TIBCO ActiveMatrix® Adapter for Database

Concepts

Software Release 6.0
April 2009

Important Information

Contents

Figures . . . . . **v**

Tables . . . . . **vii**

Preface . . . . . **ix**

Related Documents x
TIBCO Product Documentation x
Other TIBCO Product Documentation x
Typographical Conventions xii
How to Contact TIBCO Customer Support xiv

Chapter 1 Introduction. **1**

What is an Adapter? 2
Adapter Components 3
Adapter Key terms 4
Adapter Services 5
Publication Service 5
Subscription Service 5
Request-Response Service 5
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.Request-Response Invocation Service 5
Adapter Services Summary 6
Choosing an Adapter Service 7

Chapter 2 TIBCO ActiveMatrix Adapter for Database. **9**

Overview 10
Adapter Services 11
Publication and Subscription Services 11
Request-Response Service 12
Adapter Architecture 14
Run-time Adapter 15
Polling or Alerter 15
Referencing External Schemas 16
Exception Table 16
Master-Master Replication 17
Multi-file Format Projects 17

DAT File Format . . . 18
Fault Tolerance . . . 18
Supported SQL Operations . . . 19

Chapter 3 Adapter Infrastructure Tools . . . 21

TIBCO Runtime Agent . . . 22
TIBCO Domain Utility . . . 22
TIBCO Designer . . . 23
TIBCO Administrator . . . 24
TIBCO Administration Domain . . . 24
TIBCO Administration Server . . . 25
TIBCO Administrator GUI . . . 25
TIBCO ActiveMatrix BusinessWorks . . . 27
TIBCO Hawk . . . 28
Adapter Microagents . . . 28
TIBCO Business Studio . . . 30
ActiveMatrix Resource Wizard . . . 31
Composite Element Editors . . . 31
Debugger . . . 33
Service Assembly Editor . . . 34
TIBCO ActiveMatrix Administrator . . . 35
TIBCO ActiveMatrix Administrator Architecture . . . 35
Enterprise and Environment Administration . . . 37
Service Administration . . . 37

Figures

Figure 1 Typical Publication-Subscription Services Flow 12
Figure 2 Typical Request-Response Service Flow. 13
Figure 3 The TIBCO ActiveMatrix Adapter for Database Environment 14
Figure 4 TIBCO Designer main window 23
Figure 5 TIBCO Administrator GUI. 26
Figure 6 Component Property Sheet 31
Figure 7 Debugger 33
Figure 8 Service Assembly Editor 34
Figure 9 TIBCO ActiveMatrix Administration Architecture 35
Figure 10 TIBCO ActiveMatrix Administrator 36

Tables

Table 1 General Typographical Conventions xii
Table 2 Adapter Services Summary 6

Preface

This document describes TIBCO ActiveMatrix Adapter for Database.

Topics

- *Related Documents, page x*
- *How to Contact TIBCO Customer Support, page xiv*
- *How to Contact TIBCO Customer Support, page xiv*

Related Documents

This section lists documentation resources.

TIBCO Product Documentation

The following documents form the TIBCO ActiveMatrix Adapter for Database Documentation set:

TIBCO ActiveMatrix Adapter for Database

- *TIBCO ActiveMatrix Adapter for Database Concepts* — Read this manual to gain an understanding of adapters in general that you can apply to the various tasks you may undertake.
- *TIBCO ActiveMatrix Adapter for Database Installation*— Read this manual to learn how to install TIBCO ActiveMatrix Adapter for Database.
- *TIBCO ActiveMatrix Adapter for Database Configuration and Deployment*— This manual explains how to create and configure adapter projects. Information on deploying adapter projects is also included.
- *TIBCO ActiveMatrix Adapter for Database Examples*— Read this manual to work through the examples provided with the adapter.
- *TIBCO ActiveMatrix Adapter for Database Release Notes*— Read this document for information about new features, deprecated features, and open and closed issues.

TIBCO ActiveMatrix Adapter Service Engine for Database

- *TIBCO ActiveMatrix Adapter Service Engine for Database Installation* Read this manual to learn how to install TIBCO ActiveMatrix Adapter for Database.
- *TIBCO ActiveMatrix Adapter Service Engine for Database Configuration and Deployment* This manual explains how to create and configure adapter projects. Information on deploying adapter projects is also included.
- *TIBCO ActiveMatrix Adapter Service Engine for Database Release Notes* Read this document for information about new features, deprecated features, and open and closed issues.

Other TIBCO Product Documentation

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

- TIBCO ActiveMatrix BusinessWorks Service Engine
- TIBCO Runtime Agent™

- TIBCO ActiveMatrix® Service Grid
- TIBCO ActiveMatrix® Service Bus
- TIBCO Business Studio™

Typographical Conventions

The following typographical conventions are used in this manual

Table 1 General Typographical Conventions

Convention	Use
TIBCO_HOME *ENV_HOME* *AMX_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`.
`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]
italic font	Italic font is used in the following ways: • To indicate a document title. For example: See *TIBCO 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*

Table 1 General Typographical Conventions (Cont'd)

Convention	Use
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.

How to Contact TIBCO Customer Support

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

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

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

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

  https://support.tibco.com

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

Chapter 1 Introduction

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

Topics

- *What is an Adapter?, page 2*
- *Adapter Components, page 3*
- *Adapter Key terms, page 4*
- *Adapter Services, page 5*
- *Choosing an Adapter Service, page 7*

What is an Adapter?

