM-367	US7ASCII	7-bit ASCII
ISO-8859-1	ISO8859_1	IBM-819	language_territory.WE8ISO8859P1	ISO8859-1 (Latin-1), West European
ISO-8859-2	ISO8859_2	IBM-912	language_territory.EE8ISO8859P2	ISO8859-2 (Latin-2), East European
ISO-8859-3	ISO8859_3	IBM-913	language_territory.SE8ISO8859P3	ISO8859-3 (Latin-3), South European
ISO-8859-4	ISO8859_4	IBM-914	language_territory.NEE8ISO8859P4	ISO8859-4 (Latin-4), North European
ISO-8859-5	ISO8859_5	IBM-915	language_territory.CL8ISO8859P5	ISO8859-5, Cyrillic

Table 4 ISO Character Sets (Cont'd)

Value in Encoding Field Pick List	Encoding for Runtime	IBM CCSID	Oracle NLS String	Description
ISO-8859-6	ISO8859_6	IBM-1089	ARABIC_UNITED ARAB EMIRATES.AR8ISO8859P6	ISO8859-6, Arabic
ISO-8859-7	ISO8859_7	IBM-813	GREEK_GREECE.EL8ISO8859P7	ISO8859-7, Greece
ISO-8859-8	ISO8859_8	IBM-916	HEBREW_ISRAEL.IW8ISO8859P8	ISO8859-8, Hebrew
ISO-8859-9	ISO8859_9	IBM-920	TURKISH_TURKEY.WE8ISO8859P9	ISO8859-9 (Latin-5), Turkey
ISO-8859-13	ISO8859_13	IBM-921	N/A	ISO8859-13, Baltic
ISO-8859-15	ISO8859_15_FDIS	IBM-923	N/A	ISO8859-15 (Latin-9), Latin-1 with Euro
Shift_JIS (CP943)	CP943	IBM-943	JAPANESE_JAPAN.JA16SJIS	Japanese Shift-JIS, CP943
KSC_5601	KSC5601	IBM-949	KOREAN_KOREA.KO16KSC5601	Korean KSC-5601
Big5	Big5	IBM-1370	TRADITIONAL CHINESE_TAIWAN.ZHT16BIG5	Traditional Chinese Big5 (with Euro Sign)
GBK	GBK	IBM-1386	SIMPLIFIED CHINESE_CHINA.ZHS16CGB231280	Simplified Chinese GBK, super set of GB2312-80
EUC-JP	EUC_JP	IBM-954	JAPANESE_JAPAN.JA16EUC	Japanese EUC
EUC-KR	EUC_KR	IBM-970	KOREAN_KOREA.KO16KSC5601	Korean EUC
EUC-CN	EUC_CN	IBM-1383	SIMPLIFIED CHINESE_CHINA.ZHS16CGB231280	Simplified Chinese EUC, compatible with GB2312-80

Table 4 ISO Character Sets (Cont'd)

Value in Encoding Field Pick List	Encoding for Runtime	IBM CCSID	Oracle NLS String	Description
Shift_JIS (TIBCO)	Shift_JIS	N/A	JAPANESE_JAPAN.JA16SJIS	Variant of IBM-943, flavoring some MS-932 conversions
Shift_JIS(932)	MS932	IBM-932	JAPANESE_JAPAN.JA16SJIS	Shift_JIS encoding in MS-932 flavour
UTF8	UTF8	IBM-1208	AMERICAN_AMERICA.UTF8	Unicode Transformation Format-8
UTF16_BigEndian	UnicodeBig	N/A	N/A	UTF16 (UCS-2) Big Endian
UTF16_LittleEndian	UnicodeLittle	N/A	N/A	UTF16 (UCS-2) Little Endian

Table 5 lists the EBCDIC character sets.

