TIBCO Adapter™ for J. D. Edwards OneWorld Xe

User's Guide

Software Release 5.3.0
December 2007

Important Information

Contents

Figures

Tables

Preface

TIBCO Adapter for J. D.Edwards OneWorld Xe allows integration between J. D. Edwards OneWorld Xe and other applications, using TIBCO infrastructure components. The adapter provides bi-directional communication with the J. D. Edwards OneWorld Xe application through a rich set of interfaces. Real-Time events from J. D. Edwards OneWorld Xe can be published using the adapter's Publication service. The adapter's Subscription and Request-Response services allow external applications to execute BusinessFunctions or perform XML based queries in the J.D. Edwards OneWorld Xe system.

This manual tells you how to install, configure and use TIBCO Adapter for J. D.Edwards OneWorld Xe. It includes examples and exercises to help you get started in using the adapter.

Topics

Related Documentation

This section lists documentation resources you may find useful.

TIBCO Product Documentation

The following documents form the Product documentation set:

- *TIBCO Adapter Concepts* — Read this manual to gain an understanding of adapters in general that you can apply to the various tasks you may undertake.
- *TIBCO Adapter for J. D. Edwards OneWorld Xe User's Guide* — This manual explains concepts relating to the adapter and the application with which it interacts. Installation, configuration and deployment information is included in this manual.
- *TIBCO Adapter for J. D. Edwards OneWorld Xe Examples Guide* — Read this manual for instructions on creating adapter instances of different types, creating and populating repositories, and testing the configurations.
- *TIBCO Adapter for J. D. Edwards OneWorld Xe Release Notes* — Read this document for information about new features, deprecated features, and open and closed issues.
- *README for TIBCO Adapter for J. D. Edwards OneWorld Xe* — Read this document to check the current release number and see a summary of software and hardware requirements for installing and running the adapter.

Other TIBCO Product Documentation

You may find it useful to read the documentation for the following TIBCO products. Note that only books that relate to adapters are listed. Each of the books is available from the `doc` directory in the product's installation area:

- TIBCO ActiveEnterprise™ software:
  - *TIBCO ActiveEnterprise Concepts*
- TIBCO Designer™ software:
  - *TIBCO Designer User's Guide*
  - *TIBCO Designer Palette Reference*
  - *TIBCO Designer Release Notes*

- TIBCO Administrator™ software:
  - *TIBCO Administrator User's Guide*
  - *TIBCO Administrator Server Configuration Guide*
  - *TIBCO Administrator Release Notes*
- TIBCO BusinessWorks™ software:
  - *TIBCO BusinessWorks Concepts*
  - *TIBCO BusinessWorks QuickStart*
  - *TIBCO BusinessWorks Process Design Guide*
  - *TIBCO BusinessWorks Palette Reference*
  - *TIBCO BusinessWorks Installation*
  - *TIBCO BusinessWorks Release Notes*
- TIBCO IntegrationManager™ software:
  - *TIBCO IntegrationManager Concepts*
  - *TIBCO IntegrationManager Administrator's Guide*
  - *TIBCO IntegrationManager Process Design Guide*
  - *TIBCO IntegrationManager Reference*
  - *TIBCO IntegrationManager Release Notes*
- TIBCO Rendezvous™ software:
  - *TIBCO Rendezvous Concepts*
  - *TIBCO Rendezvous Administration*
  - *TIBCO Rendezvous Configuration Tools*
- TIBCO Enterprise Message Service™ software:
  - *TIBCO Enterprise Message Service User's Guide*
  - *TIBCO Enterprise Message Service Installation*
  - *TIBCO Enterprise Message Service Application Integration*
  - *TIBCO Enterprise Message Service Release Notes*
- TIBCO Hawk® software:
  - *TIBCO Hawk Installation and Configuration*
  - *TIBCO Hawk Administrator's Guide*

- TIBCO Adapter™ SDK
  - *TIBCO Adapter SDK Concepts*
- TIBCO Runtime Agent™ software
  - *TIBCO Runtime Agent Release Notes*
  - *TIBCO Runtime Agent Installation*
  - *TIBCO Runtime Agent Domain Utility User's Guide*
  - *TIBCO Runtime Agent Upgrading to Release 5.5*

Typographical Conventions

The following typographical conventions are used in this manual.

Table 1 General Typographical Conventions

Convention	Use
`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*
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 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 Concepts

This chapter explains how TIBCO Adapter for J. D. Edwards OneWorld Xe interacts with the J. D. Edwards OneWorld Xe application.

Topics

Adapter Overview

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

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

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 BusinessWorks.

Adapter Components

The adapter has two main components: design time and run time.

- The design-time component consists of the adapter palette. The adapter palette is an adapter-specific GUI that seamlessly integrates with TIBCO Designer. It allows you to configure the adapter, and will store the configuration in persistent storage.
- The run-time component uses this configuration to publish real-time events using the Publication service, post data to the TIBCO ActiveEnterprise applications using the Subscription service, or allow exchange of data between the TIBCO ActiveEnterprise applications and the J. D. Edwards OneWorld Xe system using the Request-Response Server service.

Adapter Features

This section summarizes the features of the adapter. These features make use of the interoperability that characterizes the J.D. Edwards OneWorld Xe system:

An Easy-to-Use GUI

The adapter provides its own design-time component, namely the adapter palette, which 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, subscription and request-response server services.

Support for Querying Master Data

The J. D. Edwards OneWorld Xe system stores all enterprise data within the J. D. Edwards OneWorld Xe System in database tables. The adapters' Request-Response service queries these application tables using the XML Query List function. If the query returns a very large data set, the adapter can also distribute the data across multiple invocations. Streaming the data across multiple invocations prevents the client programs from being overwhelmed by huge amounts of data in a single invocation.

Support for Mapping and Translation

The adapter performs mapping of functions and translation of data formats between external applications and the corresponding J.D. Edwards OneWorld Xe Business Functions. You can use TIBCO BusinessWorks or TIBCO IntegrationManager 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, re-cycle, reuse and release connections to J. D. Edwards OneWorld Xe.

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 J. D. Edwards OneWorld Xe system, the connection pooling mechanism automatically recreates 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 Business Functions 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 J.D. Edwards OneWorld Xe system. Authentication is configured at design-time when setting connection parameters.

Support for 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.

Support for Dual Message Transports

The adapter supports the following message transports:

- `TIBCO Rendezvous Transport` — This transport uses subject-based addressing to provide support for both multicast or broadcast and point-to-point communications. You can configure the delivery modes of the messages and the wire format used when you configure the adapter service.
- `JMS Transport` — TIBCO Enterprise for JMS must be installed to use the JMS transport. The JMS administration interfaces allow you to create and manage

administered objects such as Connection Factories, Topics, and Queues. JMS clients can retrieve references to these objects by using Java Naming and Directory Interface (JNDI). Creating static administered objects allows clients to use these objects without having to implement the object within the client. When a JMS client starts, it performs a JNDI lookup for the connection factories that it needs. For details on JNDI, see the *TIBCO Enterprise for JMS User's Guide*. You can configure the connection factory type and the delivery mode using TIBCO Designer, when you configure the adapter service.

Quality of Service Support

The adapter supports the following levels of service for TIBCO Rendezvous wire formats:

- Reliable message delivery, which ensures that each multicast or broadcast message is received as long as the physical network and packet recipients are working, and that the loss of a message is detected.
- Certified message delivery, which guarantees that every certified message reaches its intended recipient and is received in the order sent.
- Distributed Queue, explained in Distributed Queue Support on page 5.

The adapter supports the following levels of service for the JMS wire formats:

- Durable
- Non-durable

Distributed Queue Support

A distributed queue is a group of cooperating transport objects, each in a separate process. Each transport object is called a member. To balance the transmission load among servers, the adapter can use distributed queues for *one-of-n* delivery of messages to a group of servers. Each member of a distributed queue must listen for the same subject using the TIBCO Rendezvous Distributed Queue listener objects. Even though many members listen for each inbound message (or task), only one member processes the message. For details on distributed queues, see *TIBCO Rendezvous Concepts*.

In the queue mode within TIBCO Enterprise for JMS, each listener is a single receiver of a point-to-point message. However, the listeners can be configured as a set of receivers, each of which receives a fraction of the messages. For details on TIBCO Enterprise for JMS distributed queues, see the *TIBCO Enterprise for JMS User's Guide*.

Load balancing for the processing of TIBCO Rendezvous or JMS certified messages is supported using distributed queuing. The messages from TIBCO Rendezvous or TIBCO Enterprise for JMS are distributed equally among all instances that belong to the same group. This distributes the data load over several adapter instances. However, the order in which the data is sent to the application server is not guaranteed.

TIBCO Wire Format Support

Two wire formats are supported by the adapter:

- ActiveEnterprise Message wire format is the TIBCO ActiveEnterprise standard wire format. It can include metadata information.
- JMS Message format is a standard message which may have XML content. It follows Sun's Java Messaging Service wire format.

Tracing and Exception Handling Support

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

Use of the TIBCO Adapter for J. D. Edwards OneWorld Xe

Consider a scenario where a customer buys few items from a Store. The Store manages all its enterprise functions using the J. D. Edwards OneWorld Xe system. The Point of Sale clerk creates an invoice; the customer pays by a store credit card. The delivery of items will be made to the customer in few weeks time. The customer must also receive a detailed confirmation of his order and a statement of balance or amount he owes to the store.

The scenario above can be accomplished with the TIBCO Adapter for J. D. Edwards OneWorld Xe and an integration platform such as TIBCO BusinessWorks or TIBCO IntegrationManager.

The system integrator carries out the following tasks:

- Creates a real-time outbound event for the invoice document in the J. D. Edwards OneWorld Xe system.
- Creates a new adapter configuration using the TIBCO Adapter for J. D. Edwards OneWorld Xe palette in the TIBCO Designer, and configures a new publication service and chooses the real-time event he just created as the publication data event.

Whenever a new invoice document is created or an existing one modified, the J. D. Edwards OneWorld Xe system automatically sends the modified or the new document to the adapter in the form of a real-time event. The adapter will publish this document for consumption by other TIBCO applications.

The system integrator uses TIBCO BusinessWorks to map the invoices (published as real-time events) to an accounting system. In the accounting system, information from the invoice is used to update the customer's account and eventually prepare a monthly statement.

The information about the line items or items the customer ordered and information pertaining to the ship to address of the customer is published to the shipping department, which is responsible for packaging and delivering the order to the customer. All the above transactions happen in real-time, immediately after the Customer places an order at the Point of Sale counter.

Adapter Services

The adapter provides the following services

- Publication Service
- Subscription Service
- Request-Response Service

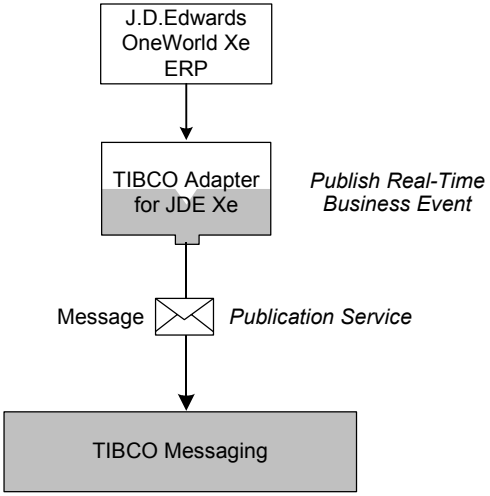
Publication Service

The adapter uses J.D. Edwards OneWorld Xe real-time event notification to publish data generated by J.D. Edwards OneWorld Xe applications to TIBCO ActiveEnterprise applications.

If the J. D. Edwards OneWorld Xe system has been configured so that an event is generated whenever data changes inside the system, the adapter can publish this event to TIBCO ActiveEnterprise applications. A typical example is the publication of a sales order event whenever a new sales order is created.

In this publication service, the real-time event (an invoice prepared by the point-of-sale clerk, for example) originates in the J. D. Edwards OneWorld Xe system and is published by the adapter to other applications, such as "the BackOffice" Accounting System.

Figure 1 Typical Publication Service Flow



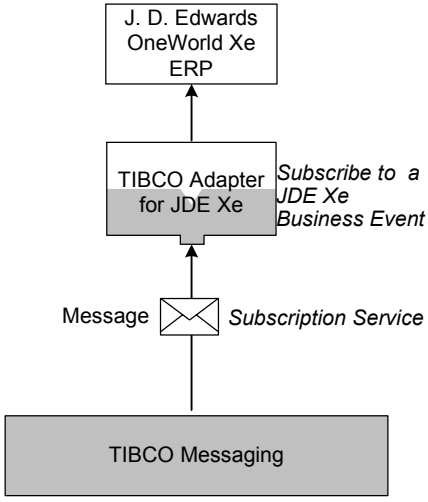
Subscription Service

In the subscription service scenario, the subscription service of the adapter gets the message from the TIBCO environment and sends it to the J. D. Edwards OneWorld Xe application. A subscription service does not return a response.

The adapter unpacks events from the message and performs a Business Function invocation on the J. D. Edwards OneWorld Xe enterprise system. The adapter can invoke one Business Function, or it can invoke multiple Business Functions that are chained together in a group.

For example, `CalculateQtyOnHand` is a single Business Function. However, a sales order can contain multiple Business Functions. For example, the task `CreateSalesOrder` requires multiple Business Functions `AddHeader`, `AddLine` (multiple times), and `EndDoc` to be executed in order. `CreateSalesOrder` is an example of a Business Function Group.

Figure 2 Typical Subscription Service Flow



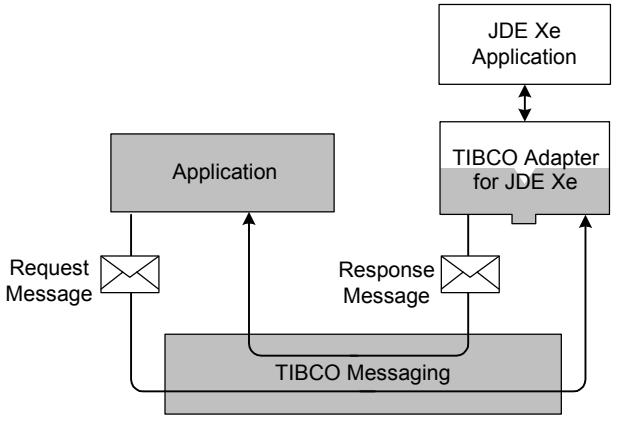
Request-Response Service

The adapter, acting as a Request-Response server, enables and allows the TIBCO ActiveEnterprise clients (for example, TIBCO BusinessWorks or TIBCO IntegrationManager) to invoke Business Functions or Business Function Groups on the adapter. It also allows the users to perform efficient table queries with the XML QueryList operation. The adapter executes these operations in the J. D. Edwards OneWorld Xe system and returns the result of the invocations.

For extracting data from J. D. Edwards OneWorld Xe database tables, views, or table conversions, the adapter has a request-response Query service. The database table or view information is required when a one time data load is required in an external application. For example, if an application is implemented in the enterprise after the J. D. Edwards OneWorld Xe ERP system was implemented, using request response, the customer master information can be extracted from J. D. Edwards OneWorld Xe and imported into the new system. Similarly, item catalog information may be loaded from the J. D. Edwards OneWorld Xe system to the CRM system.

Using the request-response service, the adapter gets a request from the TIBCO environment. It unpacks the request and converts it into a format understandable by the J. D. Edwards OneWorld Xe System. The adapter then uses this data to make a Business Function or XML query list invocation in the J. D. Edwards OneWorld Xe system. The data returned by the resulting invocation is packed into a reply and sent back to the calling application. If any errors are encountered during invocation they are sent back in the error field of the reply.

Figure 3 Typical Request Response Service Flow



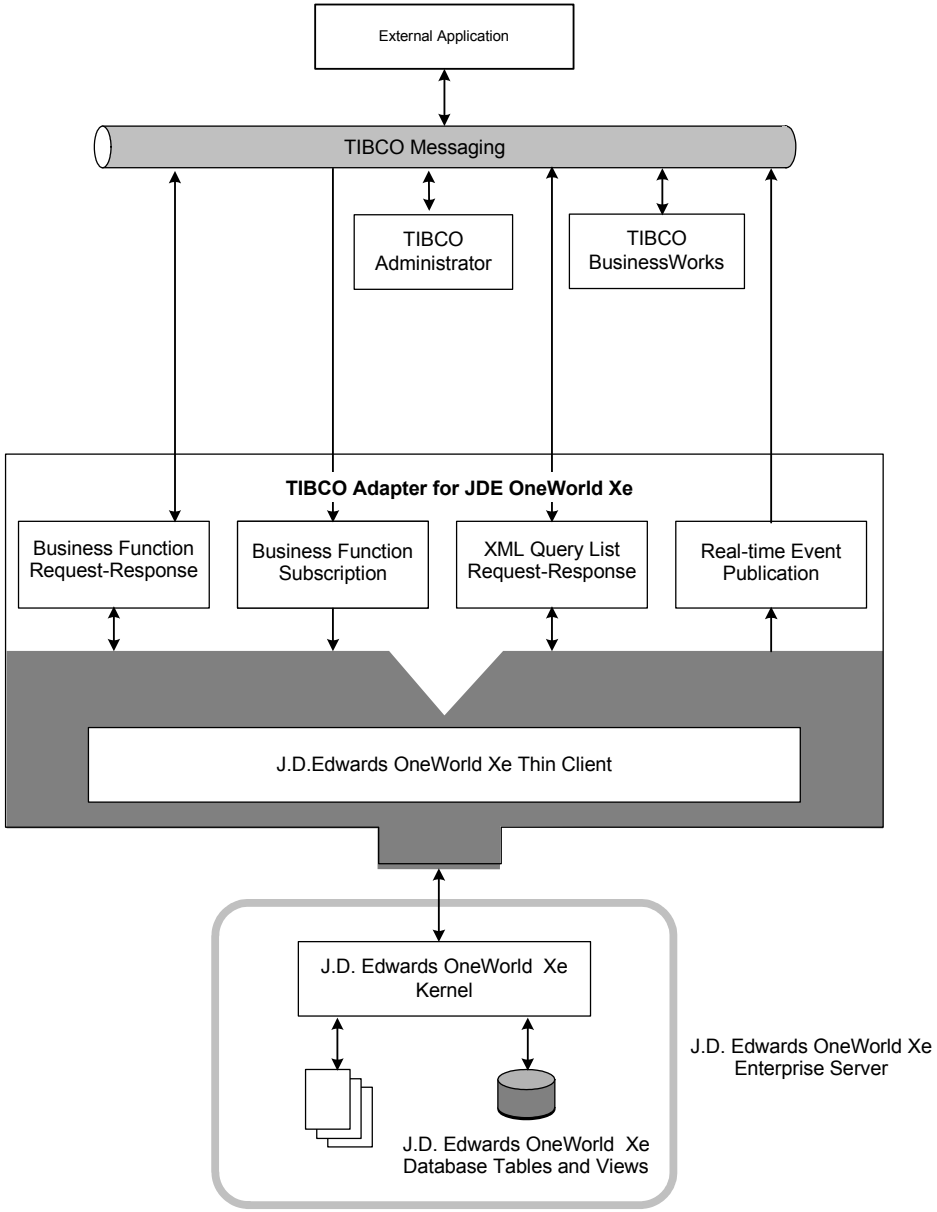
Adapter and J. D. Edwards OneWorld Xe Interaction

This section provides a brief overview of the way that the TIBCO Adapter for J. D. Edwards OneWorld Xe interacts with the J. D. Edwards OneWorld Xe system and with TIBCO ActiveEnterprise applications such as TIBCO BusinessWorks.

Logical Architecture

A typical logical architecture that integrates the adapter is shown in the following figure. The adapter can run on the machine that hosts the J. D. Edwards OneWorld Xe server, or on a different machine. The adapter requires the J. D. Edwards OneWorld Xe thin client libraries on the machine on which the adapter is running.

Figure 4 Logical Architecture



Typical Workflow

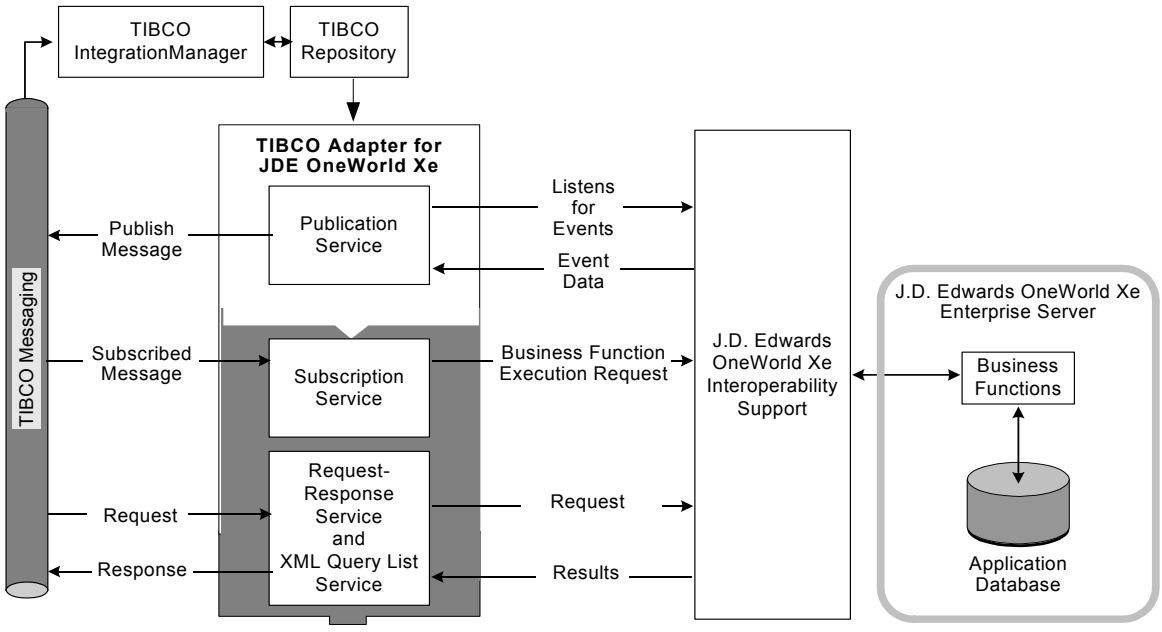
The next figure shows the adapter integrated with J. D. Edwards OneWorld Xe and illustrates the typical flow of a business function execution, and the adapter services related to the business function execution.

TIBCO Administrator manages the repositories that contain configuration data and schema used by the adapter services. TIBCO BusinessWorks is used to transform data.

As shown in the diagram, four adapter services are available: publication, subscription and two types of request-response.

- Using the publication service, the adapter listens for real-time events on J. D. Edwards OneWorld Xe server and publishes them using a TIBCO messaging transport.
- Using the subscription service, the adapter listens for messages sent from the ActiveEnterprise environment and posts them to the J. D. Edwards OneWorld Xe server.
- The adapter uses the request-response service to pass requests from an application in the ActiveEnterprise environment to the J. D. Edwards OneWorld Xe server, then returns the responses to the application.
- The XML Query List service is also a request-response service that returns Application Table data from the J. D. Edwards OneWorld Xe server. This service can use J.D.Edwards Tables, Views or Table Conversions to return the data.

Figure 5 Typical Workflow



Chapter 2 Installation

This chapter explains how to install the adapter on Microsoft Windows and UNIX systems.

This software may be available on multiple operating systems. However, not all operating system platforms for a specific software version are released at the same time. Please see Table 6 on page 30 and Table 7 on page 36 for the availability of this software version on a specific operating system platform.