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

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

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

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

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

Adapter Components

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

The adapter components are:

- Standalone

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

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

- Adapter Service Engine

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

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

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

  Existing standalone adapter configurations can also be deployed as services.

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

Adapter Key terms

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

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

Adapter Services

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

Publication Service

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

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

Subscription Service

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

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

Request-Response Service

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

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.Request-Response Invocation Service

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

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

Adapter Services Summary

The next table summarizes the services introduced in this section.

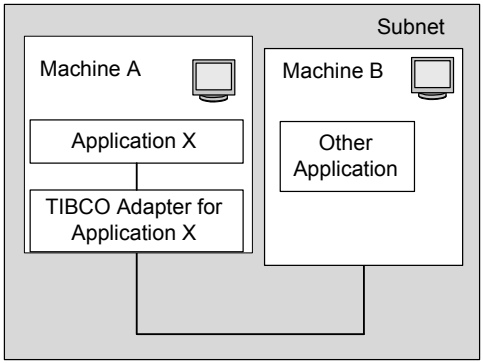
Table 2 Adapter Services Summary

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:

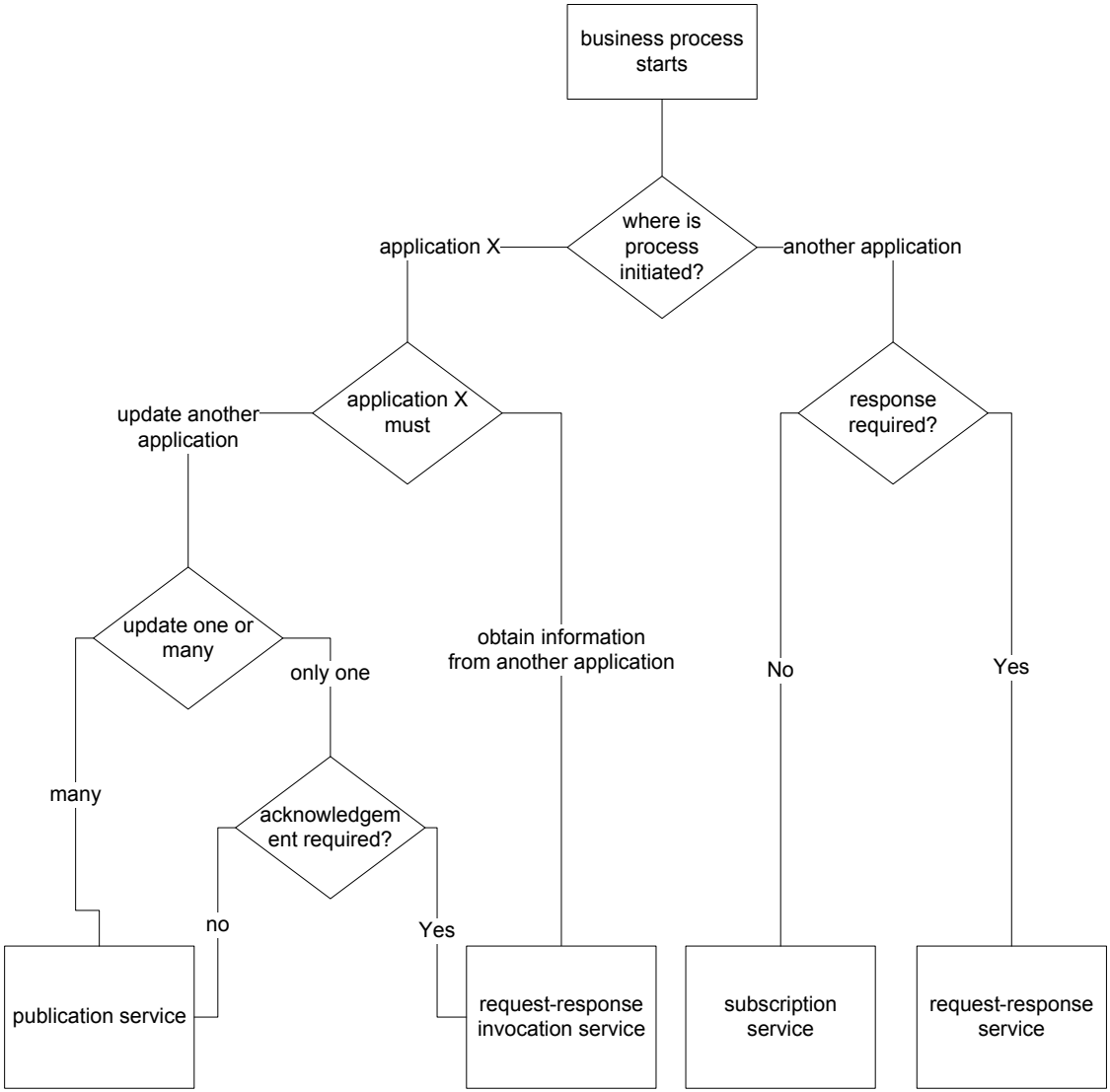


In this scenario, data must be exchanged between the application X and another application. The other application could be a customer management system, such as PeopleSoft, or another TIBCO application, such as TIBCO ActiveMatrix BusinessWorks.