Table 5 EBCDIC Character Sets

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

Table 5 EBCDIC Character Sets (Cont'd)

Encoding Value	Encoding for Runtime	IBM CCSID	Oracle NLS String	Description
IBM-500	CP500	IBM-500	N/A	EBCDIC Latin-1 (International)
IBM-1148	CP1148	IBM-1148	N/A	EBCDIC Latin-1 (with Euro Sign)
IBM-1047	N/A	IBM-1047	N/A	EBCDIC Open System Latin-1
IBM-1153	N/A	IBM-1153	N/A	EBCDIC Latin-2 (with Euro Sign)
IBM-1154	N/A	IBM-1154	N/A	EBCDIC Cyrillic
IBM-1156	N/A	IBM-1156	N/A	EBCDIC Baltic
IBM-1157	N/A	IBM-1157	N/A	EBCDIC Estonia
IBM-290	N/A[1]	IBM-290	N/A	EBCDIC Japanese Katakana (SBCS).
IBM-1159	N/A[2]	IBM-1159	N/A	EBCDIC Traditional Chinese (SBCS)
IBM-300	N/A[3]	IBM-300	N/A	EBCDIC Japanese (DBCS)
IBM-834	N/A[4]	BM-834	N/A	EBCDIC Korean (DBCS)
IBM-835	N/A[5]	IBM-835	N/A	EBCDIC Traditional Chinese (DBCS)
IBM-837	N/A	IBM-837	N/A	Forward compatible with IBM-935
IBM-930	CP930	IBM-930	N/A	EBCDIC Japanese (MBCS)
IBM-933	CP933	IBM-933	N/A	EBCDIC Korean (MBCS)
IBM-935	CP935	IBM-935	N/A	EBCDIC Simplified Chinese (MBCS)

Table 5 EBCDIC Character Sets (Cont'd)

Encoding Value	Encoding for Runtime	IBM CCSID	Oracle NLS String	Description
IBM-937	CP937	IBM-937	N/A	EBCDIC Traditional Chinese (MBCS)
IBM-1390	N/A	IBM-1390	N/A	EBCDIC Japanese (MBCS) with Euro
IBM-1364	N/A	IBM-1364	N/A	EBCDIC Korean (MBCS) with Euro
IBM-1371	N/A	IBM-1371	N/A	EBCDIC Traditional Chinese (MBCS) with Euro

1. Forward compatible with CP930
2. Forward compatible with CP937
3. Forward compatible with CP930
4. Forward compatible with CP933
5. Forward compatible with CP937

Table 6 lists the Microsoft Windows character sets.

Table 6 Microsoft Windows Character Sets

Encoding Value	Encoding for Runtime	IBM CCSID	Oracle NLS Encoding	Description
Windows Latin-1	CP1252	IBM-1252	language_territory.WE8MSWIN1252	Windows Latin-1 with Euro Sign
Windows Latin-2	CP1250	IBM-1250	language_territory.EE8MSWIN1250	Windows Latin-2 with Euro Sign
Windows Cyrillic	CP1251	IBM-1251	language_territory.CL8MSWIN1251	Windows Cyrillic with Euro Sign
Windows Greek	CP1253	IBM-1253	GREEK_GREECE.EL8MSWIN1253	Windows Greek with Euro Sign
Windows Turkish	CP1254	IBM-1254	TURKISH_TURKEY.TR8MSWIN1254	Windows Turkish with Euro Sign

Table 6 Microsoft Windows Character Sets (Cont'd)

Encoding Value	Encoding for Runtime	IBM CCSID	Oracle NLS Encoding	Description
Windows Hebrew	CP1255	IBM-1255	HEBREW_ISRAEL.IW8MSWIN1255	Windows Hebrew with Euro Sign
Windows Arabic	CP1256	IBM-1256	ARABIC_UNITED ARAB EMIRATES.AR8MSWIN1256	Windows Arabic with Euro Sign
Windows Baltic	CP1257	IBM-1257	language_territory.BLT8MSWIN1257	Windows Baltic with Euro Sign

Index

L

O

P

R

S

T

U

V

W