In TIBCO Adapter for J.D. Edwards OneWorld Xe user documentation, J.D. Edwards OneWorld Xe can refer to any of the following:

- J.D. Edwards OneWorld Xe
- PeopleSoft EnterpriseOne 8.9
- PeopleSoft EnterpriseOne 8.10
- PeopleSoft EnterpriseOne 8.11
- J.D. Edwards EnterpriseOne 8.12

Topics

Preparing your Environment for Installation

The most time-consuming part of an adapter installation is the collection of environment information and parameters. This section helps you complete this process. It provides a check list of parameters you should obtain from various system administrators within your organization before installing the adapter. Note that obtaining a J. D. Edwards OneWorld Xe account can take some time depending on your corporate policies — so plan in advance!

Operating System Requirements

Obtain the following information from the administrator of the machine on which you plan to install the adapter:

System name:
System IP address:

Username and password to access the system and run the adapter:

Username:
Password:

Do you have the required credentials to run the installer?

- On Microsoft Windows, administrator privileges are required to install.
- On UNIX systems, you can install as root or a regular user. See Installer Account on page 32 for details.
- Note that the TIBCO Runtime Agent (TRA) must be installed prior to installing the adapter and the adapter installation always places files under the TIBCO root directory that was set when the TIBCO Runtime Agent was installed.

Is there enough space on that disk or partition to install the adapter? The adapter needs space in your temp area and the directory where it is installed.

- See Installation Registry on page 26 for details about temp folder space requirements on Microsoft Windows and UNIX systems.
- See Table 6 on page 30 for Microsoft Windows installations
- See Table 7 on page 36 for Unix System installations.

You must have write permissions to these directories to install the adapter. In addition, on UNIX systems certain other permissions must be set to run the adapter. See Permission Requirements on UNIX Systems on page 39 for details.

- To run the adapter you must have permissions to access the project where adapter configuration is stored.
- Depending on whether TIBCO Administrator is used to set access permissions, you may need an account identified by Administrator. See the *TIBCO Administrator User's Guide* for details.

Determine how the adapter installation files are to be transferred to this system. The installation files can be downloaded from download.tibco.com (if you have an account setup to download).

Requirements for J.D. Edwards Applications

The adapter works with J. D. Edwards OneWorld Xe enterprise server running on any platform supported by Oracle PeopleSoft (formerly J.D.Edwards). This includes IBM AS 400 and AIX, HP-UX, Sun Solaris, Microsoft Windows, and so on. Chapter 3 on page 45 mentions enterprise server configuration for many different platforms.

Determine which J.D. Edwards OneWorld Xe versions are supported.

- See Supported J.D. Edwards Applications on page 31 for Microsoft Windows.
- See Supported J.D. Edwards Applications on page 36 for UNIX systems.

Obtain the following information from your J. D. Edwards OneWorld Xe administrator. This is a list of J. D. Edwards OneWorld Xe server parameters that you will need to configure the adapter, along with the J.D. Edwards OneWorld Xe credentials that the adapter requires.

For the adapter to connect to the J. D. Edwards OneWorld Xe enterprise server, JAR files for the java connectivity API must be copied to the machine on which the adapter will be installed. These JAR files can be found in the `system/classes` directory on the enterprise server.

Table 2 J.D. Edwards Connectivity JAR Files

Software	Jar Files	Location
J.D. Edwards OneWorld Xe	• `connector.jar` • `kernel.jar`	`System/classes` directory on Enterprise Server

Table 2 J.D. Edwards Connectivity JAR Files

Software	Jar Files	Location
PeopleSoft EnterpriseOne 8.9	• `connector.jar` • `kernel.jar` • `jdeutil.jar` • `database.jar` • `log4j.jar`	System/classes directory on Enterprise Server.
	JDBC driver jar file. For example, `Classes12.jar` for Oracle	*<ORACLE_INSTALLATION_HOME>* `/ora92/jdbc/lib`
PeopleSoft EnterpriseOne 8.10	• `connector.jar` • `kernel.jar` • `jdeutil.jar` • `database.jar` • `log4j.jar`	System/classes directory on Enterprise Server.
	JDBC driver jar file. For example, `Classes12.jar` for Oracle.	*<ORACLE_INSTALLATION_HOME>* `/ora92/jdbc/lib`

Table 2 J.D. Edwards Connectivity JAR Files

Software	Jar Files	Location
PeopleSoft EnterpriseOne 8.11	• ApplicationAPIs_JAR.jar • ApplicationLogic_JAR.jar • Base_JAR.jar • BizLogicContainerClient_JAR.jar • BizLogicContainer_JAR.jar • castor.jar • Connector.jar • EventProcessor_EJB.jar • EventProcessor_JAR.jar • JdbjBase_JAR.jar • JdbjInterfaces_JAR.jar • JdeNet_JAR.jar • log4j.jar • PMApi_JAR.jar • Spec_JAR.jar • System_JAR.jar • xerces.jar	System/classes directory on Enterprise Server.
	JDBC driver jar file. For example, Classes12.jar for Oracle	*<ORACLE_INSTALLATION_HOME>* /ora92/jdbc/lib
J.D.Edwards EnterpriseOne 8.12	• All files listed in EnterpriseOne 8.11 • jdelog.properties	System/classes directory on Enterprise Server.

Copy all these jar files into the same location on the adapter machine. During the installation of the adapter, you are prompted for this information.

Test Connectivity

If you install the J.D. Edwards OneWorld Xe client, start it and ensure you can connect to the J.D. Edwards OneWorld Xe server using the username and password that you were provided by your administrator.

Pre-Installation Worksheet

Use this form to capture the information you will need to collect before starting installing the adapter

Adapter Machine Information

Table 3 Adapter Machine Information

Field Name	Field Description	Field Value
Hostname (Example: adapter1.tibco.com)	Name of the machine on which the adapter is being installed.	
IP address (Example: 192.168.12.12)	IP address of the machine on which the adapter is being installed.	
User account (Example: administrator)	User account to be used for the installation.	
User domain (if Windows) (Example: ENGR2)	Network domain to which the user belongs.	
User password		
Disk and path on which to install adapter (Example: /opt/tibco)		
How will machine be accessed	[ ] directly [ ] terminal server [ ] xterm [ ] telnet [ ] other: ……	
How will installation files be transferred to machine	[ ] CD-drive [ ] internet download [ ] FTP to machine [ ] network disk mounting	

J.D. Edwards OneWorld Xe Information

Field Name	Field Description	Field Value
User account (Example: jsmith)	User account to be used for the installation	
User password		
Enterprise Server hostname (Example: adapter1.tibco.com)		

Field Name	Field Description	Field Value
Listener Port		
Transaction Server hostname (Applies only to PeopleSoft EnterpriseOne version 8.11 or higher)		
Transaction Server HTTP Listener port(Applies only to PeopleSoft EnterpriseOne version 8.11 or higher)		The default value is 9082.

J.D. Edwards OneWorld Xe Software

Please specify where software for the J.D. Edwards OneWorld Xe can be found within your organization:

[ ] It's already installed on adapter machine.

Location on disk: …….

[ ] Installation files are available via :

FTP (Server : …… User : …… Password: ……)

Disk mount (full path : ……..)

CD provided during install by (name/extension) : ……..

Installer Overview

The installer allows you to run it in different modes. Each mode is supported on all platforms.

- GUI mode
- Console mode
- Silent mode

GUI Mode

In GUI mode, the installer presents panels that allow you to make choices about product selection, product location, and so on. When you invoke the installer by double-clicking on the icon, GUI mode is used.

Console Mode

Console mode allows you to run the installer from the command prompt or terminal window. This is useful if your machine does not have a Windows environment.

Silent Mode

Silent mode either installs using default settings or uses a response file that was saved during an earlier installation. Silent mode installs without prompting you for information.

- If no response file has been recorded earlier and you invoke the installer with the `-silent` argument, the default installation parameters are used.
- If a response file exists, and the installer is started with `-options` <*responseFileName*> as an argument, the installer uses the values specified by the user when the response file was generated.

Upgrading the Adapter

Software from TIBCO uses three numbers to indicate whether the release is major, minor or a patch. For example, 5.0.0 indicates a major release, 5.2.0 indicates a minor release and 5.2.1 indicates a patch release. The installer for a patch release performs an automatic upgrade. For example, the installer automatically upgrades TIBCO Runtime Agent 5.0.0 to 5.0.1 by overwriting the contents of the 5.0 directory.

For a major and minor release, the installer prompts whether you wish to upgrade, and informs you if incompatible products are on your system. If you proceed, major or minor releases are installed under a new directory that is named using the major or minor release numbers.

For example, if you have installed the 5.0.0 release and are upgrading to a 5.3.0 minor release, it will be installed under the 5.3 directory. This allows both the 5.0 and 5.3 releases to coexist on the same machine.

If you are upgrading the adapter, or reinstalling a clean version of the software, you may uninstall the product first or allow the installer to perform the upgrade or reinstall.

Note that, if you are reinstalling over the same adapter version:

- You are not prompted to supply the installation location. The software is automatically reinstalled where the previous version was installed.
- If any files are currently locked (that is, in use), the installer marks the file for deletion in the install location. After installation, the installer prompts you to reboot your system. You must reboot before using the software.

Uninstalling the Adapter

- If another product is dependent on the product you wish to uninstall, you are informed that you must uninstall the other product first.

Microsoft Windows

Use one of the following options to uninstall the Adapter from the Microsoft Windows platform:

- Click **Start>Programs>TIBCO><*Adapter*>Uninstall**
- Navigate to the `_uninst` directory located in the Adapter installation directory and invoke the `Tibuninstall.exe` program.
- Click **Start>Programs>TIBCO>TIBCO Installation Manager**
- Use Add/Remove Programs from the Control Panel.

UNIX

Use one of the following options to uninstall the Adapter from the supported UNIX platform:

- Navigate to the `_uninst` directory located in the Adapter installation directoryand invoke the `Tibuninstall.bin` program.

- Run TIBCO Installation Manager which is located in the *<install-path>*`/tibco/TibcoInstallationManager.bin`.

Installation Registry

The installer maintains an installation registry. The registry location depends on the platform. This section explains where the registry files are located. The files have vpd as a prefix, which stands for Vital Product Database. Note that the installer does not recognize TIBCO ActiveEnterprise 4.x products.

Do not edit, modify, rename, move, or remove any of the registry vpd files.

Microsoft Windows Platforms

ActiveEnterprise 5.1 products maintain the installation registry in the `SystemDrive:\Windows` directory. The following files represent the installation registry:

```
SystemDrive:\Windows\vpd.properties
SystemDrive:\Windows\vpd.properties.tibco.systemName
```

Installer Disk Space Requirements in Temporary Area

The entire package is extracted into a temp folder, typically `SystemDrive:\Temp` or `SystemDrive:\Documents and Settings\<user_name>\Local Settings\Temp`.

The installer requires 17 MB of free space in the temp directory.

UNIX Platforms

The installation registry is maintained in the following files in the user's home directory:

```
User_Home_Directory/vpd.properties
User_Home_Directory/vpd.properties.tibco.systemName
```

If installation is performed by super-user (root), the installation registry is maintained as follows:

- On Solaris, in the root user's home directory (which is `/`) as two vpd files.

Installer Disk Space Requirements in Temporary Area

The installer launcher first extracts a Java Virtual Machine (JVM) in a temporary directory and uses this JVM to launch itself. The size of the extracted JVM differs from platform to platform.

On UNIX platforms the following disk space is required in the temporary area:

- On Solaris, 50 MB of free disk space in `/var/tmp`

If your system does not have sufficient free disk space in the above temporary area, you can still run the installer with a different temporary area by using the following option when starting the installer:

```
<install_package_name>.bin -is:tempdir /new_tmp
```

where `/new_tmp` has sufficient free disk space.

Disk Space Requirement in User's Home Directory

On UNIX platforms when a regular (non-root) user installs a TIBCO 5.x product, the installation registry (two vpd files) is maintained in the user's home directory. As more products are installed, entries are added into these vpd files.

The user's home directory must at least have 500 KB of free disk space.

Installation History

The installer and uninstaller creates a file called `TIBCOInstallationHistory.xml` in the same location where the installation registry is created. Each time an installation and uninstallation is performed, entries are appended to the file.

On Windows: `SystemDrive:\Windows\TIBCOInstallationHistory.xml`

On UNIX: `Users_Home_Directory/TIBCOInstallationHistory.xml`

The file `TIBCOInstallationHistory.xml` therefore contains the record of all installation and uninstallation activities of all products, features and components.

Do not edit, modify, rename, move, or remove the `TIBCOInstallationHistory.xml` file.

Adapter Components and Compatible Software

You can install different adapter components on different machines. For example, you can run the runtime adapter on one machine and install the design-time components on another machine. This allows you to configure an adapter on one machine and run it on another.

Adapter Components

Table 4 describes the adapter components on the adapter installation package.

Table 4 TIBCO Adapter components

Component	
Run-time adapter	This process does the actual work of passing and converting data to and from the vendor application. Parameters of data exchanges are stored in projects created using the adapter palette.
Adapter palette	Adapter-specific GUI that is loaded in TIBCO Designer (see next section for details) at configuration time.

Required and Optional TIBCO Products

Depending on the tasks you wish to perform, you must install one or more other TIBCO products. The next table describes required and optional products and their purpose. Refer to the `Readme.txt` for version numbers of the following products.

Table 5 Required and Optional TIBCO Products

Component	Purpose
TIBCO Runtime Agent	Required. TIBCO Runtime Agent supplies a number of TIBCO and third-party libraries used by the adapter and other TIBCO products both at design-time and runtime. This includes, for example, TIBCO Rendezvous software. You must install TIBCO Runtime Agent on each machine that hosts an adapter. TIBCO Runtime Agent must be installed before you install the adapter.

Table 5 Required and Optional TIBCO Products

Component	Purpose
TIBCO Administrator	Required. TIBCO Administrator includes the following modules: • User Management. Management of authentication, roles and users, that is, connecting roles (groups) and users to access control lists (ACLs). This includes security for server-based projects at design-time and for deployed applications at runtime. • Resource Management. Monitoring of machines and of 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. Uploading of Enterprise Archive (EAR) files, creation, configuration, deployment, and monitoring of applications. This console is also used to start and stop applications. TIBCO Administrator is available as a separate installation and can be installed after installing the adapter.
TIBCO BusinessWorks	Optional. TIBCO BusinessWorks is a scalable, extensible, and easy to use integration platform that allows you to develop integration projects. TIBCO Adapters are typically part of integration projects created using BusinessWorks. TIBCO BusinessWorks is available as a separate installation and can be installed after installing the adapter.
TIBCO EnterpriseMessage Service	Optional. TIBCO Enterprise Message Service (EMS) allows you to use the Java Messaging Services (JMS) as the message transport for your adapter. TIBCO Enterprise Message Service is available as a separate installation and can be installed after the adapter is installed.

Installing on Microsoft Windows

Before starting the installation procedure, review the topics in this section to determine that your system meets the basic requirements and that you have the prerequisite software installed.

The following is a list of prerequisites for installing the adapter on Microsoft Windows systems. See Installer Disk Space Requirements in Temporary Area on page 26 for additional disk space requirements.

The following is a list of prerequisites for installing the adapter on Microsoft Windows systems, See Installer Disk Space Requirements in Temporary Area on page 26 for additional disk space requirements.

The following table also lists the platform-specific installation packages, where <version_num> is the Adapter release number. For example, the installation package name for the TIBCO Adapter for J.D.Edwards OneWorld Xe 5.3.0 on the Microsoft Windows 2000 platform is `TIB_adjdexe-simple_<version_num>_w32.exe`.

Table 6 Supported platforms, hardware, package names, and disk space for Microsoft Windows

Platform	Hardware	Package Names	Disk Space (MB)
Microsoft Windows 2000	x86	`TIB_adjdexe-simple_<version_num>_w32.exe`	51
Microsoft Windows XP Professional			
Microsoft Windows 2003			

Supported J.D. Edwards Applications

The supported J.D. Edwards applications are:

- J.D. Edwards OneWorld Xe
- PeopleSoft EnterpriseOne 8.9
- PeopleSoft EnterpriseOne 8.10
- PeopleSoft EnterpriseOne 8.11
- J.D.Edwards EnterpriseOne 8.12

J.D. Edwards OneWorld Xe

In order to interoperate with TIBCO Adapter for J. D. Edwards OneWorld Xe, you must install J. D. Edwards OneWorld Xe with Service Pack 22. To use Real-Time Event Publication with the adapter, you must also install the following Electronic Software Updates (ESUs) for J. D. Edwards OneWorld Xe:

- `ESU JD13998`
- `ESU JD19327`

Before proceeding to adapter installation, ensure you can connect to the target application using the vendor client.

PeopleSoft EnterpriseOne 8.9 and PeopleSoft EnterpriseOne 8.10

In order to interoperate with TIBCO Adapter for J. D. Edwards OneWorld Xe and use Real-Time Events Publication, you must install PeopleSoft EnterpriseOne 8.9 or PeopleSoft EnterpriseOne 8.10 with Tools Release 8.94 or higher.

Before proceeding to adapter installation, ensure you can connect to the target application using the vendor client.

PeopleSoft EnterpriseOne 8.11

In order to interoperate with TIBCO Adapter for J. D. Edwards OneWorld Xe and use Real-Time Events Publication, you must install PeopleSoft EnterpriseOne 8.11 with the following:

- Service Pack 1
- ESUs: `JJ10474 and JJ10684`
- PeopleSoft EnterpriseOne Transaction Server

  The adapter supports Transaction Server running on Websphere 6.0. For information on how to install and configure Transaction Server for Real Time

Events Publication, refer PeopleSoft Transaction Server product documentation.

Before proceeding to adapter installation, ensure you can connect to the target application using the vendor client.

J.D.Edwards EnterpriseOne 8.12

In order to interoperate with TIBCO Adapter for J. D. Edwards OneWorld Xe and use Real-Time Events Publication, you must install J.D.Edwards EnterpriseOne 8.12 along with tools release 8.96 or higher.

The adapter supports Websphere 6.0 on PeopleSoft EnterpriseOne Transaction Server. For information on how to install and configure Transaction Server for Real Time Events Publication, refer PeopleSoft Transaction Server product documentation.

Before proceeding to adapter installation, ensure you can connect to the target application using the vendor client.

TIBCO Runtime Agent Must be Installed Before the Adapter

Before you can install the adapter, you must install the TIBCO Runtime Agent. If you use the Typical installation, the installer places all libraries and other products required by the adapter into the TIBCO HOME directory.

During installation, the adapter installer checks for the availability of all dependent products in the target system. If any of the dependencies are not available, the installer will immediately exit. Otherwise installation will proceed.

Installer Account

You must have administrator privileges for the machine on which the adapter is installed.

If you do not have administrator privileges, the installer will exit. You must then log out of the system and log in as a user with the required privileges, or request your system administrator to assign the privileges to your account.

Installing from Network Drive

If you intend to install the product on a network drive, you must ensure that the account used for installation has permission to access the network drive.

Installing on Windows 2000 and 2003 Terminal Server

There are two modes in Windows Terminal Server: *Execute* and *Install*. Users are logged on by default in Execute mode, which allows them to run applications. To install an adapter so that everyone can use it, log on as administrator in Install mode. When the adapter is installed in the Install mode, the installation registry is maintained in *SystemDrive*`:\Windows\`.

Windows Terminal Server must be running in remote admin mode, not application sharing mode. The adapter is not supported if installed on a machine that is using Windows Terminal Server in application sharing mode.

The best way to install the adapter on Windows Terminal Server is to use the Add/Remove Programs control panel applet. This automatically sets your mode to Install during the installation and then back to Execute afterwards. Alternatively, you can manually change your mode to Install before starting the installation by typing the following at a command prompt:

```
C:\> change user /install
```

Change back to Execute mode after installation is complete by typing:

```
C:\> change user /execute
```

To check your current mode, type the following:

```
C:\> change user /query
```

Installing the Adapter on Microsoft Windows

You can either download the adapter package or install the package from a CD. The installer prompts you to accept the license agreement, then to choose to perform a typical install or custom install.

- A typical install has minimal prompts and installs standard components in default locations.
- A custom install prompts you to choose which components of the product suite to install and installs only those components.

The installer checks your system for the installation home directory that was established when TIBCO Runtime Agent was installed. The adapter is installed under the installation home directory.

Use one of the following modes to install the software.

Install Using GUI Mode

GUI Mode allows you input values in panels. Type the following at the command prompt:

```
TIB_adjdexe-simple_<version_num>_w32.exe
```

Install Using Console Mode

Console mode allows you to install the software in a non windows environment. The installer will prompt you for values. Type the following at the command prompt:

```
TIB_adjdexe-simple_<version_num>_w32.exe -is:javaconsole -console
```

When running in console mode you can move through the installation process as described next:

```
Enter Key = Moves forward in the installer
2 = Goes back to previous screen
3 = Cancels the Wizard and exits the installation or uninstallation
4 = Redisplays the current screen
```

Install Using Silent Mode

Silent mode allows you to install the software without prompts. Type the following at the command prompt:

```
TIB_adjdexe-simple_<version_num>_w32.exe -silent
```

Install and Generate a Response File

You can generate a response file during installation which you can later use to invoke the installer with the selected values as default values (GUI mode) or as selected values (silent mode).

To install and generate a response file, type the following at the command prompt:

```
TIB_adjdexe-simple_<version_num>_w32.exe -options-record
                                        C:\directory\<responseFile>
```

The response file does not record selections at the component level It does record all other selections, for example, which products you wished to install.

Install Using a Response File

You can use a previously generated response file for installation. For non-silent modes, the response file determine the defaults that are presented. For silent mode, the response file determines what will be installed.

To install using a response file, type the following at the command prompt:

```
TIB_adjdexe-simple_<version_num>_w32.exe -options
C:\directory\<responseFileName>
```

Combining Options

You can combine the different available options. For example, to install in silent mode using a response file, use:

```
TIB_adjdexe-simple_<version_num>_w32.exe -silent -options
<responseFileName>
```

To install using Console mode and generate a response file, use:

```
TIB_adjdexe-simple_<version_num>_w32.exe -is:javaconsole -console
-options-record <responseFileName>
```

Installation on UNIX Systems

Your operating system must meet the minimum patch requirements listed next. See Installer Disk Space Requirements in Temporary Area on page 26 for additional disk space requirements.

The following table also lists the platform-specific installation packages, where <version_num> is the Adapter release number. For example, the installation package name for the TIBCO Adapter for J.D.Edwards OneWorld Xe 5.3.0 on the HP-UX version platform is `TIB_adjdexe-simple_5.3.0_s4_58.bin`.

Table 7 Supported platforms, hardware, package names, patches, and disk space for UNIX systems

Platform	Hardware	Package Names	Minimum Patch(es)	Disk Space (MB)
Solaris 10	SPARC 32-bit	TIB_adjdexe-simple_<version_num>_s4_57.bin		51
Solaris 9				
Solaris 8			108940-33 108773-12 108652-37 109740-04 109742-04 109279-18 108921-13	
Solaris 7			107544-03 106980-17 106950-16 108376-33 106300-12 106327-11 106541-18 107081-40 107636-08 107226-18 108374-05 107656-07 107702-09 107153-01	
Linux 2.4 kernel:2.4.18 or above	x86	TIB_adjdexe-simple_<version_num>_lnx86_24.bin		

Supported J.D. Edwards Applications

For information on supported J.D. Edwards applications refer , Supported J.D. Edwards Applications, on page 31.