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.

Chapter 2 TIBCO ActiveMatrix Adapter for Database

Topics

- *Overview, page 10*
- *Adapter Services, page 11*
- *Adapter Architecture, page 14*
- *Supported SQL Operations, page 19*

Overview

TIBCO ActiveMatrix Adapter for Database software (the adapter) allows data changes in a database to be sent as they occur to other databases and applications. It extends publish-subscribe and request-response technology to databases, making multiple levels of delivery services available to applications that need access to these databases. ODBC and JDBC compliant databases such as Oracle, Sybase, and Microsoft SQL Server are supported. While the adapter does not run on z/OS and iSeries systems, it can remotely connect to a DB2 database running on these systems.

Adapter Services

The adapter offers publication, subscription, and request-response services. Communication parameters, database connectivity parameters, and many other parameters can be configured for these adapter services using TIBCO Designer.

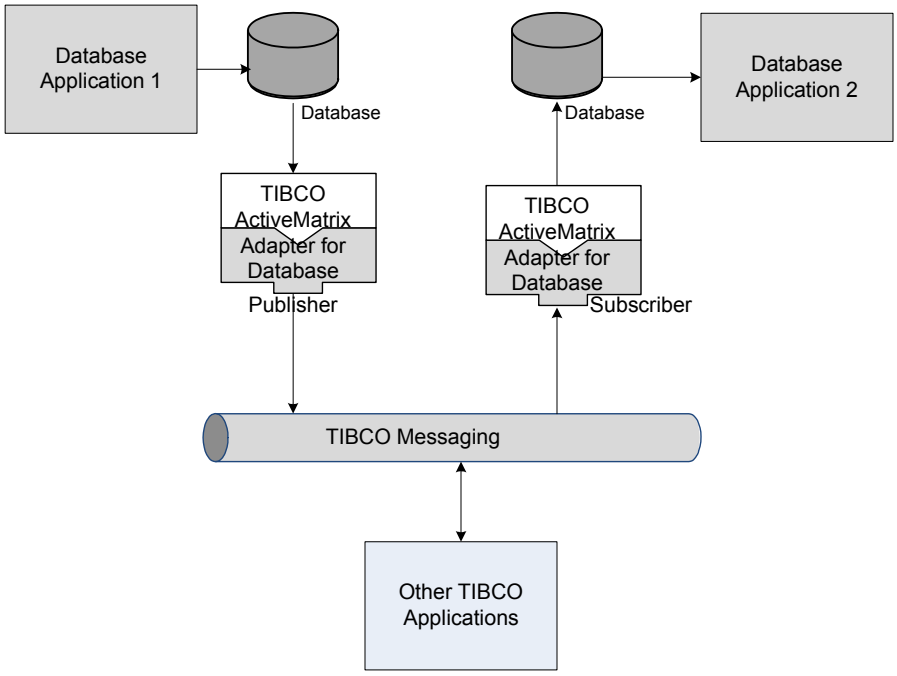
The following sections describe how these adapter services interact with the database environment to provide message connectivity with it.

Publication and Subscription Services

The following diagram illustrates publication and subscription. Database Application 1 updates a table in a database monitored by a TIBCO ActiveMatrix Adapter for Database publication service. The adapter extracts data from the changed rows of the database tables and publishes it on the specified transport (TIBCO Rendezvous or TIBCO Enterprise Message Service). This data is then available to other applications listening on these transports.

One particular type of listening application could be another instance of a TIBCO ActiveMatrix Adapter for Database adapter, running a subscription service. Upon receiving a message, it updates the relevant tables in its associated database. The following figure shows data flow when using the standalone adapter.

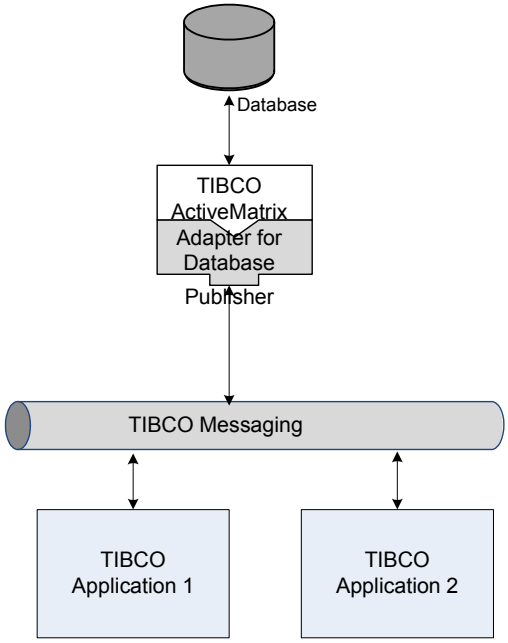
Figure 1 Typical Publication-Subscription Services Flow



Request-Response Service

The request-response feature in TIBCO ActiveMatrix Adapter for Database allows an application to submit one or more SQL statements, stored procedures, or both for the adapter to execute. After execution, the adapter returns one or more result sets and a result code. The request and responses are formulated using nested self-describing messages. This data flow is illustrated in the figure below.

Figure 2 Typical Request-Response Service Flow



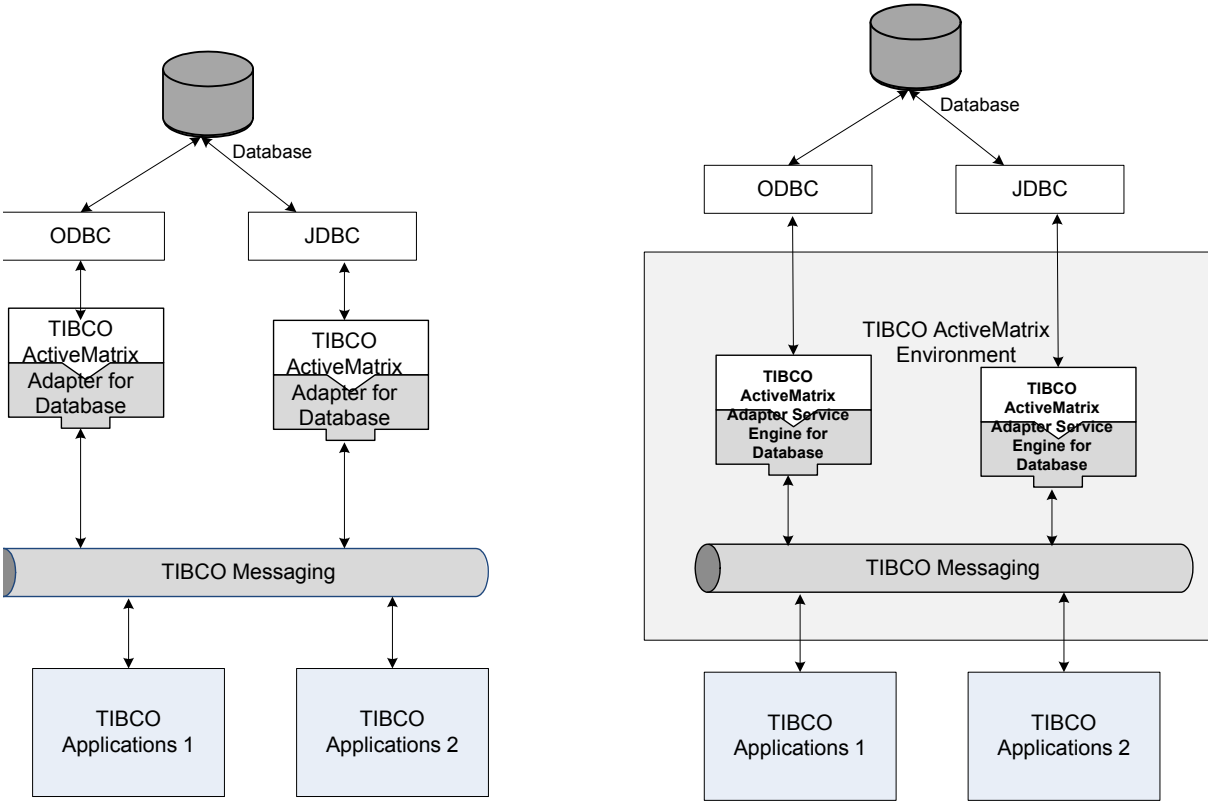
The adapter can also be configured to manage requests using load balancing among a set of adapters that participate in the same named queue. Additionally, the adapter has been designed to allow multiple threads to process application requests.

Adapter Architecture

TIBCO ActiveMatrix Adapter for Database software allows data stored in a database to be exchanged with applications configured for the TIBCO environment. For Oracle databases, the adapter includes an alerter component that can be used when the default method of polling for database changes is not appropriate. An alerter is also available for Microsoft SQL Server and Sybase and is run as a separate executable.

The following diagram shows the architecture of the standalone adapter and the adapter service engine. Configuration data is stored in projects that can be used by one or more adapters. The components are explained in this section and the following section.

Figure 3 The TIBCO ActiveMatrix Adapter for Database Environment



Run-time Adapter

A run-time adapter service acts as a bi-directional bridge between your database and the TIBCO environment. A publication service monitors your database, extracting data from relevant database tables and sending it on an appropriate subject using the configured quality of service, reliable or certified. A subscription service listens for messages sent on configured subjects and inserts, deletes or updates rows in its associated database tables with the information.

An adapter can be configured with both a publication and subscription service, sending messages and writing them to more than one table. An adapter can publish and subscribe to the same table, provided the publish destination is different from the subscribe destination.

An adapter can also be configured to use request-response semantics, which allows the adapter to act on behalf of a TIBCO application, processing its requests and returning responses. A set of adapters can be configured to participate in a queue (TIBCO Rendezvous Distributed Queue or JMS Queue) that allows load balancing, so requests are handled promptly.

Adapter services using TIBCO Rendezvous transport can use the TIBCO Rendezvous certified messaging service. Message parameters are stored in the project and can be modified for your environment. Adapters using JMS can use the durable messaging service.

Polling or Alerter

An adapter publication service can use periodic polling or an alerter process to monitor changes to a database table.

- The default method is polling. The adapter polls (checks) 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.
- In the case where the publishing tables change infrequently, polling may result in many unnecessary database accesses. For this situation the alerter can be used to asynchronously alert the adapter of changes in the database. This removes the need for the adapter to poll its publishing table for existence of new rows. It is recommended to use the alerter only when database changes are infrequent. When the adapter receives an alert, it publishes the new data.