TIBCO Runtime Agent Must be Installed Before the Adapter

Before you can install the adapter, you must install the TIBCO Runtime Agent. If you use the Typical installation, the installer places all libraries and other products required by the adapter into the TIBCO HOME directory.

Installer Account

TIBCO 5.x products can be installed by a regular (non-root) user and super-user (root). Different users can install the same product at different locations.

Product dependencies at install time are resolved at user level through the installation registry maintained at user's home directory. See Installation Registry on page 26 for more information.

Windows Environment

A Windows environment such as CDE (that is, X Windows) is required to run the installer in GUI mode. It is not required for a console installation.

Installing the Adapter on UNIX

After unpacking the software and accepting the license agreement, you can choose to perform a typical install or custom install.

- A typical install has minimal prompts and installs standard components in default locations.
- A custom install prompts you to choose which pieces of the product suite to install and installs only those components.

The installer checks your system for the installation home directory that was established when TIBCO Runtime Agent was installed. The adapter is installed under the installation home directory.

Use one of the following modes to install the software. The examples assume you are installing the adapter on Solaris 2.8.

Install Using GUI Mode

GUI Mode allows you input values in panels. Type the following in a terminal window:

```
% ./TIB_adjdexe-simple_<version_num>_s4_58.bin
```

Install Using Console Mode

Console mode allows you to install the software in a non windows environment. The installer will prompt you for values. Type the following in a terminal window:

```
% ./TIB_adjdexe-simple_<version_num>_s4_58.bin -is:javaconsole
-console
```

When running in console mode you can move through the installation process as described next:

```
Enter Key = Moves forward in the installer
2 = Goes back to previous screen
3 = Cancels the Wizard and exits the installation or uninstallation
4 = Redisplays the current screen
```

Install Using Silent Mode with Default Values

Silent mode allows you to install the software without prompts using default values. Type the following in a terminal window:

```
% ./TIB_adjdexe-simple_<version_num>_s4_58.bin -silent
```

Install and Generate a Response File

You can generate a response file during installation which you can later use to invoke the installer with the selected values as default values (GUI mode) or as selected values (silent mode).

To install and generate a response file, type the following at the command prompt:

```
% ./TIB_adjdexe-simple_<version_num>_s4_58.bin -options-record
      /dir/<responseFile>
```

The response file does not record selections at the component level It does record all other selections, for example, which products you wished to install.

Install Using a Response File

You can use a previously generated response file for installation. For non-silent modes, the response file determine the defaults that are presented. For silent mode, the response file determines what will be installed.

To install using a response file, type the following at the command prompt:

```
% ./TIB_adjdexe-simple_<version_num>_s4_58.bin -options
/dir/<responseFileName>
```

Combining Options

You can combine the different available options. For example, to install in silent mode using a response file, use:

```
% ./TIB_adjdexe-simple_<version_num>_s4_58.bin -silent -options
<responseFileName>
```

To install using Console mode and generate a response file, use:

```
% ./TIB_adjdexe-simple_<version_num>_s4_58.bin -is:javaconsole
-console -options-record <responseFileName>
```

Post Installation

Setting Environment Variables

Certain environment variable settings are required for running the adapter. Each TIBCO product or product component contains a `setenv.sh` file in the `bin` directory which you must execute before using the product. The `setenv.sh` file adds information about the product component to the following environment variables on UNIX platforms:

```
PATH
LD_LIBRARY_PATH (Solaris)
LD_RUN_PATH (Solaris)
```

For example, before using TIBCO Designer, you must run the `setenv.sh` script in `$TIBCO_HOME/designer/<version_num>/bin`. Before using the adapter, you must run the `setenv.sh` script in that adapter's /bin folder.

Permission Requirements on UNIX Systems

All adapter users must have read, write, and execute permissions for the following directories:

```
$TIBCO_HOME/adapter/adjdexe/<version_num>/bin
$TIBCO_HOME/adapter/adjdexe/<version_num>/logs
$TIBCO_HOME/tra/<version_num>/logs
$TIBCO_HOME/logs
```

For example, if the adapter has been installed in `/opt/tibco`, the user who installed the adapter can make these directories writable for all other users by executing the following commands:

```
% chmod a+w /opt/tibco/adapter/adjdexe/<version_num>/bin/
% chmod a+w /opt/tibco/adapter/adjdexe/<version_num>/bin/logs
% chmod a+w /opt/tibco/tra/<version_num>/logs
```

Installation FAQs and Troubleshooting

This section lists some common errors along with their causes and solutions.

Frequently Asked Questions

Where is the installation log file located?

Install and uninstall log files are created in the `TIBCO_HOME\log` directory.

What should I do, if JVM crashes when I run the installer?

The installer first extracts the bundled JVM into a temporary area and then uses it to launch itself. It for some reason, the JVM crashes, you could still run the installer using another JVM, preferably JVM 1.5.0 or higher. The syntax is:

```
TIB_adjdexe-simple_<version_num>_w32.exe -is:javahome
C:\j2sdk1.5.0
TIB_adjdexe-simple_<version_num>_s4_58.bin -is:javahome
/opt/jre150
```

The `javahome` directory must contain `bin/java.exe` or `bin/java`.

The installer will use the externally supplied JRE to launch itself.

Will 5.3 installer recognize a 3.x or 4.x installation?

TIBCO products follow a three digit release numbering scheme:

Major.Minor.Maintenance

Product releases that differ in either Major or Minor numbers will be a separate installation, and will not recognize the old installation. In this case, 5.0 is a major release and hence will not recognize either 3.x or 4.x product installations.

Why and how should I set the DISPLAY variable on UNIX platforms for GUI mode?

The installer on UNIX, must open an additional window, generally for graphics. It uses the DISPLAY environment variable to tell it on what computer to open the window. If the environment variable is not set, the installer will either wait or abort after displaying:

```
InstallShield Wizard
Initializing InstallShield Wizard...
Preparing Java(tm) Virtual Machine...
```

```
.................................
.................................
........
```

The DISPLAY variable must be set to the IP address or name of the computer (on which the installer graphics window are to be displayed), followed by a screen address, which can be `:0.0` . For example:

```
# Bourne shell
DISPLAY=<ip_address>:0.0; export DISPLAY

# Korn shell
export DISPLAY=<ip_address>:0.0

# C-shell
setenv DISPLAY <ip_address>:0.0
```

For example, consider a scenario where you need to install the adapter on a remote HPUX machine (named `itaska`). Because you have a Solaris 5.6 machine (named `alaska`) that has a video card and monitor installed, you can run an X-window application on it. So you decide to telnet to `itaska` from `alaska`.

When you telnet to `itaska`, you will not get access to `itaska`'s monitor and will be unable to display an X-window application. That is why you must set the `DISPLAY` variable, which instructs the X-server to redirect all windows to the computer set in the variable. Before doing so, the computer (specified in the `DISPLAY` variable) must give permissions to share its monitor.

```
alaska> xhost + # give permission for all to its share monitor
alaska> telnet itaska
Welcome to HPUX itaska 11.00
User:
Password:
itaska> export DISPLAY=alaska:0.0  # set display on alaska
itaska> TIB_adjdexe-simple_<version_num>_odbc_h7_110.bin
```

What is uninst2 directory?

If the original uninstall directory is in use at uninstall time, it cannot be removed by the installer program. The installer then creates a second uninstall directory for the second installation. To remove the second installation, you must invoke the uninstall program from the second uninstall directory. The original uninstall directory can also be manually removed, if empty.

Running Out of Disk Space

The installer calculates the disk space required in product home location, for the selected components. The calculation is done before the actual installation (copying of files to system) begins. The installer will proceed only if sufficient free disk space is available in product home location.

However, if disk space is consumed by another process while the installer is copying the files, and if the required disk space is thereby reduced, the installer may fail and will then give a failure message.

Solution

While performing installation, avoid running other processes that consume disk space in product home location.

Configuring TIBCO Hawk

Error

TIBCO Runtime Agent includes the TIBCO Hawk Agent only. If you install the full TIBCO Hawk package after installing TIBCO Runtime Agent and do not have a Java Runtime Environment (other then the TIBCO JRE) installed, the TIBCO Hawk Configuration tool is unable to determine the Java home location and the JVM executable. The TIBCO Hawk services will not start correctly and you will be unable to start the TIBCO Hawk Display.

Resolution

1. Start the TIBCO Hawk Configuration tool. For example, on Microsoft Windows:

   Start>TIBCO>TIBCO Hawk>Hawk Configuration

2. Under the `General` tab, click **Advanced**.
3. In the `Java Home Directory` field, provide the path to Java. For example:

   C:\tibco\jre\1.5.0

4. In the `JVM Executable` field, provide the JVM executable. For example:

   java.exe

The services will start properly and the TIBCO Hawk Display will run.

Cannot Install the Adapter

On HP-UX and AIX platforms, even though the correct version of TIBCO Runtime Agent version is already installed on the system, installation of an adapter that depends on TIBCO Runtime Agent may fail in the dependency resolution.

The TIBCO product installer maintains the registry information in the `vpd.properties.tibco.`*systemName* file. The value for *systemName* is determined by `InetAddress.getLocalHost().getHostName()`. However, the method `getHostName()`, returns different values based on the JRE versions used. For example, on AIX, JRE 1.4.2 returns only *systemName*, whereas JRE 1.4.0 returns *systemName.domainName*. Because of this, the installer is not able to load the correct registry file.

Resolution

On UNIX platforms, the installer registry file `vpd.properties.tibco.`*systemName* is located in the user's home directory.

Case 1: If the `vpd.properties.tibco.`*systemName* file exists:

```
$ cd user's_home_directory
$ ln -s vpd.properties.tibco.systemName
vpd.properties.tibco.systemName.domainName
```

For example:

```
$ cd ~
$ ln -s vpd.properties.tibco.upside
vpd.properties.tibco.upside.tibco.com
```

where upside is *systemName*, and tibco.com is *domainName*

Case 2: If the `vpd.properties.tibco.`*systemName.domainName* file exists:

```
$ cd user's_home_directory
$ ln -s vpd.properties.tibco.systemName.domainName
vpd.properties.tibco.systemName
```

For example:

```
$ cd ~
$ ln -s vpd.properties.tibco.upside.tibco.com
vpd.properties.tibco.upside
```

where upside is *systemName*, and tibco.com is *domainName.*

Chapter 3

Preparing J. D. Edwards OneWorld Xe Application Interfaces

This chapter explains how to configure the J. D. Edwards OneWorld Xe application so that the adapter can interoperate with it.

Topics

Overview

Before configuring the adapter you must prepare the interfaces of the J. D. Edwards OneWorld Xe application, so the adapter can interoperate with it.

To use the adapter with J. D. Edwards OneWorld Xe, you must use the J. D. Edwards OneWorld Xe thin or fat client and connect to the J. D. Edwards OneWorld Xe enterprise server.

When specifying connection parameters for the adapter, please make sure that the adapter connects to the J. D. Edwards OneWorld Xe enterprise server and NOT the deployment server.

To configure J. D. Edwards OneWorld Xe for use with the adapter, do the configurations as described below.

- For Business Function calls and XML Query list, configuring the J. D. Edwards OneWorld Xe ini file is the only step required to configure J. D. Edwards OneWorld Xe for use with the adapter. For further information, see Configuring for Business Function Calls on page 47 and Configuring for XML Query Calls on page 48.
- For Real-Time events, you must configure the J. D. Edwards OneWorld Xe ini file. After the configuration of the ini file is complete, the real-time events must be defined using J. D. Edwards OneWorld Xe application interfaces. For further information, see Configuring for Real-Time Events on page 49.
- For PeopleSoft EnterpriseOne 8.11 and higher with PeopleTools 8.96, set up the Transaction Server and configure appropriate entries in the `.ini` file. For more information on how to configure the Transaction Server, refer to the Oracle PeopleSoft user documentation.

The requirements of each enterprise are unique. The configuration settings described in this manual are generalized guidelines, and may or may not suit the requirements of your enterprise.

J. D. Edwards may have changed the configuration settings from the time of printing this manual. Refer to the J. D. Edwards OneWorld Xe documentation for up-to-date information.

Configuring for Business Function Calls

For Business Function calls, you must configure the J. D. Edwards OneWorld Xe ini file, specifically, JDENET_KERNEL_DEF6 as in the example below.

Some configurations vary according to the platform that the server is on. These variations are noted below.

For an Enterprise Server Running on an IBM AS/400 Platform

```
[JDENET_KERNEL_DEF6]
krnlName=CALL OBJECT KERNEL
dispatchDLLName=JDEKRNL
dispatchDLLFunction=JDEK_DispatchCallObjectMessage
maxNumberOfProcesses=27
numberOfAutoStartProcesses=1
```

For an Enterprise Server Running on Sun Solaris

```
dispatchDLLName=libjdeknet.so
```

For an Enterprise Server Running on Microsoft Windows

```
dispatchDLLName=jdekrnl.dll
dispatchDLLFunction=_JDEK_DispatchCallObjectMessage@28
```

After making these changes to the JDE.INI file, you must restart the J. D. Edwards OneWorld Xe services. Rebooting the machine is not required.

Configuring for XML Query Calls

For XML Query calls, you must configure the J. D. Edwards OneWorld Xe ini file.

You must configure JDENET_KERNEL_DEF16 in the J. D. Edwards OneWorld Xe ini file, as in the example below.

Some configurations vary according to the platform that the server is on. These variations are noted below.

For an Enterprise Server Running on an IBM AS/400 Platform

```
[JDENET_KERNEL_DEF16]
krnlName=XML List Kernel
dispatchDLLName=XMLLIST
dispatchDLLFunction=XMLListDispatch
maxNumberOfProcesses=3
beginningMsgTypeRange=5257
endingMsgTypeRange=5512
newProcessThresholdRequest=0
numberOfAutoStartProcesses=1

[LREngine]
System=/OneWorld/export
```

For an Enterprise Server Running on Sun Solaris

```
dispatchDLLName=libxmllist.so
```

For an Enterprise Server Running on Microsoft Windows

```
dispatchDLLName=xmllist.dll
dispatchDLLFunction=_XMLListDispatch@28
```

After making these changes, please make sure that the parameter MaxKernalRanges in the JDENET section is set to 16 or greater.

Configuring for Real-Time Events

The list of pre-configured real-time events depends on the update level of J. D. Edwards OneWorld Xe application. With each new update level, J. D. Edwards OneWorld Xe adds new real-time events for business transactions.

However, due to customizations in the base application, if an update cannot be applied, applications and business functions can be modified as required to enable real-time events for business transactions. Please refer to J. D. Edwards OneWorld Xe documentation for further details.

To use the adapter with J. D. Edwards OneWorld Xe software for Real-Time Event publication, you must first configure the .ini file. After configuring the .ini file, you must do some further configuration in the adapter.

Configuring the .ini File

Some configurations vary according to the platform that the server is on. These variations are noted below.

To configure J. D. Edwards OneWorld Xe for use with Real-Time Event Publication, configure `JDENET_KERNEL_DEF19`, `JDENET_KERNEL_DEF20`, `JDEITDRV`, and `INTEROPERABILITY` as in the example below.

For an Enterprise Server Running on an IBM AS/400 Platform

```
[JDENET_KERNEL_DEF19]
krnlName=EVN KERNEL
dispatchDLLName=JDEIE
dispatchDLLFunction=JDEK_DispatchITMessage
maxNumberOfProcesses=1
numberOfAutoStartProcesses=1
```

For an Enterprise Server Running on Sun Solaris

```
dispatchDLLName=libjdeie.so
```

For an Enterprise Server Running on Microsoft Windows

```
dispatchDLLName=jdeie.dll
dispatchDLLFunction=_JDEK_DispatchITMessage@28
```

For an Enterprise Server Running on an IBM AS/400 Platform

```
[JDENET_KERNEL_DEF20]
krnlName=IEO KERNEL
dispatchDLLName=JDEIEO
dispatchDLLFunction=JDEK_DispatchIEOMessage
```

```
maxNumberOfProcesses=1
numberOfAutoStartProcesses=1
```

For an Enterprise Server Running on Sun Solaris

```
dispatchDLLName=libjdeieo.so
```

For an Enterprise Server Running on Microsoft Windows

```
dispatchDLLName=jdeieo.dll
dispatchDLLFunction=_JDEK_DispatchIEOMessage@28
```

For an Enterprise Server Running on an IBM AS/400 Platform

```
[JDEITDRV]
DrvCount=3
Drv1=Z:ZDRV
Drv2=RT:RTDRV
Drv3=JDENET:JDETRDRV
```

For an Enterprise Server Running on Sun Solaris

```
Drv1=Z:libzdrv.so
Drv2=RT:librtdrv.so
Drv3=JDENET:libjdetrdrv.so
```

For an Enterprise Server Running on Microsoft Windows

```
Drv1=Z:zdrv.dll
Drv2=RT:rtdrv.dll
Drv3=JDENET:jdetrdrv.dll
```

```
[INTEROPERABILITY]
LEVEL=TRACE,EVENTS
TRACE=TRUE
RegisteredEvents=<list of real-time events separated by commas>
FilteredEvents=*ALL
TempFileDir=<directory path for temporary XML files>
```

After making these changes, please make sure that the parameter `MaxKernalRanges` in the `JDENET` section is set to 20 or greater.

After making these changes to the `JDE.INI` file, you must restart the J. D. Edwards OneWorld Xe services. Rebooting the machine is not required.

Configuring the Application

For information on how to configure a J.D. Edwards application, refer the respective J.D. Edwards application's product documentation.

Chapter 4 Getting Started

This chapter presents examples that demonstrate key adapter features. Work through these examples to get a hands-on understanding of how the adapter works.

Topics

Prerequisites

Before starting the configuration exercise, ensure that all required software has been installed and is operating correctly. For a list of required software, see the installation instructions in Chapter 2, Installation, on page 15.

You should know how to drag and drop icons in TIBCO Designer and be familiar with saving projects. If you are not familiar with these topics, refer to the TIBCO Designer User's Guide, which is available by clicking **Help>Designer Help** in TIBCO Designer.

Scenario Overview

This chapter provides a set of short exercises that show you how to configure the adapter with publication, subscription, and request-response services.

In the following exercises, you will configure different adapter services to inter-operate with J.D. Edwards OneWorld Xe. Details for the services that will be configured are as follows:

- A subscription service to create a sales order in J.D. Edwards OneWorld Xe. After creating this subscription service, you will create a TIBCO BusinessWorks process to use this subscription service and create a sales order in J.D. Edwards OneWorld Xe.
- A publication service to listen for outbound events for sales order transactions taking place in J.D. Edwards OneWorld Xe. After creating this publication service, you will create a TIBCO BusinessWorks process to use this publication service and extract the sales order number.
- A request-response server service for a business function. The business function for this service calculates and returns different quantities (on hand, soft committed, hard committed and so forth) when an item number and branch code are passed as input parameters. After creating this service, you will create a TIBCO BusinessWorks process to invoke it. In this TIBCO BusinessWorks process, you will pass parameter values for item numbers and branch codes and get the different quantities for the item.
- Request-response server services for an XML query list. These services will be used to select and extract data from a user-defined code table. After creating these services, you will create a TIBCO BusinessWorks process to use these services and test the results.

Key Concepts

There are several key concepts to understand in J.D. Edwards OneWorld Xe and TIBCO Adapter for J. D. Edwards OneWorld Xe.

- Business Functions: Business Functions are executable functions that can be used in any application of OneWorld. Business Functions allow you to reuse the same code in several applications. A typical Business Function might be a line item or an address in a purchase order.
- Business Function Groups: Business Function Groups are groups of Business Functions that are executed together. You set up these groups within the TIBCO Adapter for J. D. Edwards OneWorld Xe. A typical Business Function group might be a complete purchase order, including header information, customer information and line items.

  A Business Function group encapsulates the logic for the creation of one Business entity in the J.D. Edwards OneWorld Xe system. The Business Functions contained in the Business Function Group or their invocation order could be different in each deployment.
- XML Tables: These data tables exist in J.D. Edwards OneWorld Xe and can be queried by the adapter. The data in these tables may or may not correspond to a given Business Function.
- Table Conversion: Table conversion is a special form of Universal Batch Engine (UBE) that enables you to do a high-speed manipulation of data in tables. You can use table conversions to gather, format, import, and export data. The adapter can run and query the table conversion output using XML Query List.
- Real-Time Events: These are events that happen in the J.D. Edwards OneWorld Xe system or another application, and are communicated between systems by the adapter, which listens for and publishes events. A real-time event might be the issuing of a certain purchase order.

Connecting to J.D. Edwards OneWorld Xe

To run the adapter, you must ensure that the J.D. Edwards OneWorld Xe system has been prepared to allow the adapter to connect to it. See Chapter 3, Preparing J. D. Edwards OneWorld Xe Application Interfaces, on page 45 for details.

Later in the process, you must validate the connectivity through the palette by clicking on the **Test Connection Button**. See Configure the Adapter on page 59 for details.

Other Examples

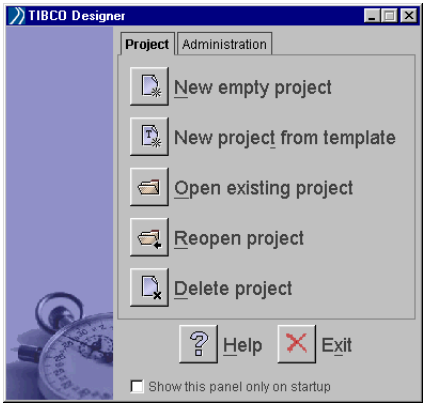
A set of examples are included online in the *<TIBCO HOME>* `\adapter\adjdexe\<version_num>\examples` directory. Examples are included for each type of service that the adapter can run. The repository is `Samples.dat`

Create the Project

The TIBCO Designer GUI is used to configure adapter instances. When starting TIBCO Designer, you must create or select a project. A project contains the configuration files and schema classes used by a run-time adapter. After a project is configured, it is exported to a repository file and available for use by the run-time adapter.

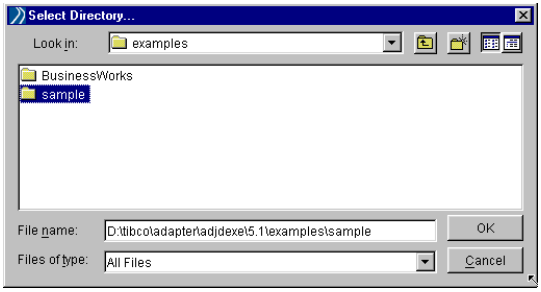
To create a project:

1. Start TIBCO Designer by executing the following command, depending on your operating system.
   - On Microsoft Windows, select: **Start>Programs>TIBCO>TIBCO Designer <version_num>Designer <version_num>**
   - On Unix, from a command window change directory to the *<install-path>*/tibco/`designer/<version_num>/bin` directory and type **./designer**
2. In the TIBCO Designer dialog, click **New empty project**.