Referencing External Schemas

When configuring an adapter, you can specify the database tables to use for each publication and subscription service. The default schema for specified source and destination table names is associated with the database user account in the `create_user.sql` script. This script is normally executed as part of the post installation procedure.

You may need to access source and destination tables that are in another schema, and not in your default schema (the schema specified in `create_user.sql`). Source and destination tables in the adapter can be configured to reference tables in an external schema.

Exception Table

You can designate an exception table for a subscription service. If a subscriber adapter cannot write data to its destination table, it will write the data to the exception table.

To write to an exception table, an adapter must be started with the `adb.useExceptTable` option set to on in the supplied properties file and an exception table must have been specified when using TIBCO Designer to add a subscription. If an error occurs when inserting data into the destination table, the adapter attempts to insert the data into the exception table.

- If the insertion into the exception table is successful and the RVCM quality of service is specified for the subscription, the transaction commits and confirmation for the message is sent back. In the publishing table, the value of the delivery status column is `C`, since the message was processed. If you are using a durable JMS subscriber, a JMS confirmation will be sent back.
- If the insertion into the exception table fails, an error message displays, a rollback occurs and a TIBCO Hawk alert is published. The adapter will then terminate. The user is advised to fix the problem with the message or the table and then to restart the adapter. The message will be delivered again after the adapter starts. (If the adapter did not terminate, after the next message was inserted successfully, its confirmation would also implicitly confirm the failed message.)

You can identify the operation that failed by checking the ADB_OPCODE column of the publishing table. The ADB_ERROR_TEXT column contains error information. Loop Detection

If a source table is used both as the source and destination table on the same subject, the loop detection feature should be enabled. The feature prevents the same changes from being published repeatedly.

If the loop detection feature is enabled, an additional column, ADB_SOURCE, is added to the source table. When an adapter receives a message, it inserts or updates the source table and enters a `T` in the ADB_SOURCE column to denote that this row was inserted or updated as a result of a message, rather than from user intervention.

Triggers created by the Adapter Configuration palette are defined to not copy rows with `T` in the ADB_SOURCE column into the publishing table, which effectively means that the row will not be published. If you must update a row that was received (that is, has a `T` in the ADB_SOURCE column) *and* want the updated row propagated, you must also change the ADB_SOURCE column to NULL. This causes the trigger to pick up the row and send it out.

Master-Master Replication

You can use the loop detection feature to implement a simple master-master replication scheme. Master-master replication allows multiple sites, acting as peers, to copy and maintain groups of replicated objects.

When loop detection is enabled, an adapter can be configured as both a publisher and a subscriber to the same table on the same subject. When the subscriber receives a message, it compares the `adapter ID` of the message to its own `adapter ID`. If `adapter ID` values match and source and destination tables are the same table in the database, the subscriber discards the message. Other subscriber adapters listening on the subject receive the message only once.

The adapter does not resolve any replication conflicts, such as when two applications each update the same row of their corresponding source table and both publish the change. For advanced replication scenarios, use the replication tools provided by your database vendor.

Multi-file Format Projects

The multi-file format creates one ActiveEnterprise XML file for each logical object (such as an adapter instance, a set of related ActiveEnterprise classes, or a TIBCO ActiveMatrix BusinessWorks process flow) that occurs in the repository instance. This kind of project is referred to as a multi-file project.

Multi-file projects can be checked into a version control system, and a project can contain more than one adapter configuration. This allows a number of people to work on the same project at the same time, with different people working on each adapter configuration: a developer can check out the specific file corresponding to an object that needs to be changed, updated the file, and check it back in. TIBCO Designer accesses the local synchronized copies of the files on the developer's hard drive.

For more information see *TIBCO ActiveMatrix Adapter for Database Configuration and Deployment*.

DAT File Format

For production and for testing with run-time adapters, developers export multi-file projects to `.dat` format using TIBCO Designer menu commands. A project can be exported as a local project for local testing or exported to be managed by a repository server.

Fault Tolerance

Within the context of the adapter, a primary instance is the adapter instance that processes messages between the TIBCO environment and the database. The secondary instance uses the same TIBCO Designer project but runs in a stand-by mode and takes over when the primary instance goes down. The secondary instance(s) need not run on the machine or platform as the primary instance.

Supported SQL Operations

The adapter supports the following SQL operations for publishing and subscribing:

- INSERT
- UPDATE
- DELETE
- UPDATE/INSERT (update if row exists, otherwise insert)

Whenever a supported SQL statement is executed against a table monitored by a publisher adapter, the adapter instance sends a message to its subscribers, which update their destination tables.