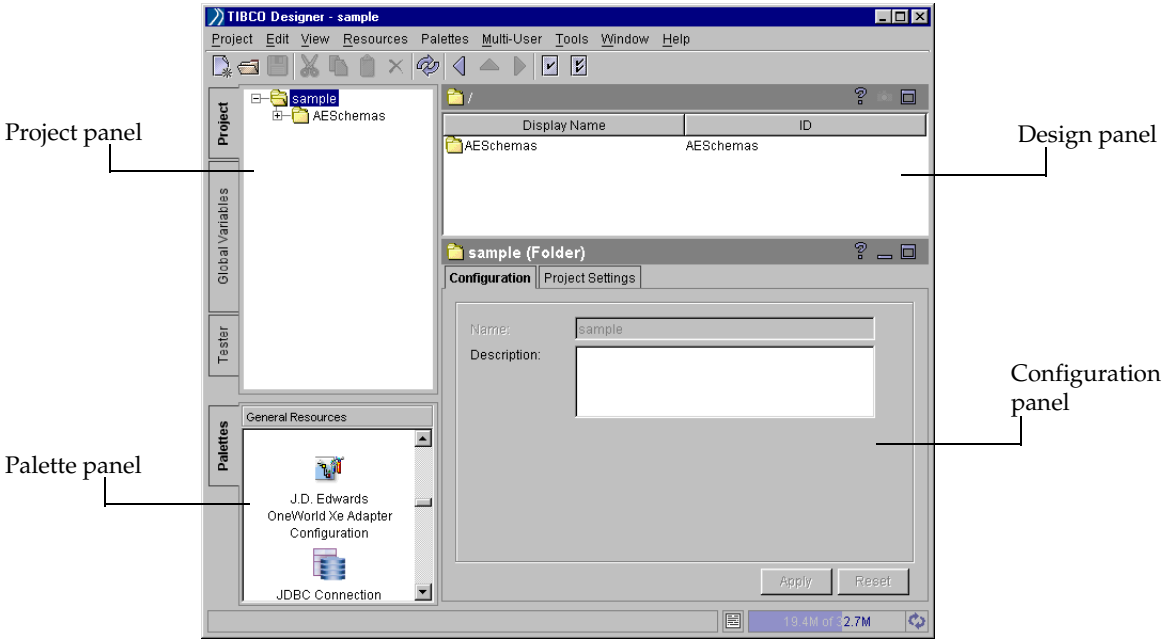
3. In the Save Project dialog, select `Multi-file Project` if it is not already selected, and click the **Browse** button. Navigate to the `C:\tibco\adapter\adjdexe\<version_num>\examples` folder and click the

`Create New Folder` icon. Name the new directory **sample** and click **OK**. The project files for the example are saved in the `sample` directory.



4. In the `Save Project` dialog, click **OK**.

   The next diagram shows the TIBCO Designer GUI with the `sample` project defined.

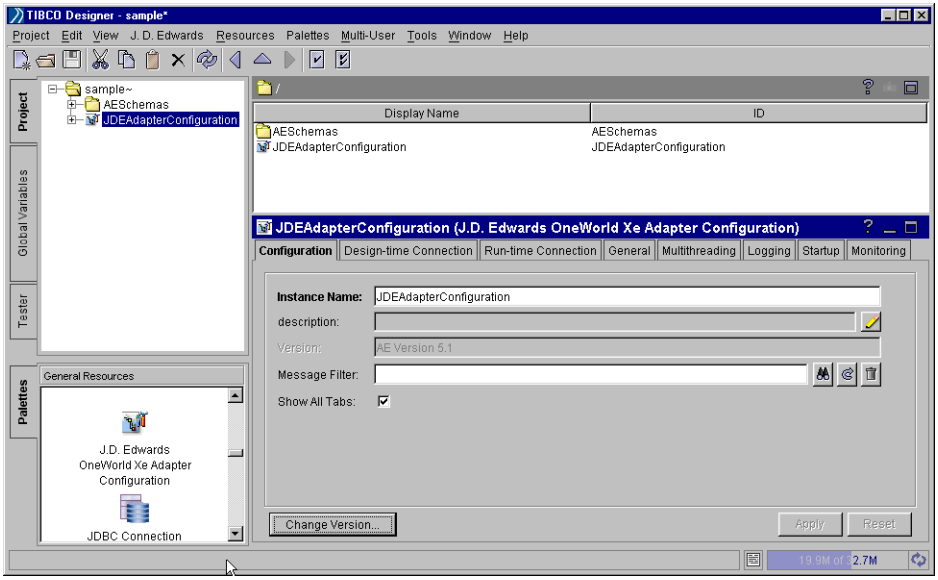


Configure the Adapter

An adapter instance can contain publication services, subscription services, request-response services, or all three types in the same adapter instance. Options for logging, startup and monitoring are set for the adapter instance. In this exercise, default values are used for these options.

To configure the adapter instance:

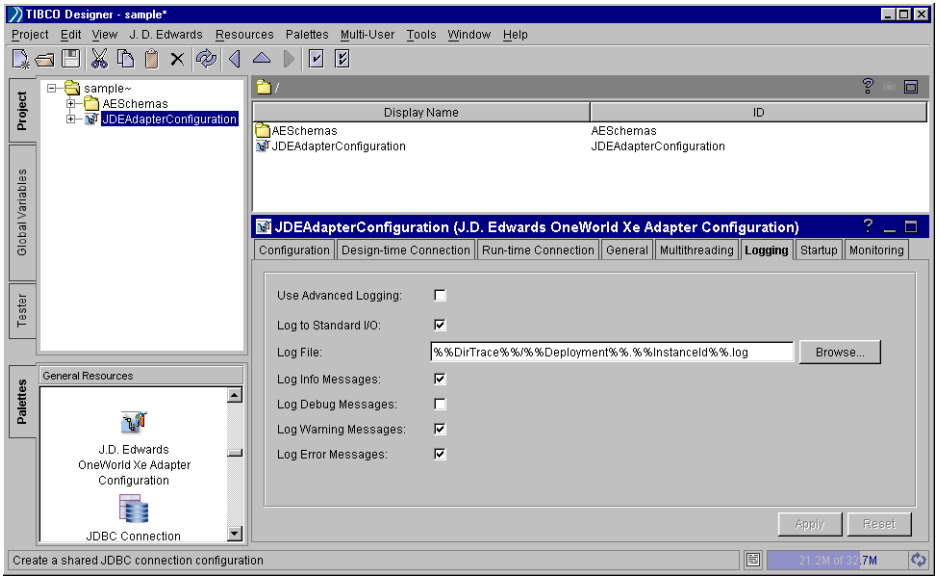
1. Drag the `J.D. Edwards OneWorld Xe Adapter Configuration` icon from the palettes panel to the design panel. This creates an adapter named by default, `JDEAdapterConfiguration`. The name can be changed, but in this exercise, names assigned by TIBCO Designer will be used.



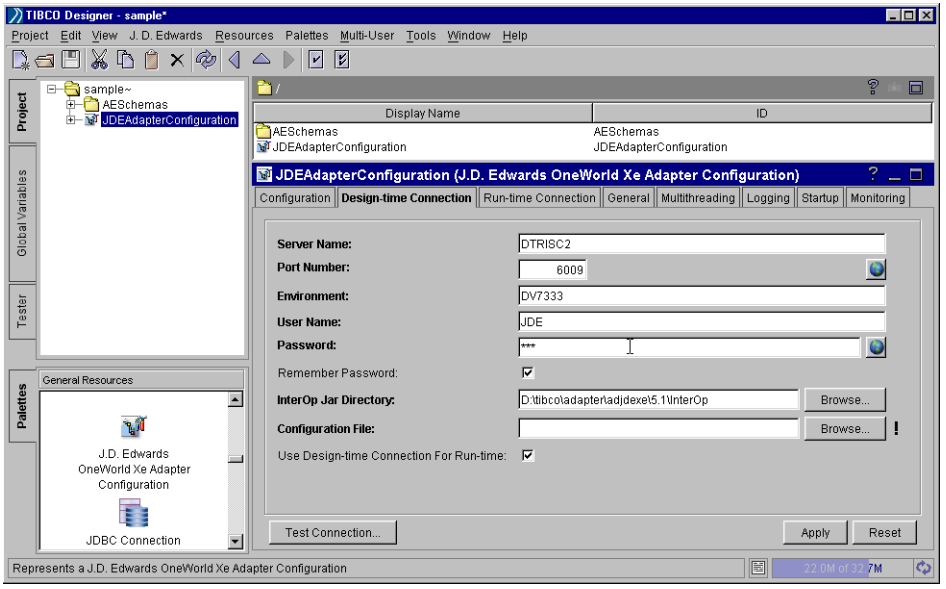
2. Click the `Logging` tab to identify the file log options.

   In the next diagram, Information, Warning and Error messages are specified to be logged to the log file and standard input. The `Log File` field lists the global variables that are used to define the log file path and name. The `DirTrace` and `Deployment` variables are set using global variables. You can click the `Global Variable` tab to display the variables in the project panel. The default settings will be used in this example. The `InstanceId` variable

need not be set. The variable automatically substitutes the adapter name at run-time.



3. Set the connection options in the `Design-time Connection` and `Run-time Connection` tabs.



These two tabs provide all the parameters necessary to establish a successful connection to the J.D. Edwards OneWorld Xe enterprise server. For more

information about connection parameters, see Design-time Connection Tab on page 123 and Run-time Connection Tab on page 125. After specifying the connection parameters in each tab, click **Apply**.

4. Click the **Test Connection** button to test the connectivity to the J.D. Edwards OneWorld Xe enterprise server. Connectivity with the enterprise server takes place in the background.

You must establish adapter connectivity with the J.D. Edwards OneWorld Xe enterprise server before you can configure any adapter services.

When you are working with these examples, or when you are specifying connection parameters for test cases, be sure to specify a J.D. Edwards OneWorld Xe **TEST** environment. If you work in the production environment, you will create new sales orders and so forth in the production environment, which may not be desirable.

The default settings will be used in this example. The `InstanceId` variable need not be set. The variable automatically substitutes the adapter name at run-time.

Configure the Subscription Service

This section explains how to configure an adapter instance with a subscription service that receives a sales order from an outside application. You must do the following:

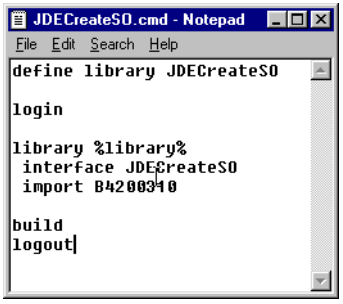
1. Create GenJava classes for Business Functions
2. Configure the Subscription Service
3. Create a TIBCO BusinessWorks Test Process to Use this Subscription Service
4. Run the Test Process and View Results

For information on building a TIBCO BusinessWorks test process, see TIBCO BusinessWorks documentation.

Task A Create GenJava classes for Business Functions

The process described in this section must be run on a J.D. Edwards OneWorld Xe client machine.

1. Create a new text file `JDECreateSO.cmd`. Save it in a directory on the J.D. Edwards OneWorld Xe client machine.
2. Type the contents of the text file as shown below.



3. Save the text file. This is the GenJava command file.
4. Start a Microsoft Windows command prompt. At the command prompt, change the directory to the directory where the GenJava command file is created. Then type the following command:

```
genjava /cmd JDECreateSO.cmd /out <GenJava Command File Directory>
```

The GenJava process will prompt you to log on to the J.D. Edwards OneWorld Xe server.

5. Log on by providing the J.D. Edwards OneWorld Xe user ID and password.

   After the process runs successfully, a file called `JDECreateSOInterop.jar` will be created in the GenJava Command file directory.

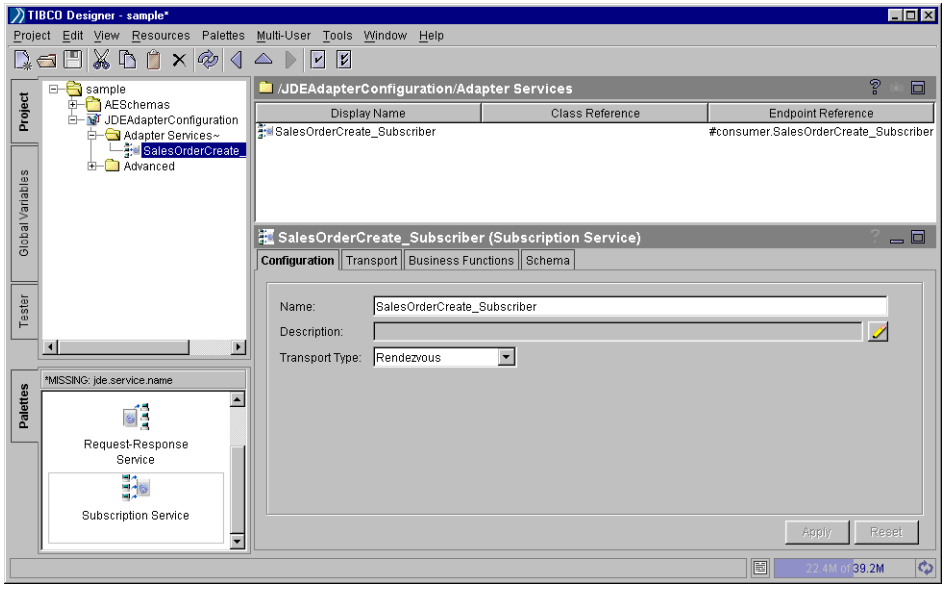
6. Copy this file to the machine where the TIBCO Adapter for J. D. Edwards OneWorld Xe will run.

   This file should be copied to the **Interop** directory as specified in the connection parameters for the adapter. For details, see Design-time Connection Tab on page 123.

Task B Configure the Subscription Service

1. In the project panel, expand the `JDEAdapterConfiguration` icon, then highlight the `Adapter Services` folder to access the `Subscription Service` icon. Drag the icon from the palette panel to the design panel.

2. In the `Configuration` tab, rename the subscription service to `SalesOrderCreate_Subscriber`. Click **Apply**.

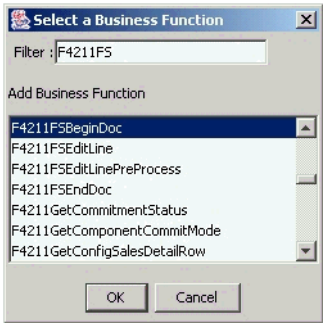


3. Click the `Business Functions` tab, choose `BusinessFunctionGroup` from the drop-down menu, and name the business function group.

   The name of the business function group can contain alphanumeric characters and can be representative of the operation. For example, a business function group for creating a sales order can be named `CreateSalesOrder`.

4. Click the `Add Business Function` icon. 

5. A list of business functions is displayed.



6. Choose one or more business functions, clicking **OK** after each addition. In this example, we have chosen `F4211FSBeginDoc`, `F4211FSEditLine`, `F4211FSEndDoc`, and `F4211FSClearWorkFile`.

The above example illustrates how multiple Business Functions can be called in a sequence.

In general, you use a Business Function Group to call a set of Business Functions in a sequence encapsulating logic to create a business entity. You pass output parameters from one Business Function invocation to the next Business Function in the sequence. The parameters that need to be passed from one Business Function invocation to the next, are indicated by the **Bind Identifier** column. Any parameter can become input to subsequent invocations of business functions by selecting the **Use Bind** column attribute for that parameter. Upon selecting **Use Bind**, the selected parameter becomes available in the **Bind Identifier** column for the succeeding business functions.

In the selected business functions, calling the business function for `F4211FSEditLine` creates a sales order line. A sales order could contain more than one sales order line. A definition for these business functions is below.

— `F4211FSBeginDoc` — Accepts data structure parameters for a sales order header, validates the values of these parameters and then creates a sales order header in the temporary work file cache.

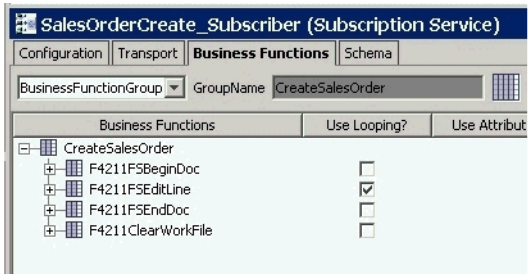
— `F4211FSEditLine` — Accepts data structure parameters for a sales order line, validates the values of these parameters and then creates a sales order line in temporary work file cache.

— `F4211FSEndDoc` — Copies the data from the temporary work file cache to the actual database tables once the creation of the sales order header and lines is completed successfully.

— `F4211ClearWorkFile` — Cleans the temporary work file cache for sales order header and lines.

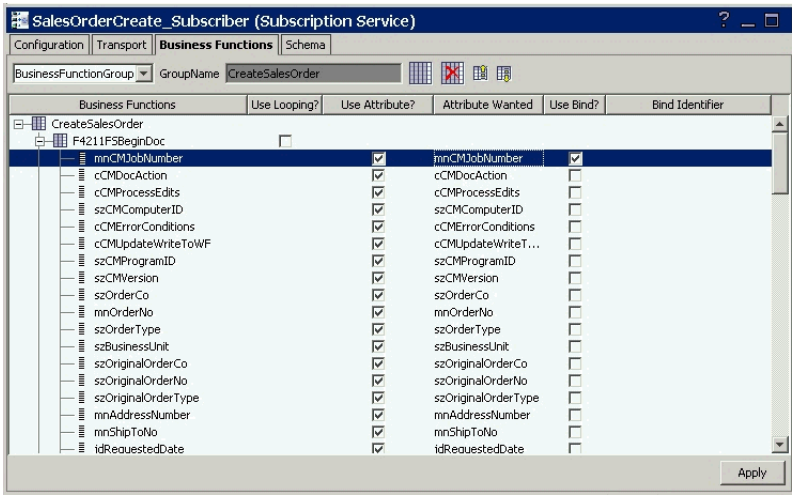
To indicate that a business function could be called multiple times in a business function group, select the **Use Looping?** column attribute. Looping is useful when repeated invocations of the same business function is required to accomplish a task. For example, while creating multiple order lines in a sales order.

7. Select the **Use Looping** check box for the business function `F4211FSEditLine`, since this function can be repeated.



8. Use the up and down arrows to rearrange the sequence of the business functions in the table. 

   The adapter provides a way to pass values returned by a business function as input parameters for subsequent business function calls. This feature is called binding of parameters.



   All the **Use Attribute** boxes are selected by default. Clear any attributes that you do not want to use.

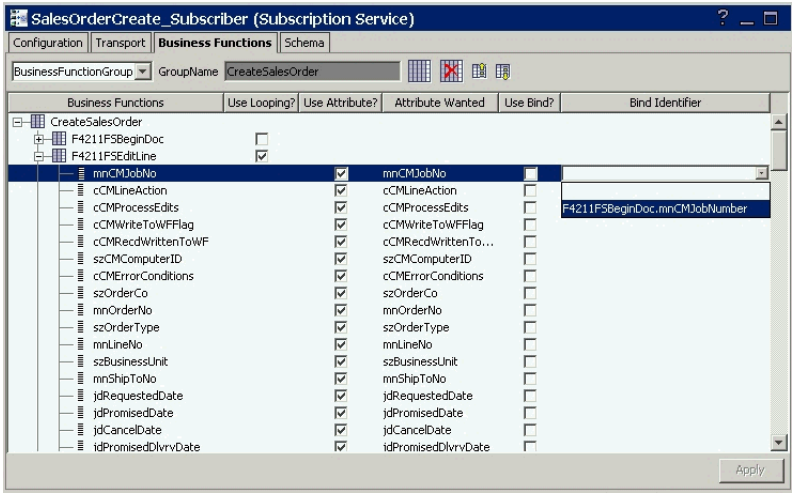
9. For the `CreateSalesOrder` subscriber, define a bind variable in business function `F4211FSBeginDoc`.

The value of this variable will be passed as input value to subsequent business functions: `F4211FSEditLine`, `F4211FSEndDoc`, and `F4211ClearWorkFile`.

Any attribute can be bound in one Business Function and therefore made available to subsequent Business Functions. The bound variables are not available for the preceding functions.

10. Expand the tree for the `F4211FSBeginDoc` business function.
11. Select a bind for `F4211FSBeginDoc` by selecting the **Use Bind** check box for the variable `mnCMJobNumber`.

    Now, use the value of the `F4211FSBeginDoc` function in the `F4211FSEditLine` business function.
12. Expand the tree for `F4211FSEditLine`. From the **Bind Identifier** drop-down box for the variable `mnCMJobNo`, select the bind variable `F4211FSBeginDoc.mnCMJobNumber`.



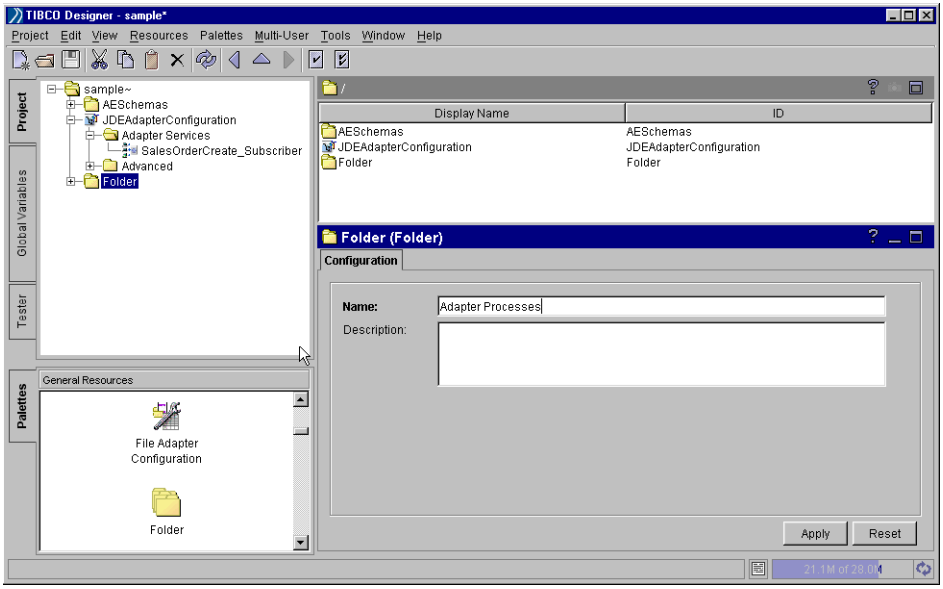
13. Similarly, to use the value of the bind variable in the business functions `F4211FSEndDoc` and `F4211ClearWorkFile`, repeat the step above, choosing variables `mnCMJobNo` and `mnJobNo` respectively.

    For more information on the columns for business function groups, see Business Functions Tab on page 140.

Task C Create a TIBCO BusinessWorks Test Process to Use this Subscription Service

You have configured a subscription service to create a sales order. Now you will use this service with TIBCO BusinessWorks.

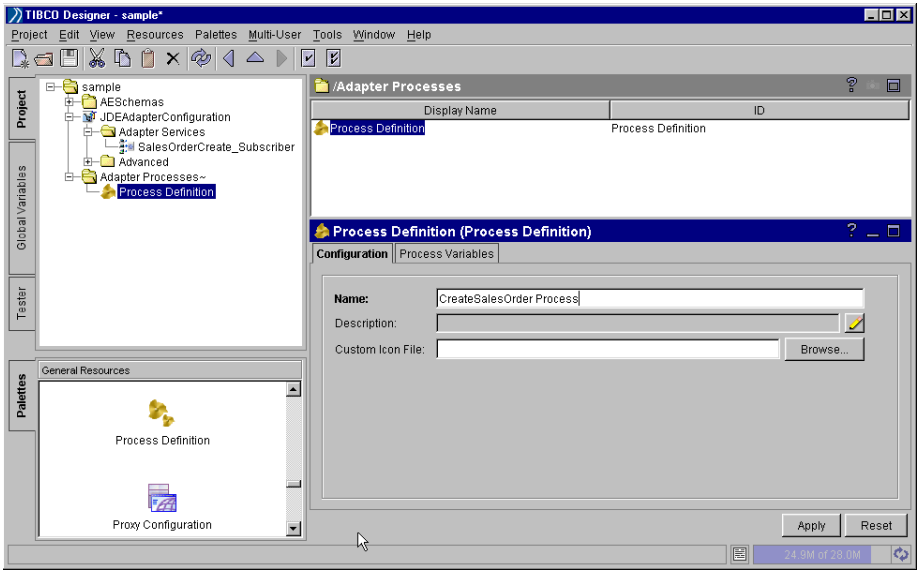
1. Select the project root in the project panel. In this example, the project is called `Sample`.
2. From the palette panel, drag and drop the **Folder** icon into the design panel. Select the newly created folder in the design panel.
3. In the `Configuration` tab, change the name of the folder to `Adapter Processes`. Click **Apply.**



4. Double-click `Adapter Processes` in the project panel.

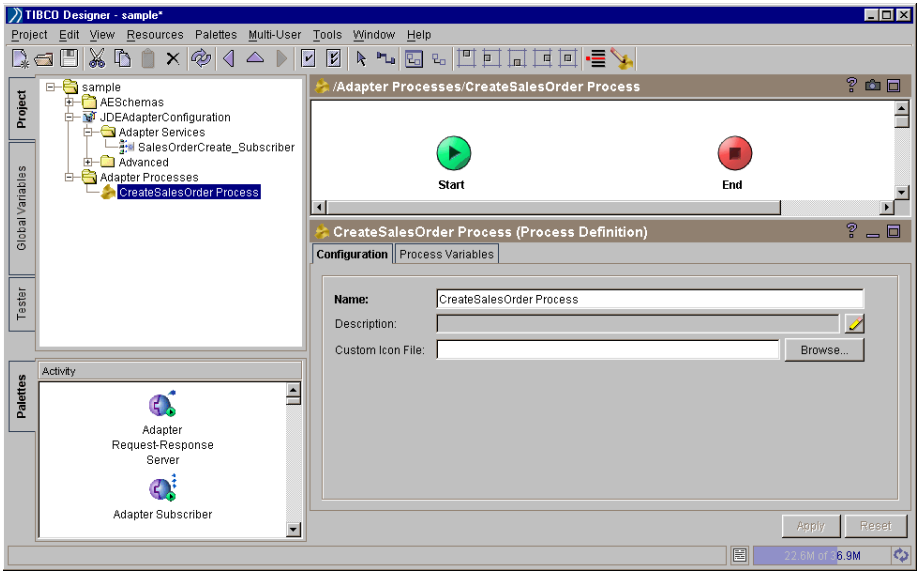
   The design panel area will now display as blank, and the icons in the palette panel will change.
5. From the palette panel, drag and drop the **Process Definition** icon to the design panel area.

6. Select the newly created process in the design panel. In the `Configuration` tab, change the name of the process to **CreateSalesOrder Process**. Click **Apply**.

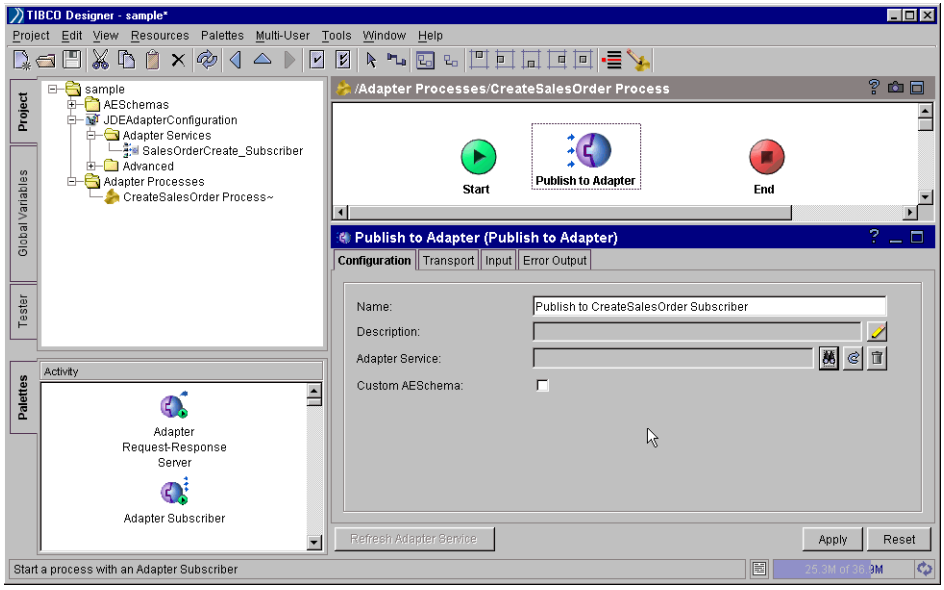


7. In the design panel, double click the process icon for **CreateSalesOrder Process**.

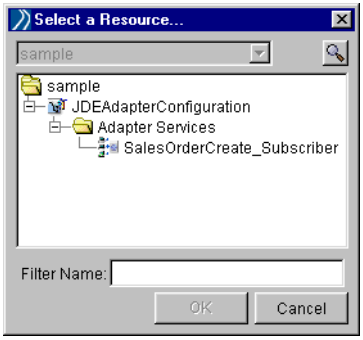
   The design panel will display an empty process with **Start** and **End** tasks and the icons in the palette panel will change.



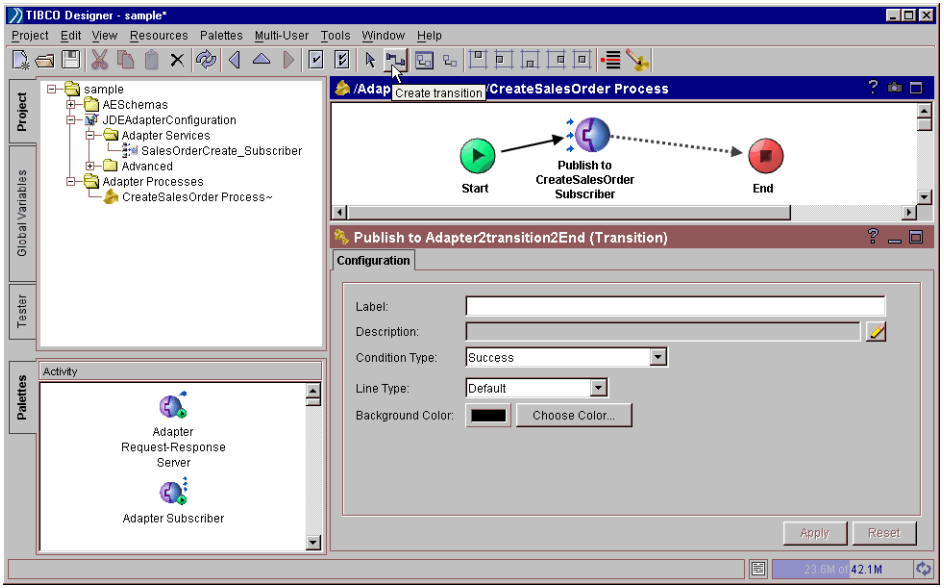
8. Drag and drop the **Publish to Adapter** icon from the palette panel to the design panel area, between the **Start** and **End** tasks.
9. Select the **Publish to Adapter** task in the design panel. In the `Configuration` tab, change the name of the task to **Publish to CreateSalesOrder Subscriber**.



10. Click the **Browse Resources** button near the Adapter Services text box. The Select a Resource dialog is displayed. Select the **CreateSalesOrder_Subscriber** service. Click **OK** to return to the previous dialog. Click **Apply**.



11. Click the **Create Transition** button in the TIBCO Designer toolbar. Connect the tasks in the process as shown:



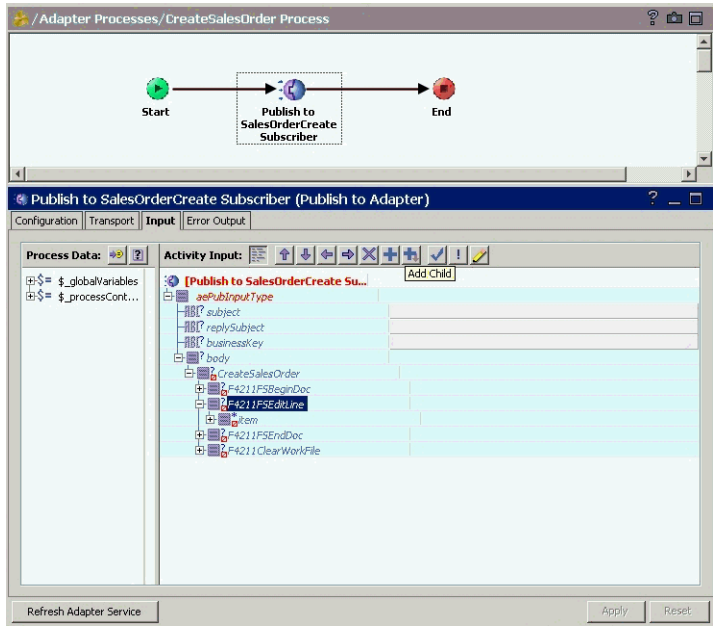
Since there could be more than one item in a sales order, the function for `F4211FSEditLine` was designated as **looping** when the subscription service was created. Now, in this test process, follow the steps below to create two sales order lines.

a. Click the **Publish to CreateSalesOrder Subscriber** task**.**

b. Select the **Input** tab.

c. Expand the **body** input element in the **Activity Input** panel.

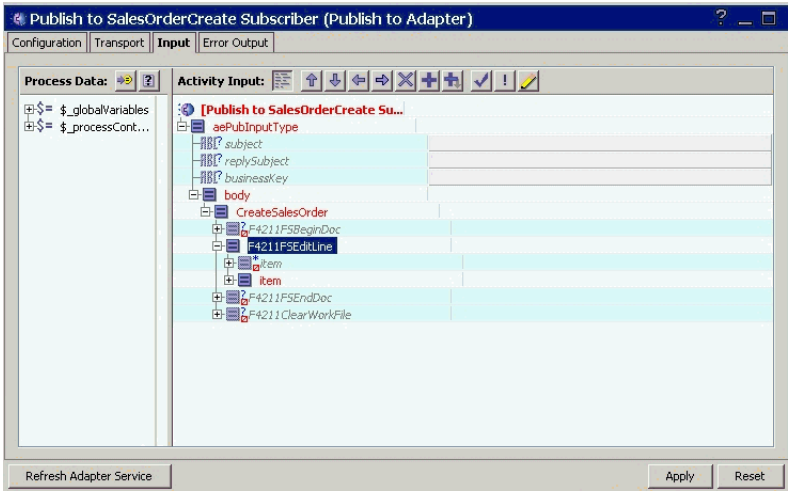
d. Select the data structure schema for the `F4211FSEditLine` business function and then click the **Add Child** button on the **Activity Input** toolbar.



A **New Statement** dialog is displayed.

e. From the **Statement Type** drop-down, select **Element**. Under **Representation**, select the **Literal** option. From the **Suggested** drop down box, select **item**. Click **OK**.

The **Activity Input** panel is displayed as shown:



12. Assign values to business function data structure schema parameters. The values to be assigned are listed below:

```
F4211FSBeginDoc
===============
cCMDocAction='A'
cCMProcessEdits='1'
szCMComputerID=<your machine name>
cCMUpdateWriteToWF='2'
szCMProgramID='TIBCO'
szCMVersion='ZJDE0001'
szOrderType='SO'
szBusinessUnit=<business unit> (right adjusted)
mnAddressNumber=<customer number>
cApplyFreightYN='Y'
mnWKSuppressProcess=0

F4211FSEditLine
==============
cCMLineAction='A'
cCMProcessEdits='1'
cCMWriteToWFFlag='2'
szCMComputerID=<your machine name>
szItemNo=<item number 1>
mnQtyOrdered=<quantity ordered>
szTransactionUOM='EA'
szCMProgramID='TIBCO'
szCMVersion='ZJDE0001'

F4211FSEndDoc
===============
szCMComputerID=<your machine name>
szProgramID='TIBCO'
szCMVersion='ZJDE0001'
cCMUseWorkFiles='2'

F4211ClearWorkFile
==================
szComputerID=<your machine name>
cClearHeaderWF='2'
cClearDetailWF='2'
szProgramID='TIBCO'
szCMVersion='ZJDE0001'
```

13. Click **Apply**.
14. Select **Project>Save**.

Task D Run the Test Process and View Results

In Tasks A and B you created an adapter subscription service. In Task C you configured a process flow and used this service in it.

In this task, you will test the process and adapter service interaction by running the process diagram in the debug mode from within TIBCO Designer.

Once configured, the process can be run on the command line by running TIBCO BusinessWorks in the engine mode. For more information, see TIBCO BusinessWorks documentation.

1. In TIBCO Designer, select **Tools>Show Adapter Tester**. The Adapter Tester is displayed. Click `JDEAdapterConfiguration` in the project panel. Specify the **Working Directory** and the **Adapter Executable**. Click **Start** to run the adapter.
2. Open the **CreateSalesOrder Process** process definition created in the previous section. If you are continuing from the last section, the process definition is already open.
3. Click the **Tester** tab in TIBCO Designer and click the **Start testing viewed process** button in the **Tester** panel toolbar, or select **Tools>Tester>Start**. The Select Processes to Load dialog is displayed. Click **Start Selected**.

   After the process runs successfully, the connecting arrows in the process turn green.
4. Using a J.D. Edwards OneWorld Xe client, log on to J.D. Edwards OneWorld Xe.
   a. Confirm that a new order was created in J.D. Edwards OneWorld Xe for the customer number that was used in the `F4211FSBeginDoc` business function.
   b. Check that the data for the order is the same as that provided in **CreateSalesOrder Process**.
5. In the Adapter Tester window, click the **Stop** button to stop the adapter.

Configure the Publication Service

This section explains how to configure a publication service to publish sales orders to other applications. The exercise assumes you have completed the previous exercise, Configure the Subscription Service on page 62. Use the same project created in the previous exercise. To complete this exercise, you must complete the following tasks:

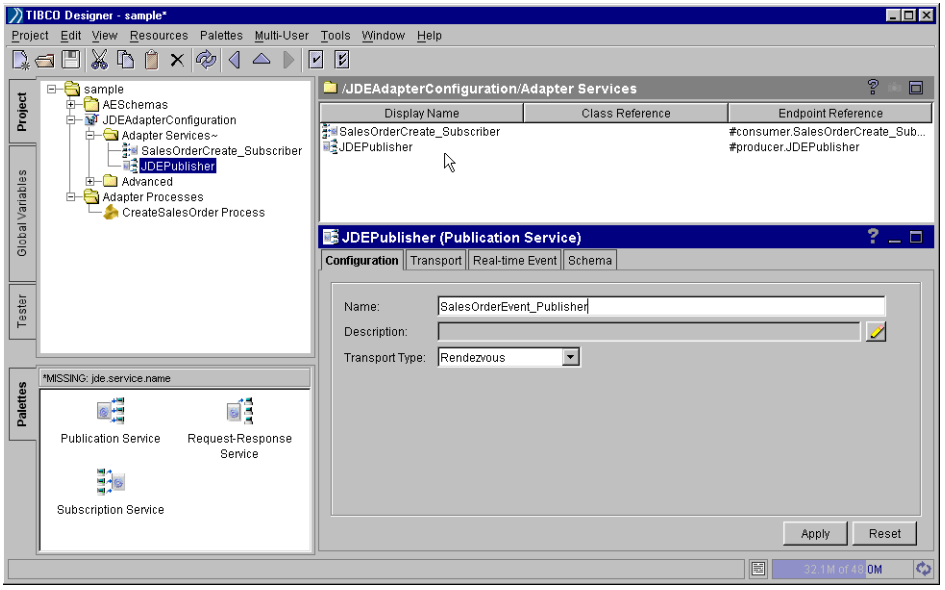
1. Configure the Publication Service
2. Create a TIBCO BusinessWorks Test Process to Use this Publication Service
3. Run the Test Process and View Results.

Task A Configure the Publication Service

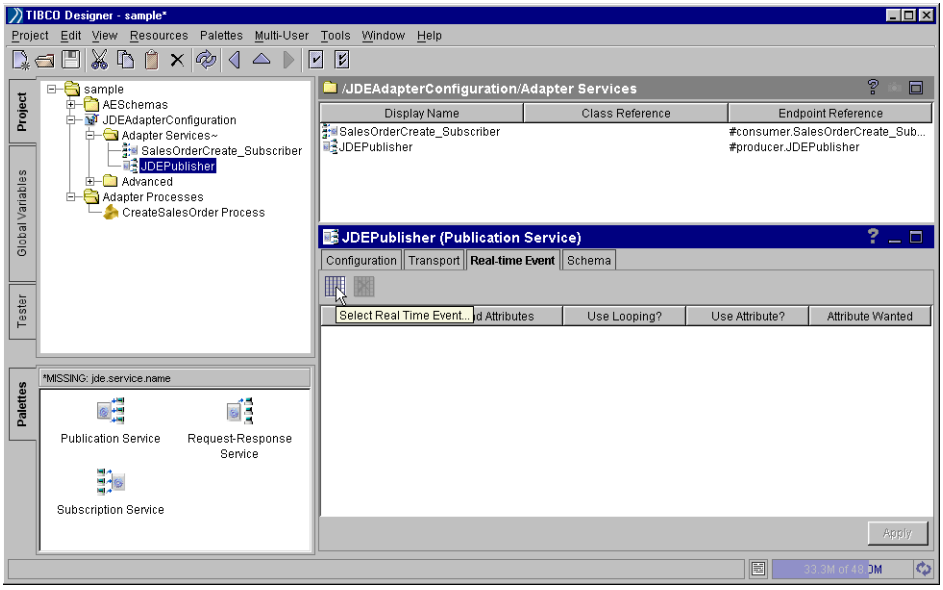
1. In the project panel, expand the `JDEAdapter` icon. The adapter services that are already configured are displayed. Select the `Adapter Services` folder to access the service icons. Drag and drop the `Publication Service` icon from the palette panel to the design panel.

   A new publication service `JDEPublisher` is created.

2. In the `Configuration` tab, rename the service to `SalesOrderEvent_Publisher`. Click **Apply**.

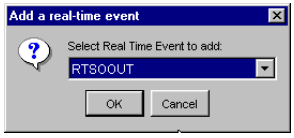


3. Select the `Real-time Event` tab, then click the **Select Real Time Event** button.



The Add a Real-time Event dialog, containing the list of all available real-time events is displayed.

4. Select the `RTSOOUT` event. This is the real-time event for sales order transactions. Click **OK**.



The `Real-time Event` tab is updated with the real-time event tree, for the event you have selected.

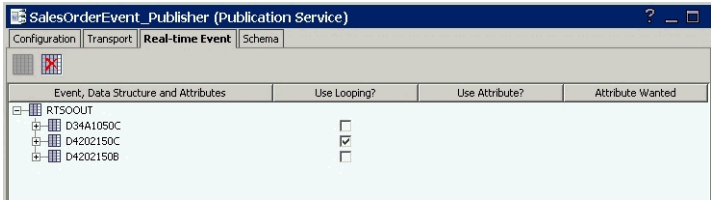
5. Expand the tree for the `RTSOOUT` event. The data structures for that event are displayed.

   You have chosen the event for a sales order transaction, and a sales order can have multiple order lines.

   To indicate that a business function could be called multiple times in a business function group, select the **Use Looping** column attribute. Looping is useful when repeated invocations of the same business function is required to accomplish a task. For example, while creating multiple order lines in a sales order.

6. Select the **Use Looping** check box for the `D4202150C` data structure. Click **Apply**.
7. Select **Project>Save**.

   The figure below shows the service with the event added.



Task B Create a TIBCO BusinessWorks Test Process to Use this Publication Service

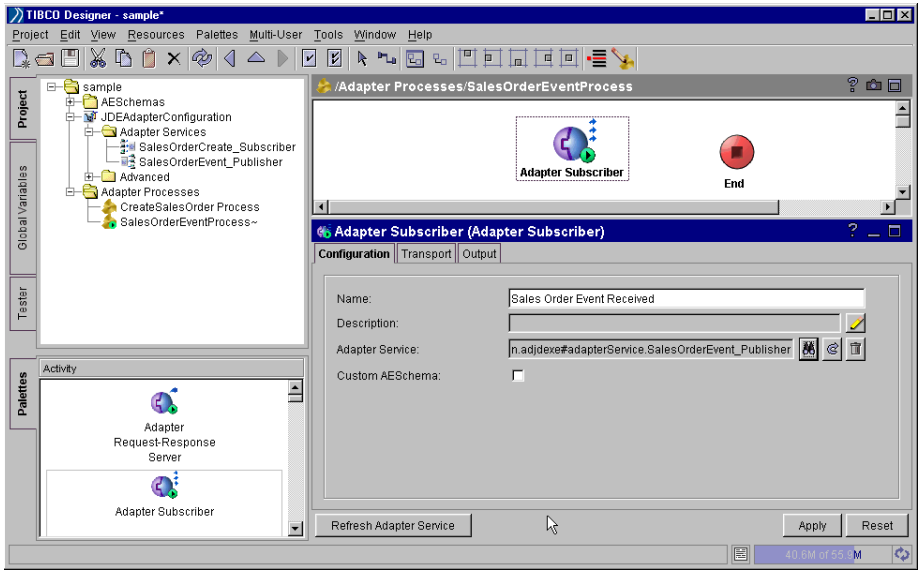
You have configured a publication service to publish a real-time sales order event. Now you will use this service with TIBCO BusinessWorks.

1. Double click the `Adapter Processes` folder in the project panel. The design panel area displays as blank, and the icons in the palette panel changes.
2. From the palette panel, drag and drop the `Process Definition` icon to the design panel area.
3. Select the newly created process in the design panel. In the `Configuration` tab, change the name of the process to **SalesOrderEventProcess**. Click **Apply**.
4. In the design panel, double click the process icon for **SalesOrderEventProcess**.

   The design panel will display an empty process with **Start** and **End** tasks and the icons in the palette panel will change.
5. From the palette panel, drag the **Adapter Subscriber** icon to the design panel area, between the **Start** and **End** tasks.

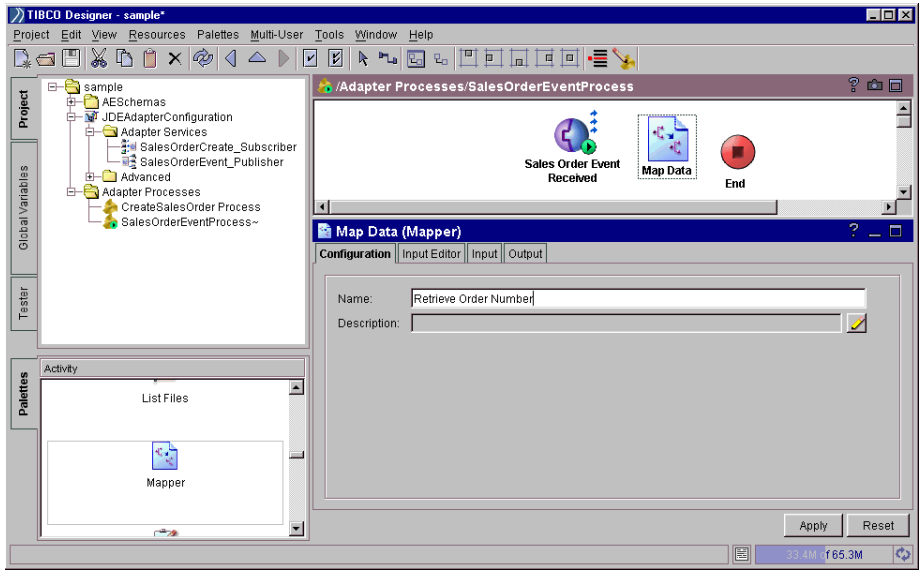
   The **Start** task will be replaced by the **Adapter Subscriber** icon.
6. Select the **Adapter Subscriber** icon in the design panel. In the `Configuration` tab, change the name of the task (in the Name field) to **Sales Order Event Received**.