Chapter 3 Adapter Infrastructure Tools

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

Topics

- *TIBCO Runtime Agent, page 22*
- *TIBCO Designer, page 23*
- *TIBCO Administrator, page 24*
- *TIBCO ActiveMatrix BusinessWorks, page 27*
- *TIBCO Hawk, page 28*
- *TIBCO Business Studio, page 30*
- *TIBCO ActiveMatrix Administrator, page 35*

TIBCO Runtime Agent

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

The TRA has two main functions:

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

TIBCO Domain Utility

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

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

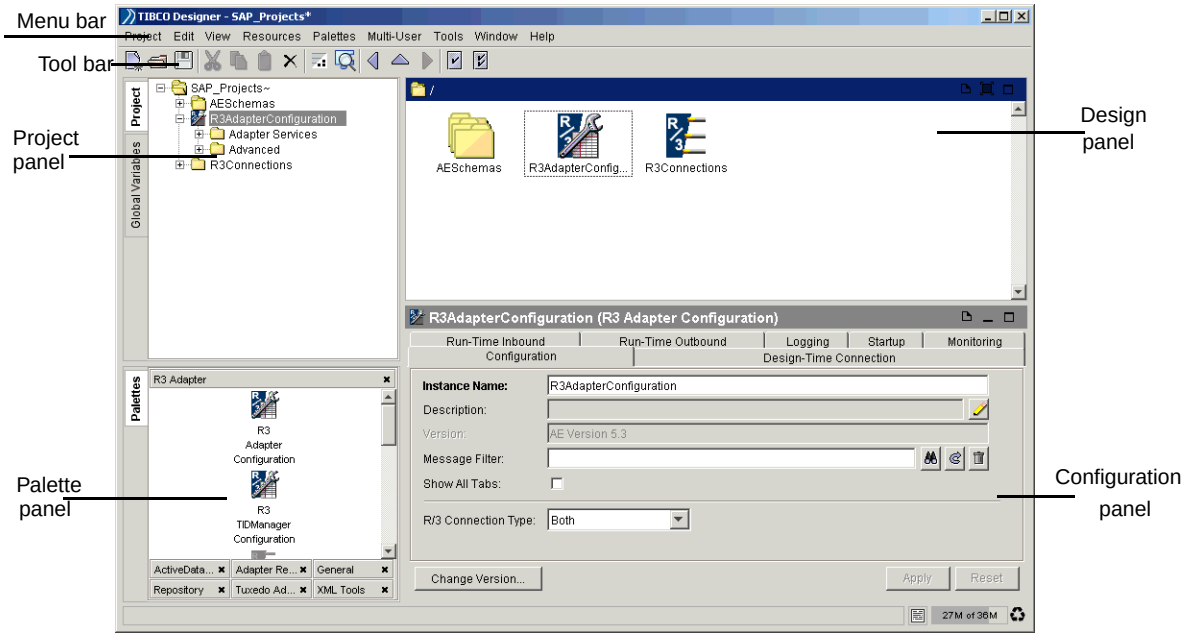
TIBCO Designer

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

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

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

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

  To get the load balancing benefit with HTTP, you must either use an IP redirector or explicitly point to a backup server to be used when a server fails. See your IP Redirector or HTTP Server documentation for information on how to do this.
- Failure recovery. You can use a load-balanced TIBCO Administration Server for failure recovery. In a completely trusted environment, you can instead use a database back-end for your server and use checkpoints in the database for failure recovery.

TIBCO Administration Domain

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

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

Each TIBCO administration domain contains one or more machines. By default, all machines within an administration domain are expected to be in the same network subnet. You can, however, set up your system to use TIBCO Rendezvous 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.

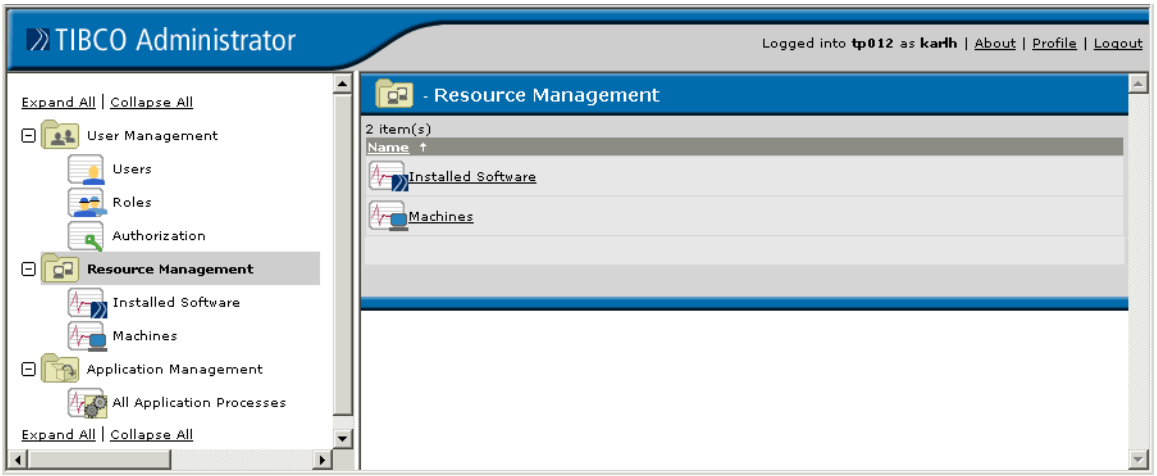
The TIBCO Administrator Server contains its own web server (Apache Tomcat) that can be accessed via the TIBCO Administrator GUI for configuration and monitoring information.

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.

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

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

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

See the TIBCO ActiveMatrix BusinessWorks documentation for more information.