7. Click the **Browse Resources** button near the Adapter Services text box. The Select a Resource dialog for selecting adapter services is displayed. Select the **SalesOrderEvent_Publisher** service. Click **OK**. Click **Apply**.

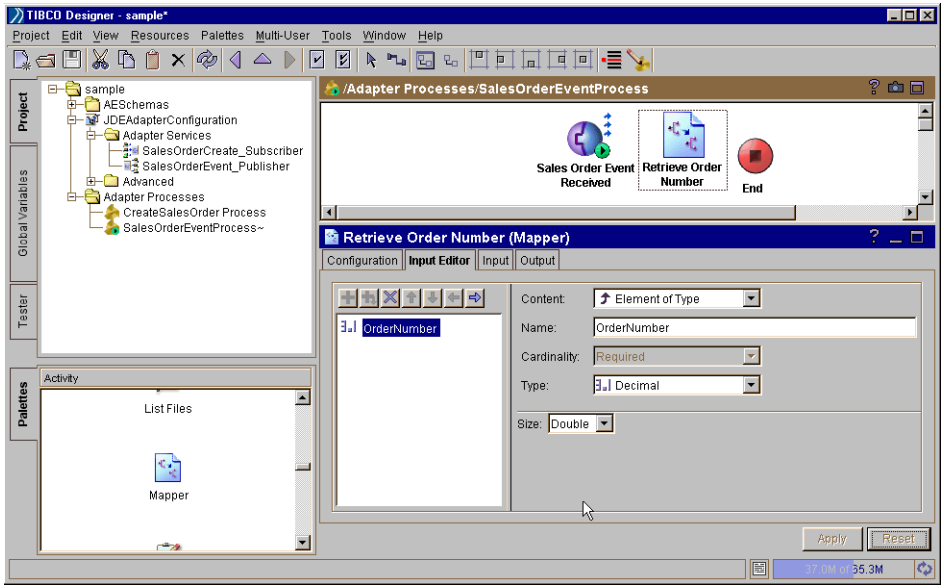


8. From the palette panel, select the `Mapper` icon and drag and drop it to the design panel area, between the **Adapter Subscriber** and **End** icons.

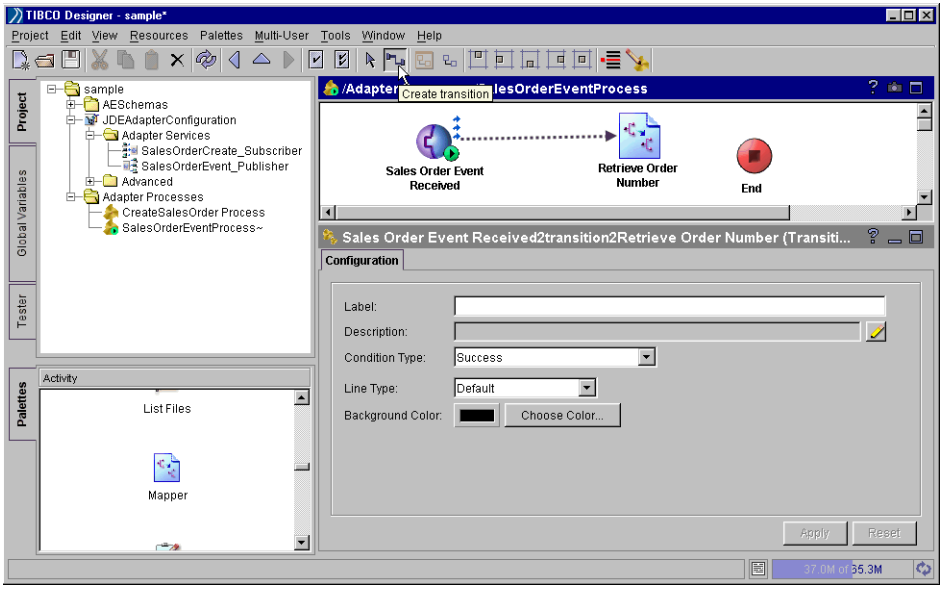
9. Select the **Mapper** task in the design panel. In the `Configuration` tab, change the name of the task to **Retrieve Order Number**. Click **Apply**.



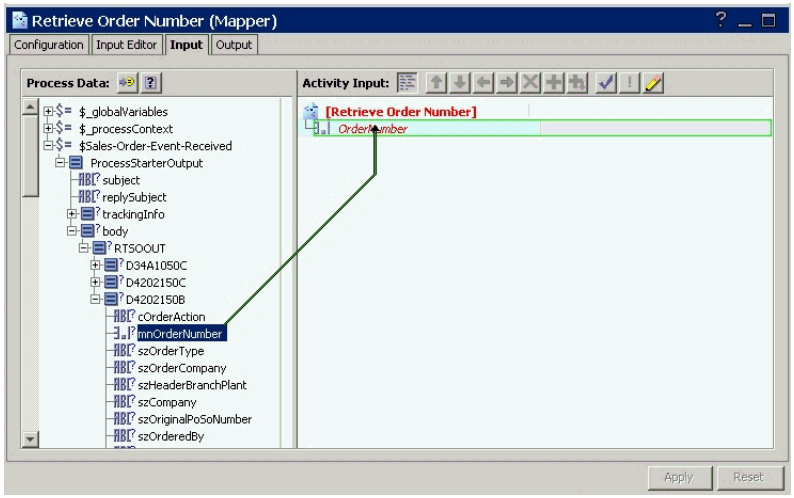
10. Click the `Input Editor` tab. Click the **Add Child** button on the toolbar of this tab. A `root` variable is added.
11. Rename the variable to **OrderNumber**. Select `Content` as **Element of Type** and the `Type` of the variable as **Decimal**. Click **Apply**.



12. In the TIBCO Designer toolbar, click the **Create transition** button. Connect the **Sales Order Event Received** task with the **Retrieve Order Number** task.

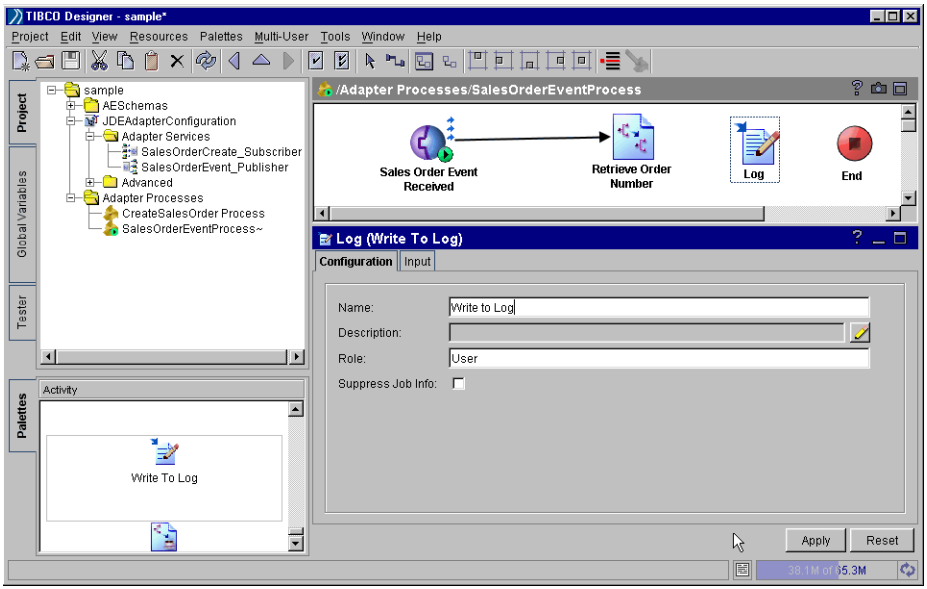


13. Select the **Retrieve Order Number** task in the design panel. Select the `Input` tab. The `Process Data` panel displays a tree structure.
14. Drill down to `$Sales_Order_Event_Received -> ProcessStarterOutput -> body -> RTSOOUT -> D4202150B` in the tree.
15. Drag the **mnOrderNumber** data element and drop it on **OrderNumber** data element in the `Activity Input` panel.

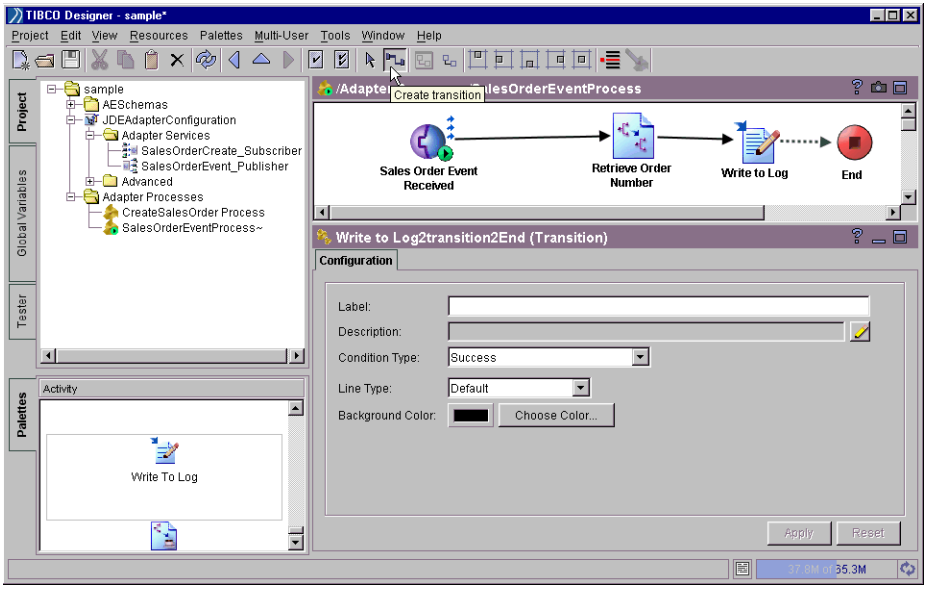


16. From the palette panel, drag and drop the `Write to Log` icon between **Retrieve Order Number** and **End** tasks in the design panel.

17. In the `Configuration` tab, rename the `Log` task to **Write to Log**. Click **Apply**.



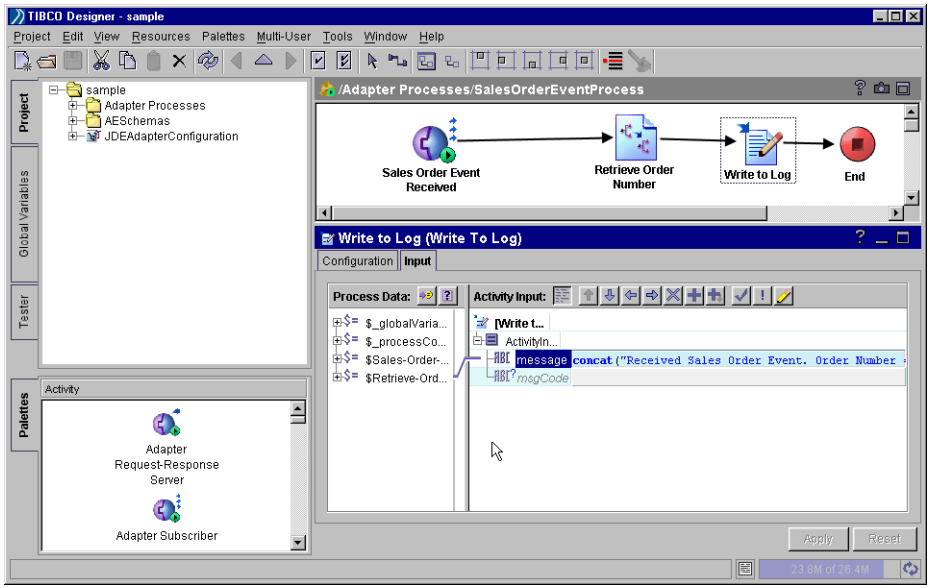
18. Click the **Create transition** button on the TIBCO Designer toolbar. Connect the **Retrieve Order Number** task with the **Write to Log** task. Then connect the **Write to Log** task with the **End** task.



19. Select the **Write to Log** task in the design panel.

20. Click the Select tool on the TIBCO Designer toolbar. Click the `Input` tab and drill down to the **message** variable in the `Activity Input` panel. Type the following value in the **Format** field:

```
concat("Received Sales Order Event. Order Number = ",
string($Retrieve-Order-Number))
```



21. Click **Apply**. Select **Project>Save** to save this process.

Task C Run the Test Process and View Results

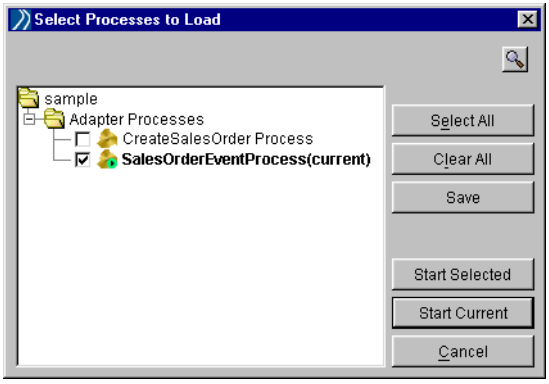
In Task A you created a publication service. In Task B you configured a process flow and used this service in it. In this task, you will test the process and adapter service interaction by running the process diagram in the debug mode from TIBCO BusinessWorks.

Once configured, the process can be run on the command line by running TIBCO BusinessWorks in the engine mode. For more information, see TIBCO BusinessWorks documentation.

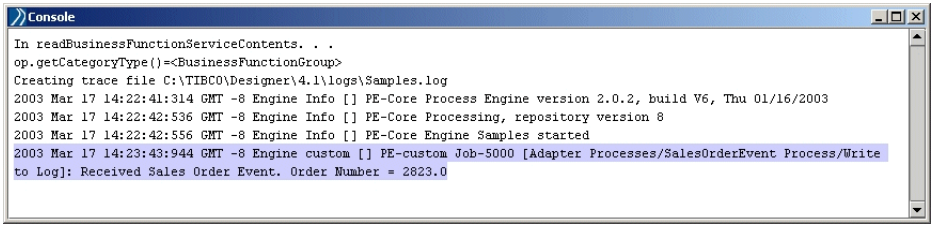
1. In TIBCO Designer, select **Tools>Show Adapter Tester**. The Adapter Tester is displayed. Click `JDEAdapterConfiguration` in the project panel. Specify the **Working Directory** and the **Adapter Executable**. Click **Start** to run the adapter.

2. Click the **Tester** tab in TIBCO Designer and click the **Start testing viewed process** button in the **Tester** panel toolbar, or select **Tools>Tester>Start**. The Select Processes to Load dialog is displayed. Expand the tree for Adapter Processes. By default, both processes are selected.

3. Select **SalesOrderEvent Process** from the Adapter Processes tree and click the **Start Selected** button.



4. Using a J.D. Edwards OneWorld Xe client, log on to J.D. Edwards OneWorld Xe. Launch the Sales Order Entry application and enter a new sales order.
5. Select the TIBCO Designer window. The **SalesOrderEvent Process** should have all connectors showing in green. This indicates that the process has run successfully.
6. Select **Window>Show Console**. Confirm that the new sales order created in J.D. Edwards OneWorld Xe is displayed in the console.



7. In the Adapter Tester window, click the **Stop** button to stop the adapter.

Configure the Request-Response Service for a Business Function

This section explains how to configure a Request-Response service for a business function. The steps are:

1. Configure the Request-Response Service
2. Create a TIBCO BusinessWorks Test Process to Use the Request-Response Service.
3. Run the Test Process and View Results.

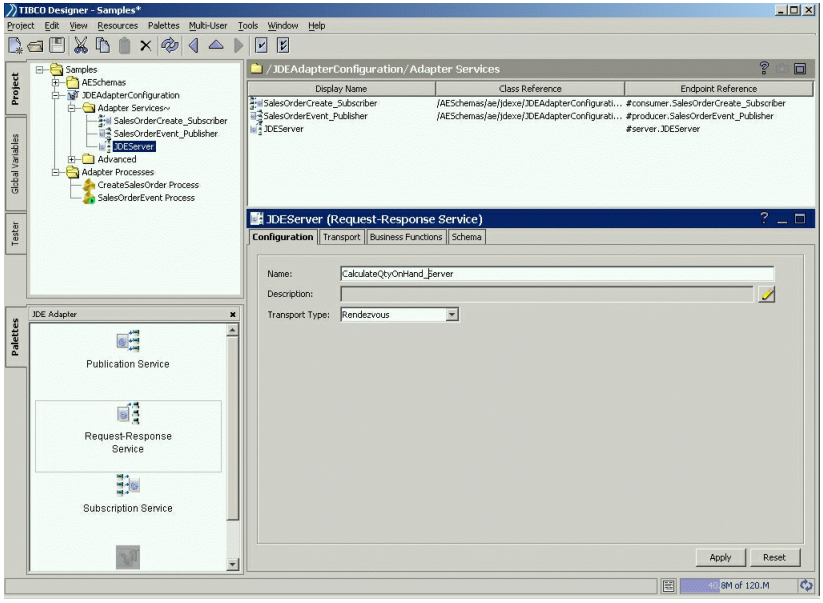
Task D Configure the Request-Response Service

1. In the project panel, expand the `JDEAdapterConfiguration` icon and select the `Adapter Services` folder to access the `Request-Response Service` icon.

   The design panel displays the adapter services that have already been configured.

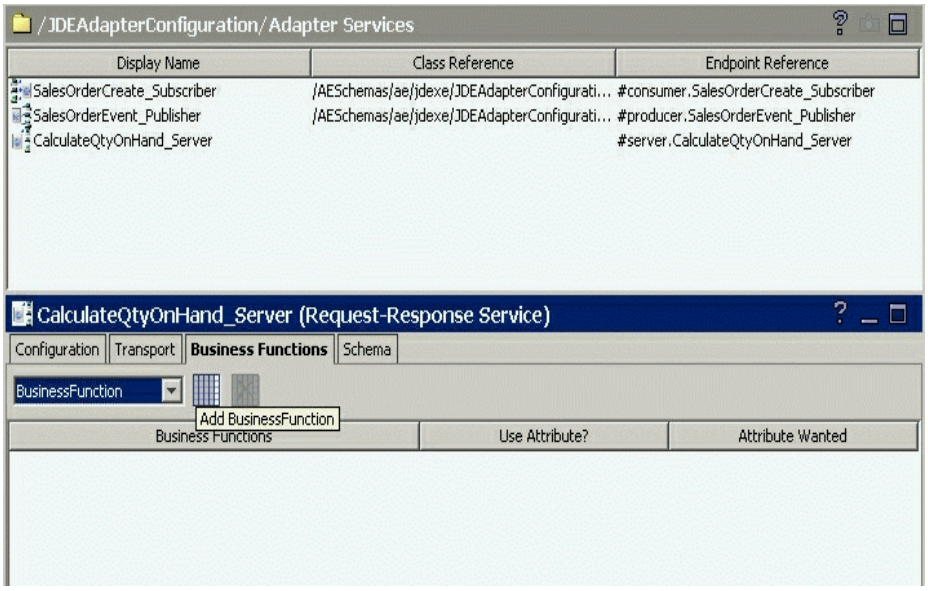
2. Drag the `Request-Response Service` icon from the palette panel to the design panel. A new request-response service `JDEServer` is created.

3. In the `Configuration` tab, rename the request-response service to `CalculateQtyOnHand_Server`. Click **Apply**.

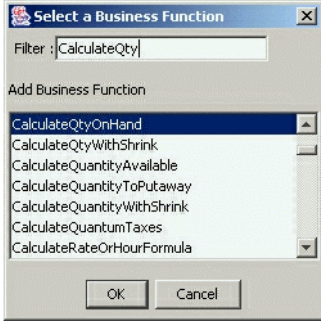


4. Click the `Transport` tab and change the Quality of Service to **Reliable**. Click **Apply**.

5. Select the `BusinessFunctions` tab. Select the **BusinessFunction** category from the **Select a Category** drop-down. The **Add BusinessFunction** is available.



6. Click the **AddBusinessFunction** button. The Select a Business Function dialog is displayed.
7. From the list of business functions, select **CalculateQtyOnHand** and click **OK**.



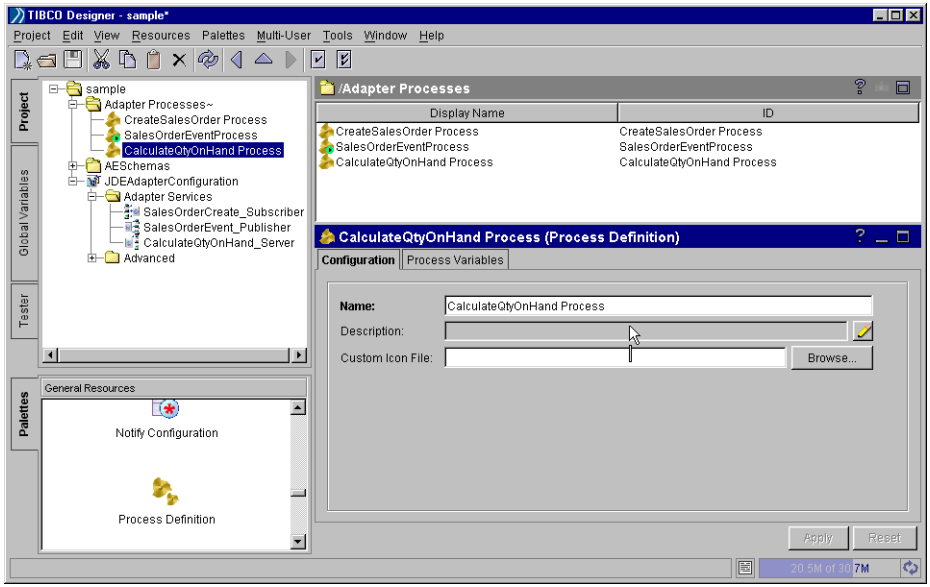
The `BusinessFunctions` tab now shows the data structure tree of the business function that you selected.

8. Expand the tree for the **CalculateQtyOnHand** business function to view the data structure attributes. Click **Apply**.
9. Click **Project > Save** to save this service.

Task E Create a TIBCO BusinessWorks Test Process to Use the Request-Response Service

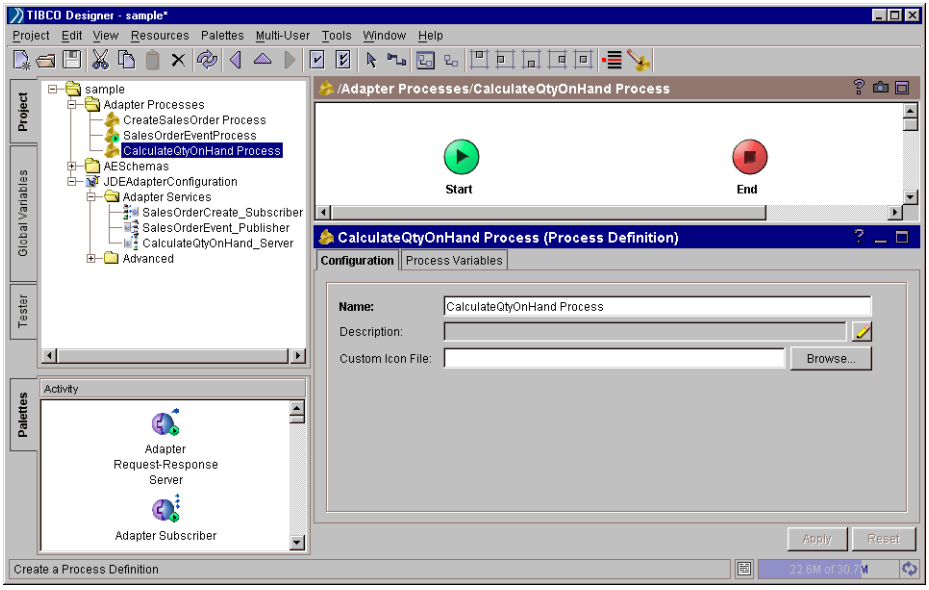
You have created a Request-Response service to create a sales order. Now you will use this service with TIBCO BusinessWorks.

1. Double click the `Adapter Processes` folder in the project panel. The processes that you configured previously will display in the design panel.
2. Drag and drop the `Process Definition` icon from the palette panel to the design panel area.
3. Select the newly created process in the design panel. In the `Configuration` tab, change the name of the process to **CalculateQtyOnHand Process**.
4. Click **Apply**.

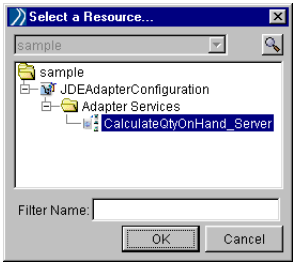


5. In the design panel, double click the process icon for **CreateQtyOnHand Process**.

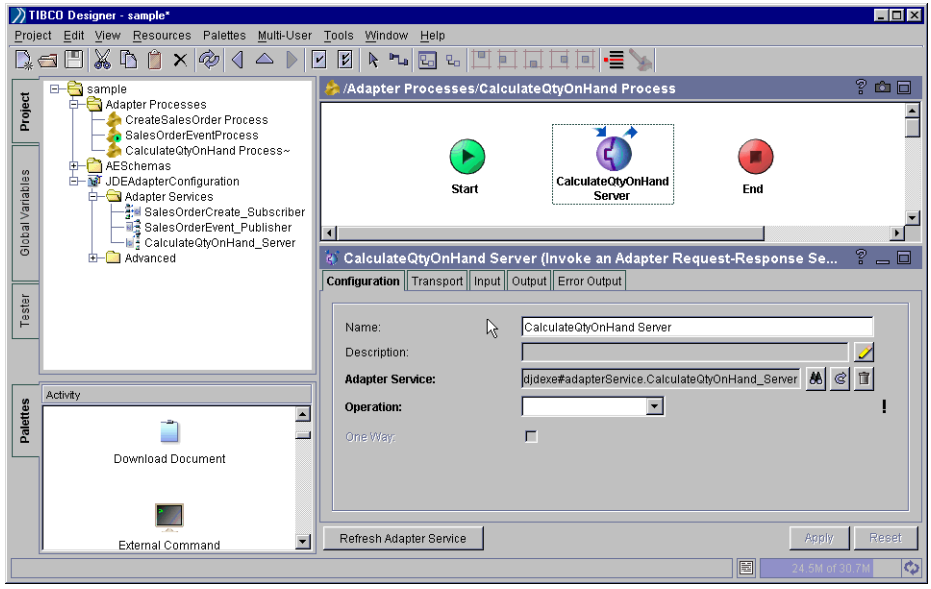
The design panel displays an empty process with **Start** and **End** tasks and the icons in the palette panel change.



6. From the palette panel, drag and drop the **Invoke an Adapter Request-Response Service** icon between the **Start** and **End** tasks on the design panel.
7. Select the **Invoke an Adapter Request-Response Service** task in the design panel. In the `Configuration` tab, change the name of the task to **CalculateQtyOnHand Server**.
8. Click the **Browse Resources** button near the Adapter Services text box. The Select a Resource dialog is displayed. Select the **CalculateQtyOnHand_Server** service. Click **OK**.



9. Click **Apply**.



10. Click the `Input` tab. Expand the tree for CalculateQtyOnHandOperation and **CalculateQtyOnHand**.
11. Assign the following values to the data structure parameters.

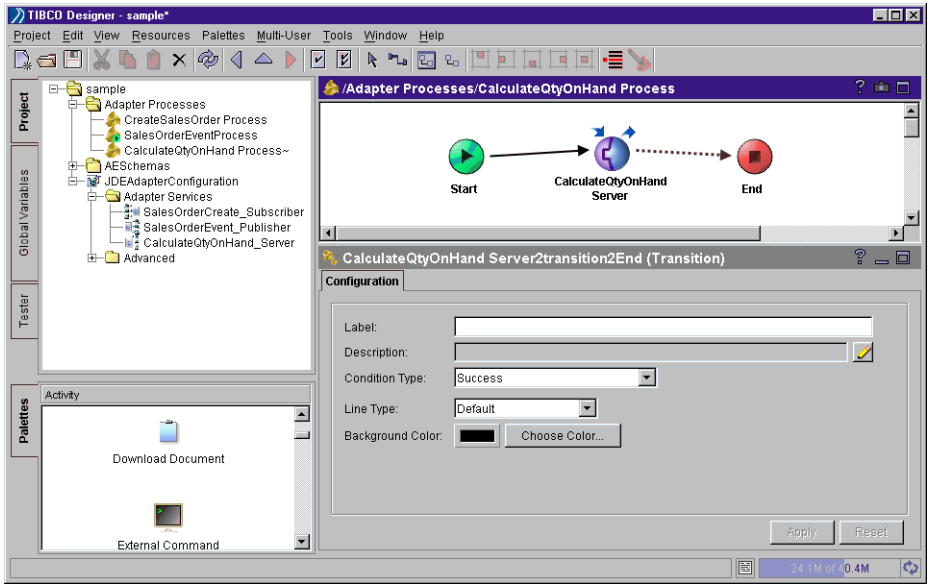
    `mnShortItemNumber`=*short item number*

    `szBranchPlan`=*branch/plant right adjusted*

    `cIndex='1'`

    `cKeys='2'`
12. Click **Apply**.
13. Click the **Create Transition** button on the TIBCO Designer toolbar.

14. Connect the **Start** task with the **CalculateQtyOnHand Server** task. Then connect the **CalculateQtyOnHand Server** task with the **End** task.



15. Select **Project>Save**.

Task F Run the Test Process and View Results

In Task A you created a request-response service. In Task B you configured a process flow and used this service in it. In this task, you will test the process and adapter service interaction by running the process diagram in the debug mode from TIBCO BusinessWorks.

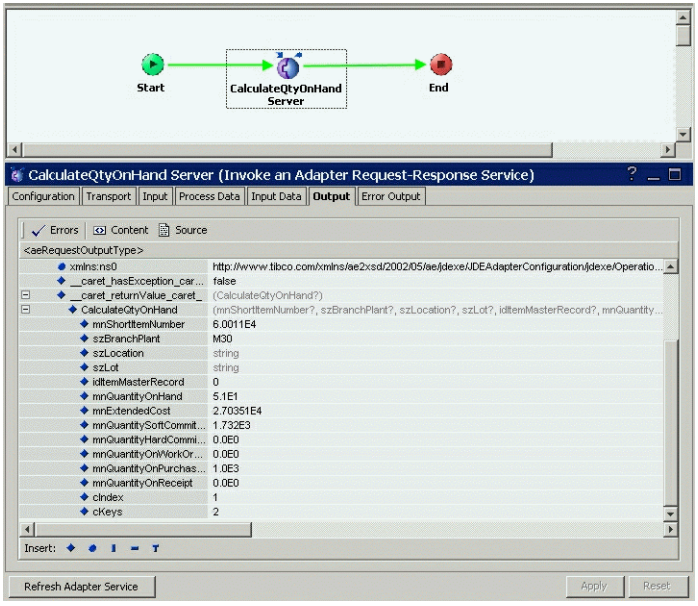
Once configured, the process can be run on the command line by running TIBCO BusinessWorks in the engine mode. For more information, refer to TIBCO BusinessWorks documentation.

1. In TIBCO Designer, select **Tools>Show Adapter Tester**. The Adapter Tester is displayed. Click `JDEAdapterConfiguration` in the project panel. Specify the **Working Directory** and the **Adapter Executable**. Click **Start** to run the adapter.
2. Open the **CalculateQtyOnHand Process** process definition created in the previous section. If you are continuing from the last section, the process definition is already open.
3. Click the **Tester** tab in TIBCO Designer and click the **Start testing viewed process** button in the **Tester** panel toolbar, or select **Tools>Tester>Start**. The

Select Processes to Load dialog is displayed. Expand the tree for Adapter Processes.

4. Select the **CalculateQtyOnHandProcess** process. Click the **Start Selected** button. In TIBCO Designer, the connectors in the process will turn green as each step is executed. Click the `Output` tab and expand the trees for **CalculateQtyOnHandOperation** and **CalculateQtyOnHand**.

   The data structure attributes for the business function will contain values for different quantities.



5. Using a J.D. Edwards OneWorld Xe client, log on to J.D. Edwards OneWorld Xe. Confirm that the quantities displayed in the process are correct.

6. In the Adapter Tester window, click the **Stop** button to stop the adapter.

Configure the Request-Response Service for an XML Query

This section explains how to configure a Request-Response service for an XML Query. The steps are:

1. Configure the Request-Response Service for List Create
2. Configure the Request-Response Service for List Retrieve
3. Configure the Request-Response Service for List Delete
4. Create a TIBCO BusinessWorks Test Process to use this Request-Response Service
5. Run the Test Process and View Results

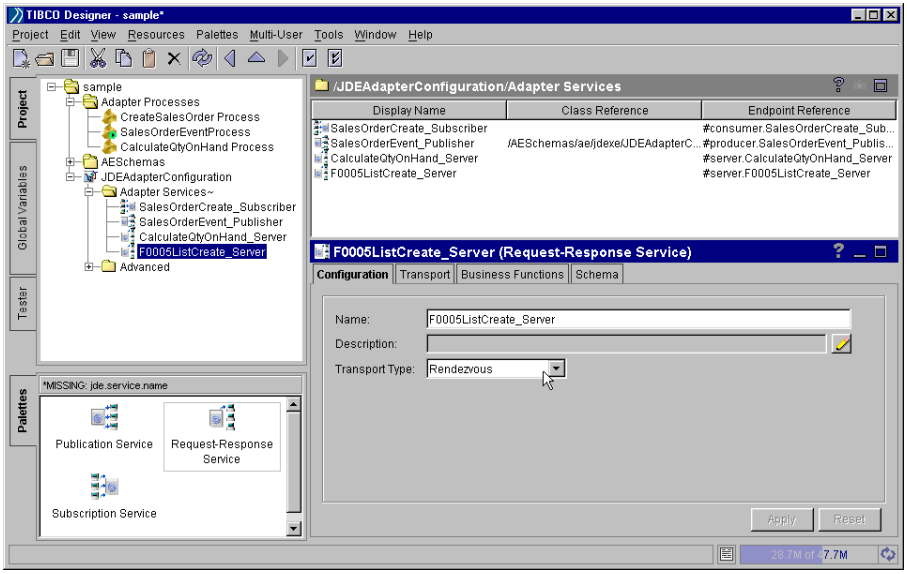
Task A Configure the Request-Response Service for List Create

1. In the project panel, expand the `JDEAdapterConfiguration` icon, and select the `Adapter Services` folder to access the `Request-Response Service` icon.

   The design panel will display the adapter services that have already been configured.

2. Drag the `Request-Response Service` icon from the palette panel to the design panel. A new request-response service `JDEServer` is created.

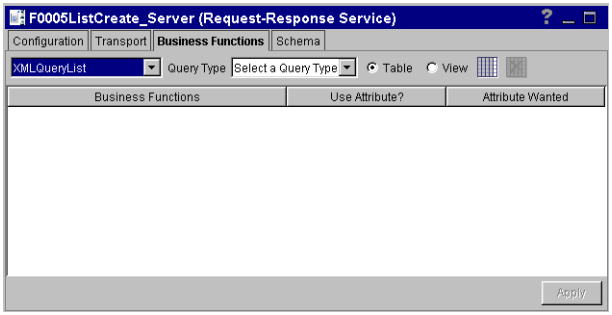
3. In the `Configuration` tab, rename the request-response service to `F0005ListCreate_Server`. Click **Apply**.



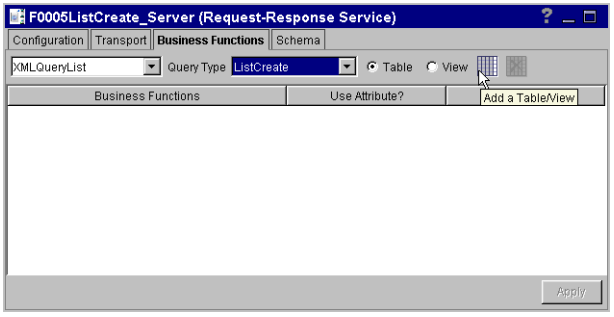
4. Click the `Transport` tab and change the Quality of Service to **Reliable.** Click **Apply**.

5. Click the `BusinessFunctions` tab. Select the **XMLQueryList** category from the Select a Category drop-down.

   The **Table** and **View** options, as well as the **Add a Table/View button** becomes available.

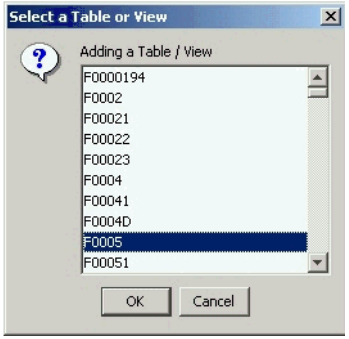


6. Select the **ListCreate** category from the **Query Type** drop-down**.** Select the **Table** option. Click the **Add a Table/View** button.



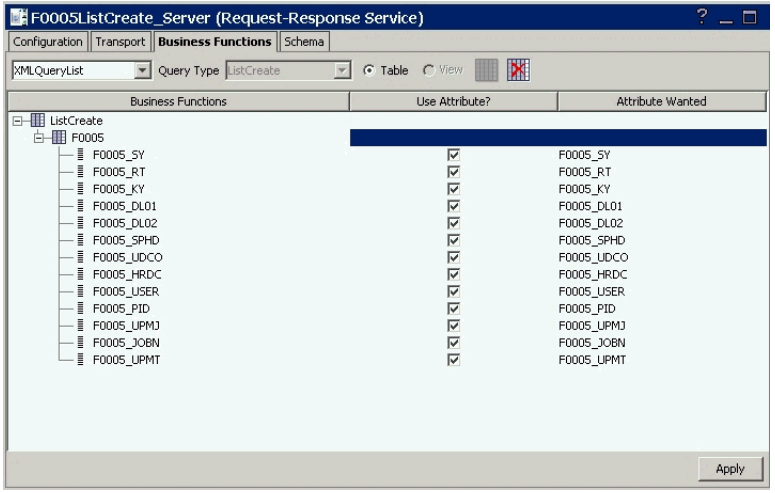
   The Select a Table or View dialog with a list of all tables, is displayed.

7. From the list of tables, select **F0005** and click **OK.**



The `BusinessFunctions` tab now shows the column structure of the table you selected.

8. Expand the tree to view the column attributes for **F0005**. Click **Apply**.
9. Select **Project>Save**.



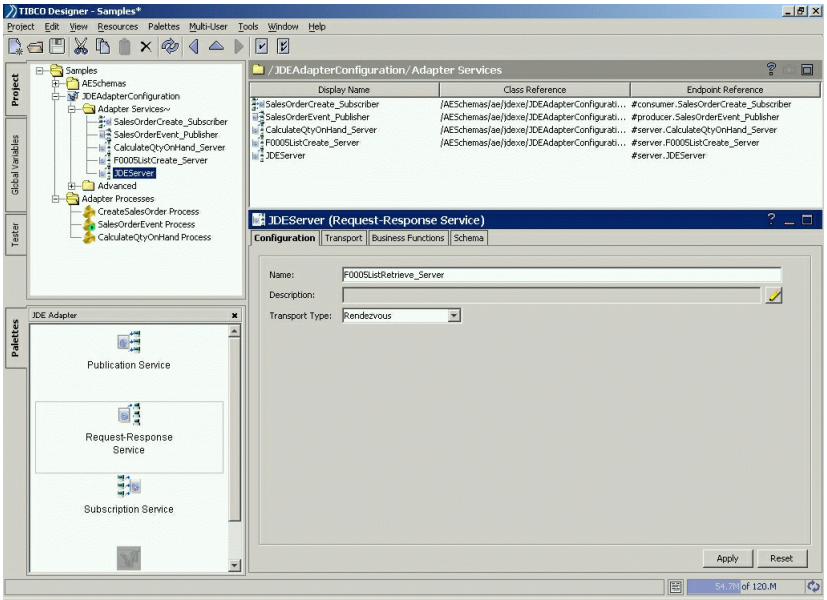
Task B Configure the Request-Response Service for List Retrieve

1. In the project panel, expand the `JDEAdapterConfiguration` icon and select the `Adapter Services` folder to access the `Request-Response Service` icon.

   The design panel will display the adapter services that have already been configured.

2. Drag the `Request-Response Service` icon from the palette panel to the design panel. A new request-response service `JDEServer` is created.

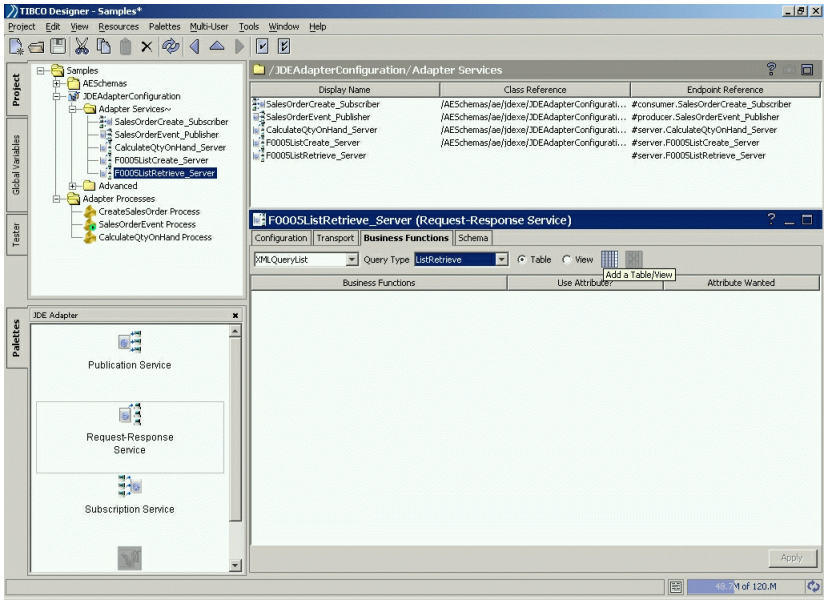
3. In the `Configuration` tab, rename the request-response service to `F0005ListRetrieve_Server`. Click **Apply**.



4. Click the `Transport` tab and change the Quality of Service to **Reliable** and click **Apply**.

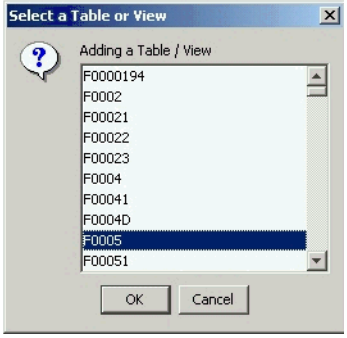
5. Select the `BusinessFunctions` tab. Select the **XMLQueryList** category from the Select a Category drop-down.

The **Table** and **View** options, as well as the **Add a Table/View button** becomes available.



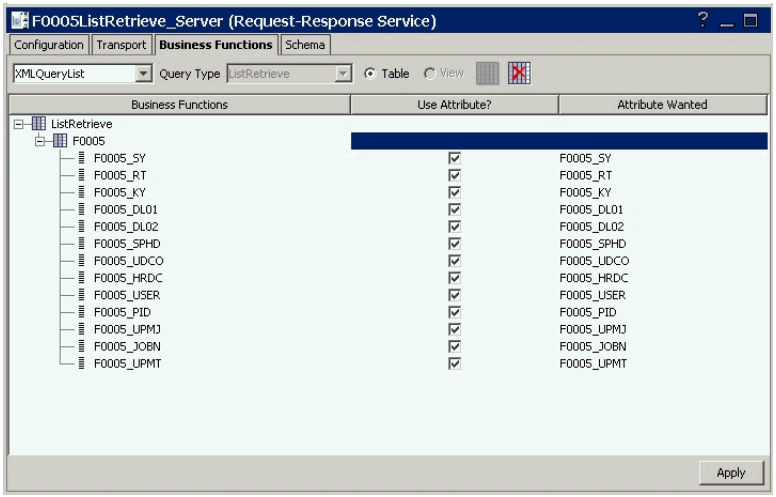
6. Select the **ListRetrieve** category from the **Query Type** drop-down**.** Select the **Table** option. Click the **Add a Table/View** button. A dialog with a list of all tables is displayed.

7. From the list of tables, select **`F0005`** and click **OK**.



The `BusinessFunctions` tab now shows the column structure of the table you selected.

8. Expand the tree to view the column attributes for **F0005**.

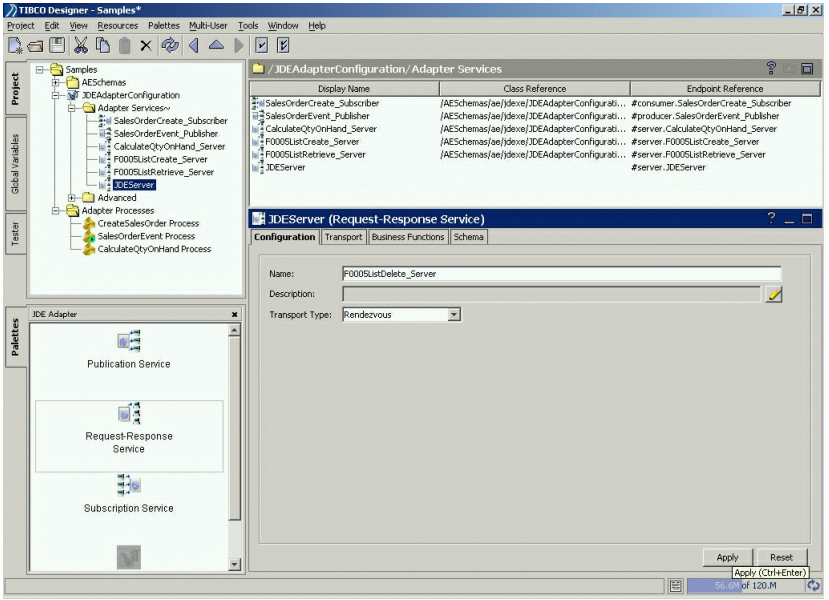


9. Click **Apply**.
10. Select **Project>Save**.

Task C Configure the Request-Response Service for List Delete

1. In the project panel, expand the `JDEAdapterConfiguration` icon and select the `Adapter Services` folder to access the `Request-Response Service` icon. The design panel will display the adapter services that have already been configured.
2. Drag the request-response icon from the palette panel to the design panel. A new request-response service `JDEServer` is created in the design panel.