TIBCO ActiveMatrix BusinessWorks Service Engine

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

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

TIBCO Hawk

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

TIBCO Hawk features include:

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

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

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

Adapter Microagents

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

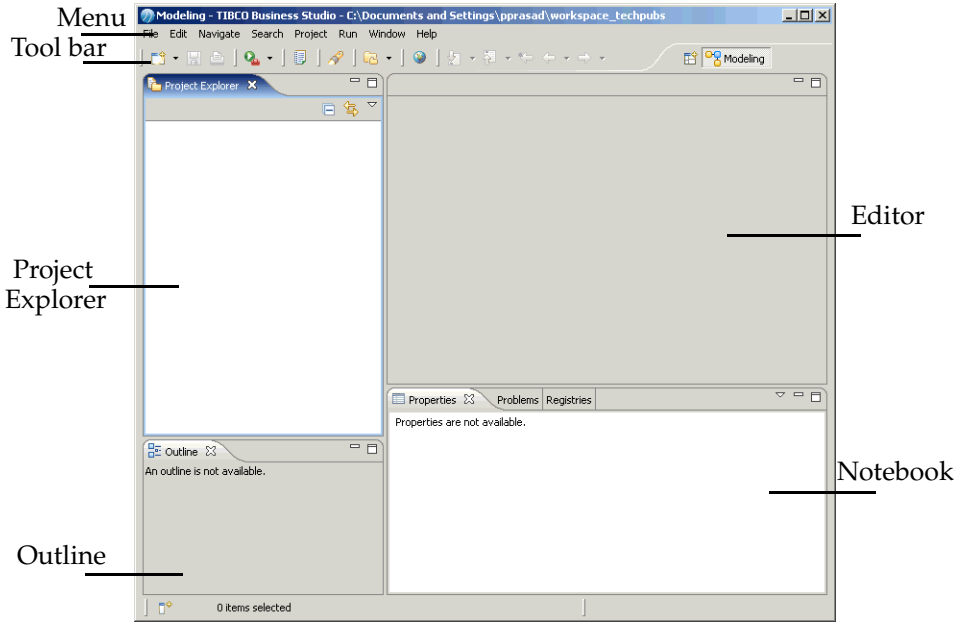
- Business level statistics—statistics that report the progress of the adapter as it interacts with the vendor application. For example, in a database adapter such

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

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

TIBCO Business Studio

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



The screen contains the following area and views:

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

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

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

- **Views** Displays under the Editor Area. Contains the following views:

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

  — **Problems** Displays validation and other errors.

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

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

ActiveMatrix Resource Wizard

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

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

Composite Element Editors

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

Figure 6 shows the an example of the Properties view.

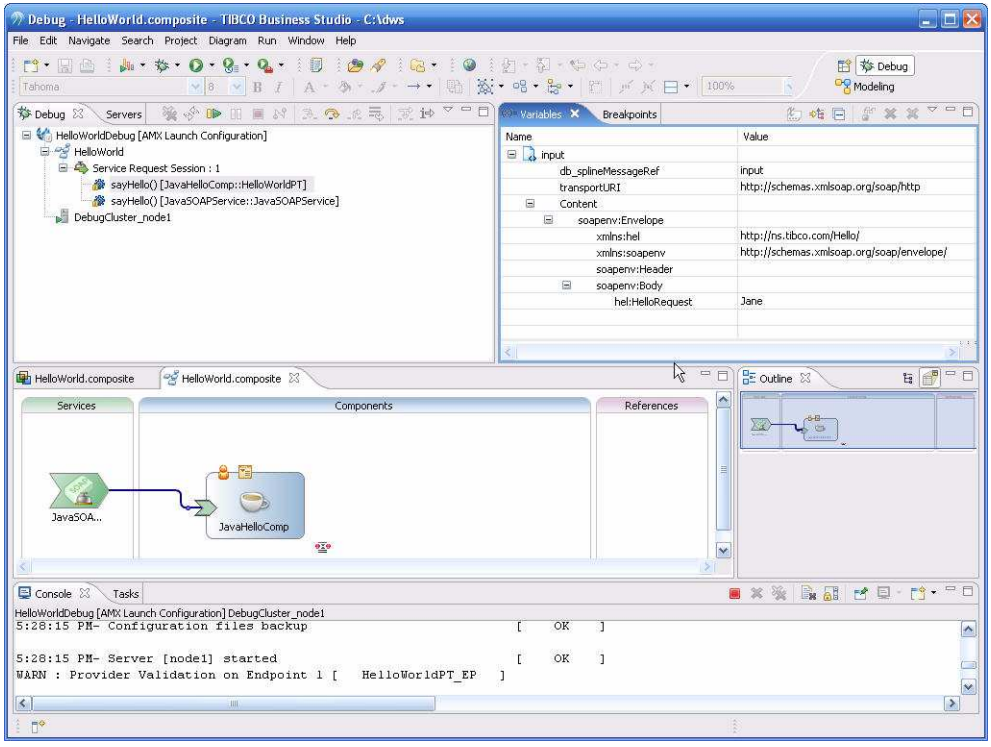
Figure 6 Component Property Sheet


Properties
Problems
Registries
General
Services
Implementation
References
Shared Resource Profiles
Properties
Appearance
Component
Name:
JavaHelloComp
Description:
Version:
1.0.0
Type:
Java

Debugger

The TIBCO Business Studio debugger provides a testing environment for stepping through composite elements and determining the sources of errors. Figure 7 on page 33 shows the debugger in the process of debugging a sample HelloWorld composite. Breakpoints have been set before and after the Java component executes, and the debugger is stopped at the before breakpoint. In the Variables view on the top-right, the value of the request is being examined.

Figure 7 Debugger



Service Assembly Editor

In order to be deployed, composites must be transformed into service units and service assemblies. Figure 8, shows a sample service assembly editor view.

Figure 8 Service Assembly Editor

TIBCO ActiveMatrix Administrator

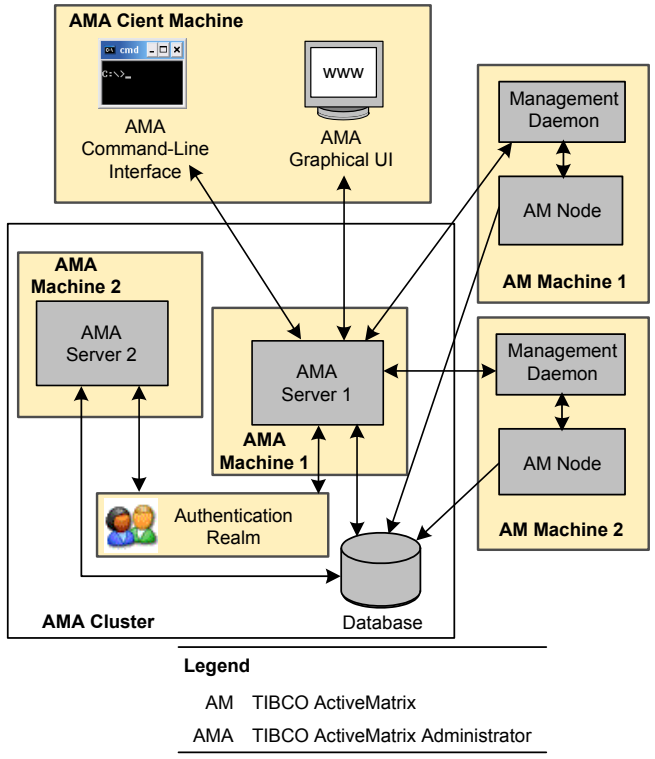
ActiveMatrix provides TIBCO ActiveMatrix Administrator for enterprise, environment, and service management. ActiveMatrix Administrator supports both graphical and command-line interfaces.

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

TIBCO ActiveMatrix Administrator Architecture

Figure 9 on page 35 shows ActiveMatrix Administrator components, and the relationship between ActiveMatrix Administrator, other servers, and ActiveMatrix machines and nodes.

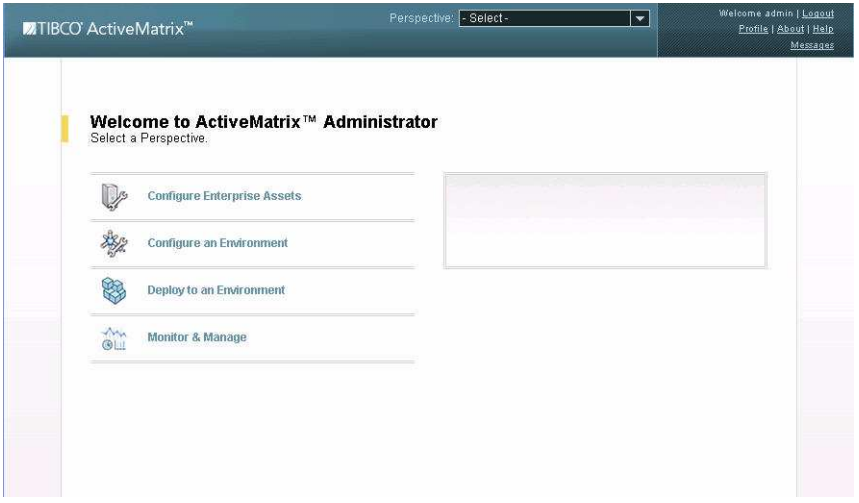
Figure 9 TIBCO ActiveMatrix Administration Architecture



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

- **TIBCO ActiveMatrix Administrator Server** Gathers management data from nodes, responds to requests from the ActiveMatrix Administrator graphical and command-line UIs, interacts with the authentication realm server to authenticate users, and interacts withTIBCO Management Daemon to manage nodes.
- **TIBCO ActiveMatrix Administrator Cluster** Groups one or more ActiveMatrix Administrator servers. ActiveMatrix Administrator servers within a cluster share a database and authentication realm and are kept synchronized.
- **ActiveMatrix Database** Stores ActiveMatrix administration data.
- **Authentication Realm** Manages user authentication data. The authentication realm can be provided either by TIBCO Administrator or by another server or a file.
- **ActiveMatrix Administrator Graphical UI** Displays the ActiveMatrix Administrator user interface. Figure 10 on page 36 shows the ActiveMatrix Administrator graphical UI welcome page. In ActiveMatrix Administrator, functionality is divided into perspectives. A *perspective* is a set of controls used to carry out a category of administration tasks.
- **ActiveMatrix Administrator Command-Line Interface** Provides a script-based interface to ActiveMatrix Administrator functions.
- **Management Daemon** Gathers installation information and exposes ActiveMatrix node life cycle operations.

Figure 10 TIBCO ActiveMatrix Administrator



Enterprise and Environment Administration

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

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

Service Administration

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

Service Deployment

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

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

Highly Available Services

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

Load Balanced Services

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

Service Monitoring and Management

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

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

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