3. In the `Configuration` tab, rename the request-response service to `F0005ListDelete_Server`. Click **Apply**.

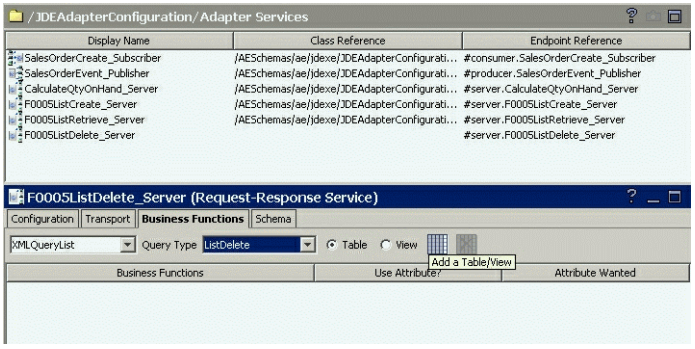


4. Click the `Transport` tab and change the Quality of Service to **Reliable**. Click **Apply**.

5. Select the `BusinessFunctions` tab. Select the **XMLQueryList** category from the Select a Category drop-down.

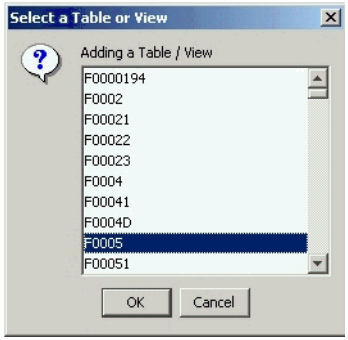
   The **Table** and **View** options, as well as the **Add a Table/View** and **Remove a Table/View buttons** become available.

6. Select the **ListDelete** category from the Query Type drop-down**.** Click the **Add a Table/View** button.



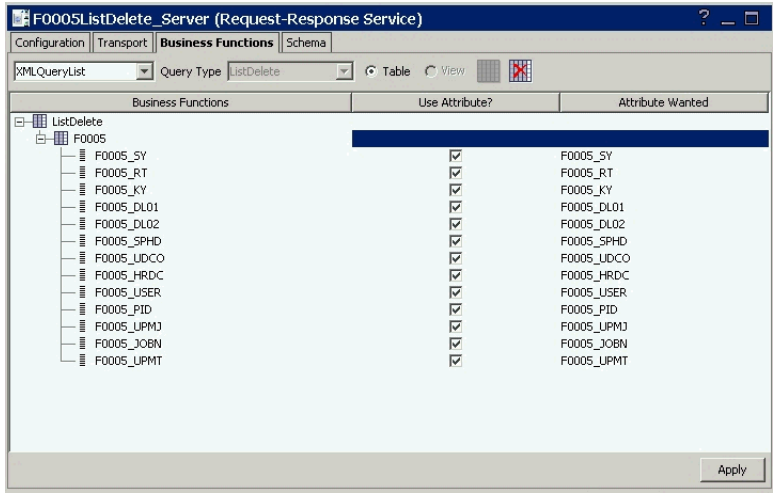
The Select a Table or View dialog with a list of all tables, is displayed.

7. From the list of tables, select **F0005** and click **OK**.



The `BusinessFunctions` tab now shows the column structure of the table you selected.

8. Expand the tree to view the column attributes for **F0005**.



9. Click **Apply**.
10. Select **Project>Save**.

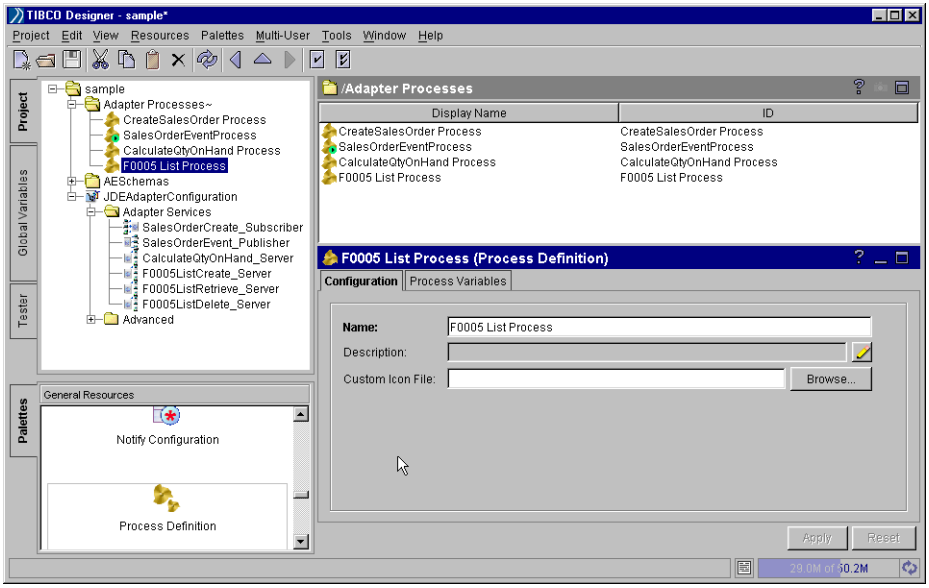
Task D Create a TIBCO BusinessWorks Test Process to use this Request-Response Service

You have made three request-response services to perform an XML query. Now you will use these services with TIBCO BusinessWorks.

1. Double click the `Adapter Processes` folder in the project panel.

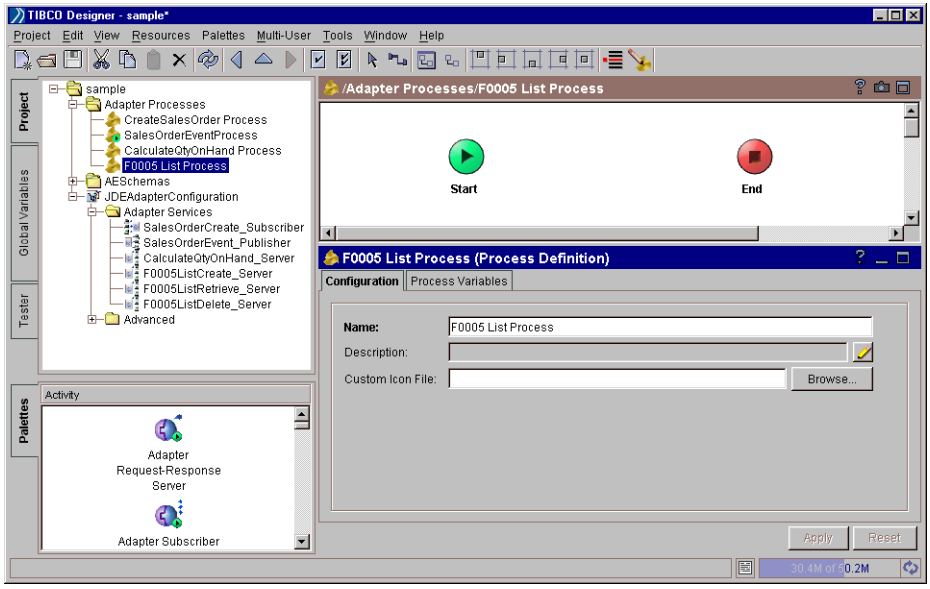
Any processes that you configured previously will display in the design panel.

2. Drag and drop the `Process Definition` icon from the palette panel, to the design panel.
3. Select the newly created process in the design panel. In the `Configuration` tab, change the name of the process to **F0005 List Process**. Click **Apply**.

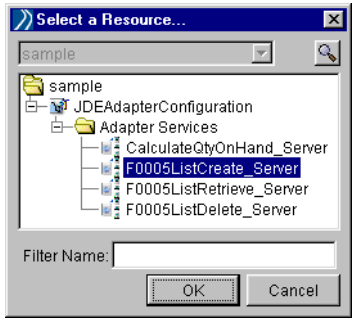


4. In the design panel, double click the process icon for **F0005 List Process**.

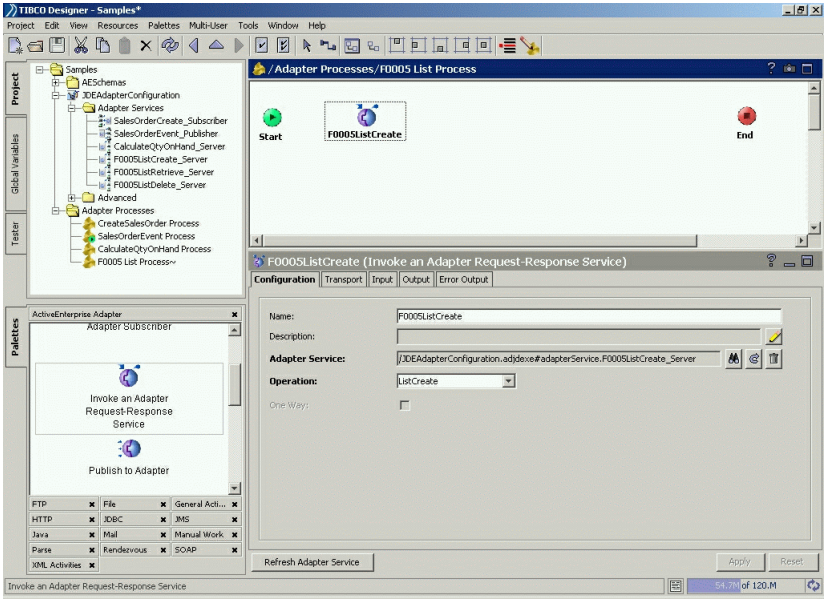
The design panel will display an empty process with **Start** and **End** tasks and the icons in the palette panel will change.



5. From the palette panel, drag and drop the **Invoke an Adapter Request-Response Service** icon between the **Start** and **End** tasks, in the design panel.
6. Select the **Invoke an Adapter Request-Response Service** task in the design panel. In the `Configuration` tab, change the name of the task to **F0005ListCreate**.
7. Click the **Browse Resources** button near the Adapter Services text box. The Select a Resource dialog is displayed. Select the **F0005ListCreate_Server** service. Click **OK**.



8. Click **Apply**.

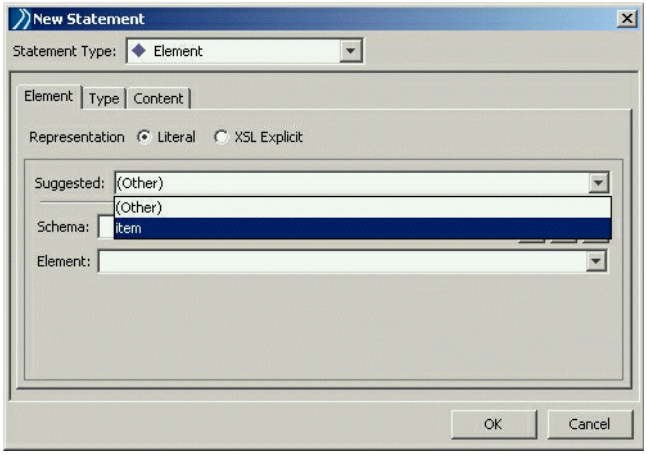


9. Click the `Input` tab in the **F0005 ListCreate** task.

10. Expand the tree for **List Create** and **Data Selection**, click **ClauseType**, and then click the **Add Child** button in the **Activity Input** panel.

    A New Statement dialog is displayed.

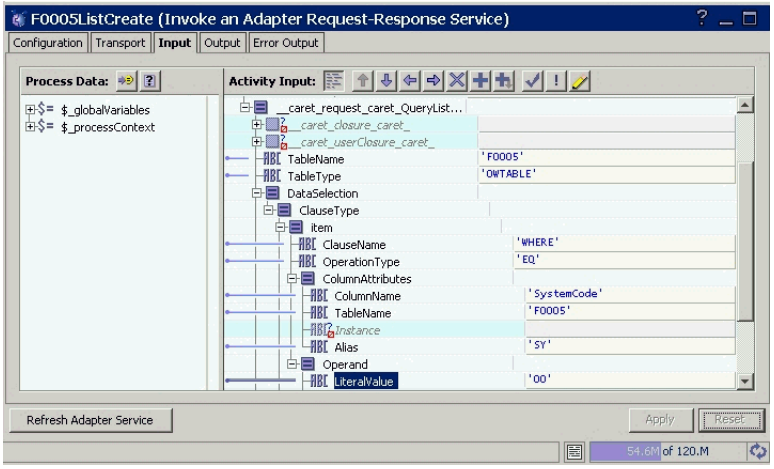
11. From the **Statement Type** drop-down, select **Element**. Under **Representation**, select **Literal**. From the **Suggested** drop-down, select **item**. Click **OK**.



12. In the `Input` tab, assign the following values to the data structure parameters.

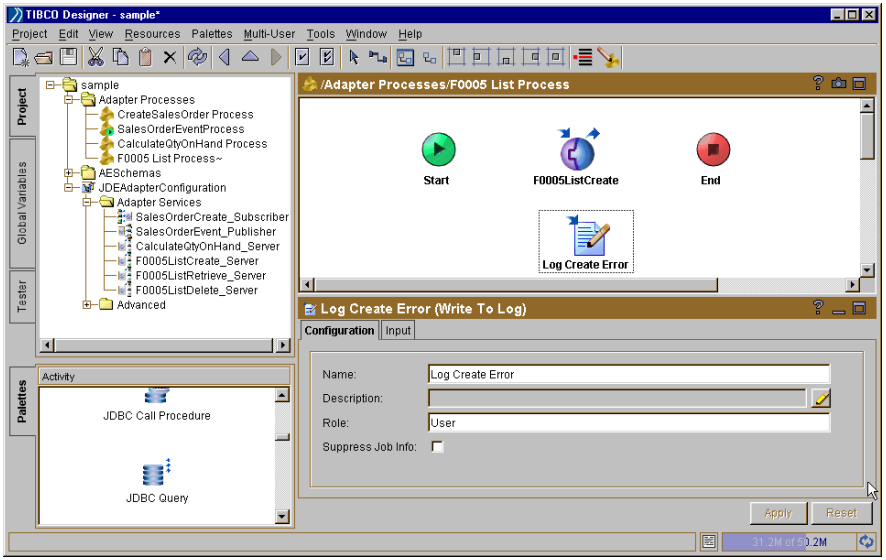
```
TableName='F0005'
TableType='OWTABLE'
DataSelection
   ClauseType
      item
         ClauseName='WHERE'
         OperationType='EQ'
         ColumnAttributes
            ColumnName='SystemCode'
            TableName='F0005'
            Alias='SY'
         Operand
            LiteralValue='00'
      item
         ClauseName='AND'
         OperationType='EQ'
         ColumnAttributes
            ColumnName='RecordType'
            TableName='F0005'
            Alias='RT'
      Operand
            LiteralValue='DT'
DataSequencing
   Sort='ASCENDING'
   ColumnAttributes
      ColumnName='KeyCode'
      TableName='F0005'
      Alias='KY'
```

13. Click **Apply**.



14. From the palette panel, drag and drop the **Write to Log** task icon in the design panel below the **F0005 List Create** task.

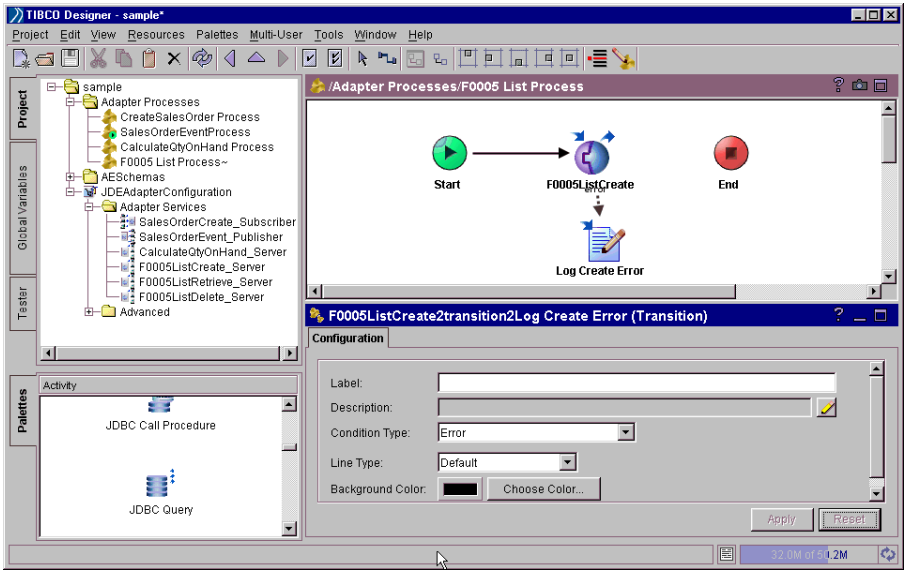
15. Select the **Log** task in the design panel. In the `Configuration` tab, rename the task to **Log Create Error**. Click **Apply**.



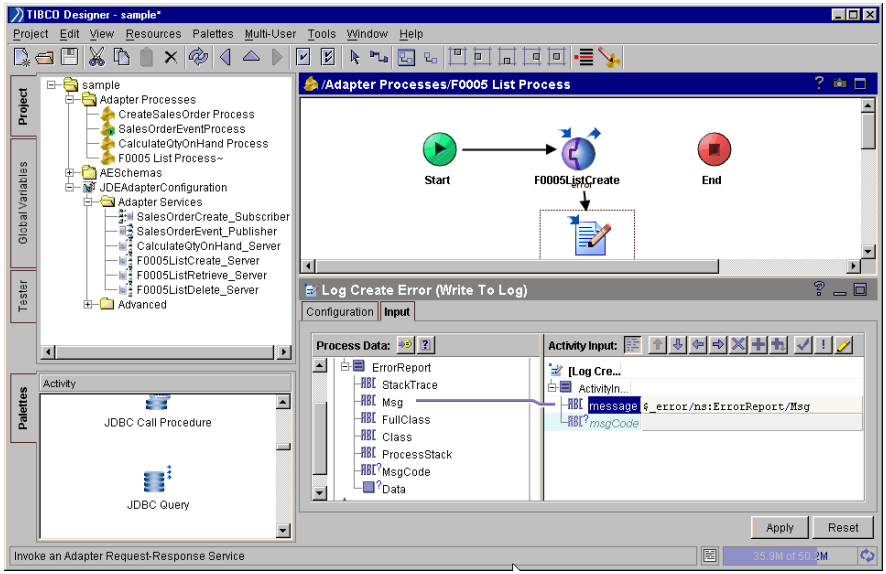
16. Click the **Create Transition** button on the TIBCO Designer toolbar.

17. Connect the **Start** task with the **F0005 List Create** task. Then connect the **F0005 List Create** task with the **Log Create Error** task.

18. In the `Configuration` tab, select the **Condition Type** value as **Error**. Click **Apply**.

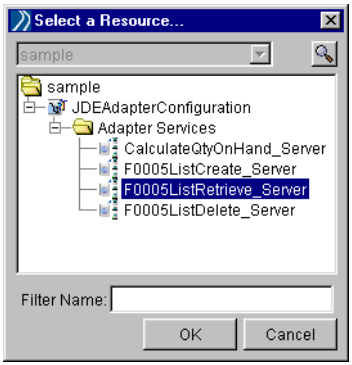


19. In the TIBCO Designer toolbar, click the **Select** button. Select the **Log Create Error** task.
20. Select the `Input` tab.
    - In the **Process Data** panel, expand the `$_error` tree.
    - In the **Activity Input** panel, drag and drop the **Msg** data element to the **message field**.
    - Click **Apply**.

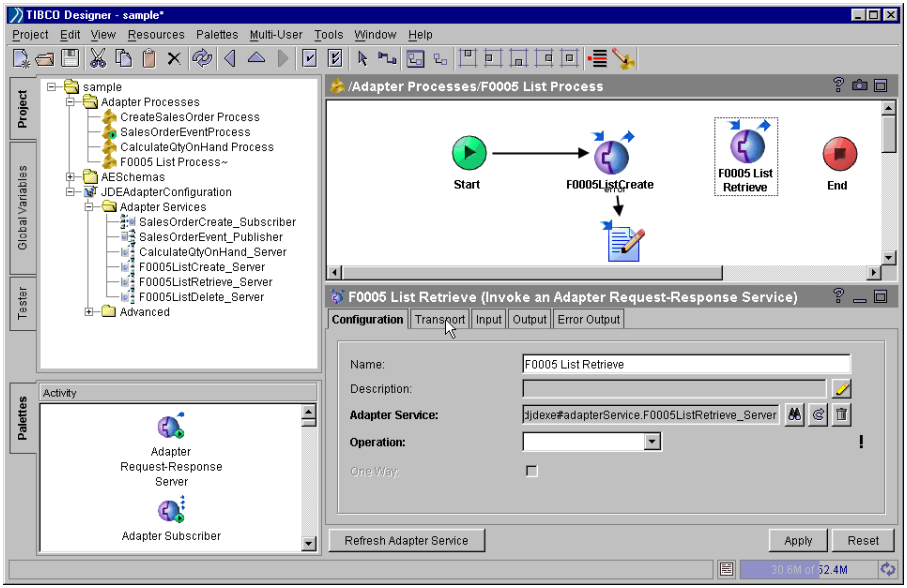


21. From the palette panel, drag and drop an `Invoke an Adapter Request-Response Service` icon to the design panel area, between **F0005 List Create** and **End** tasks.

22. Select **Invoke an Adapter Request-Response Service** task in the design panel. Rename the task to **F0005 List Retrieve** in the `Configuration` tab and select **F0005ListRetrieve_Server** in the Adapter Service box. Click **OK**.

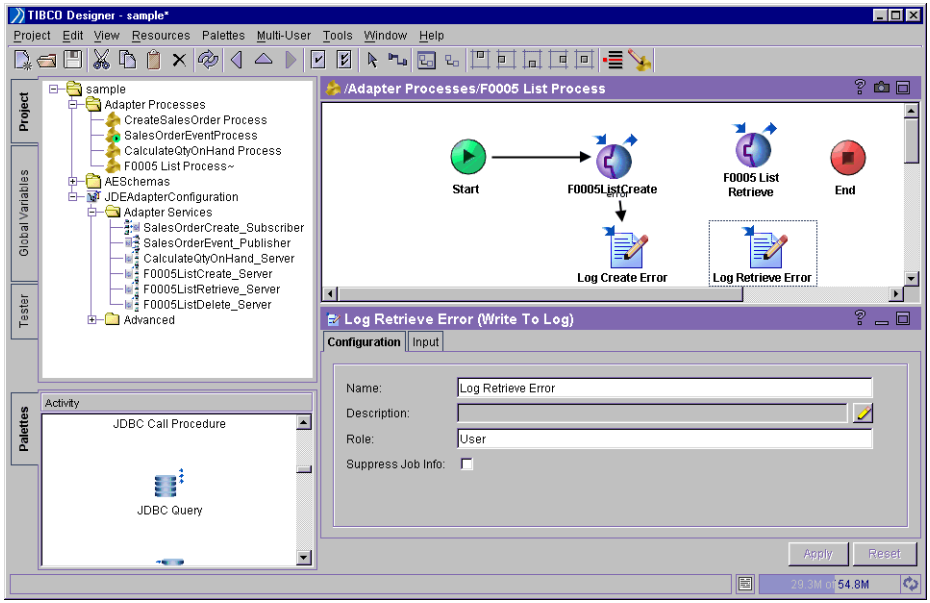


23. Click **Apply.**



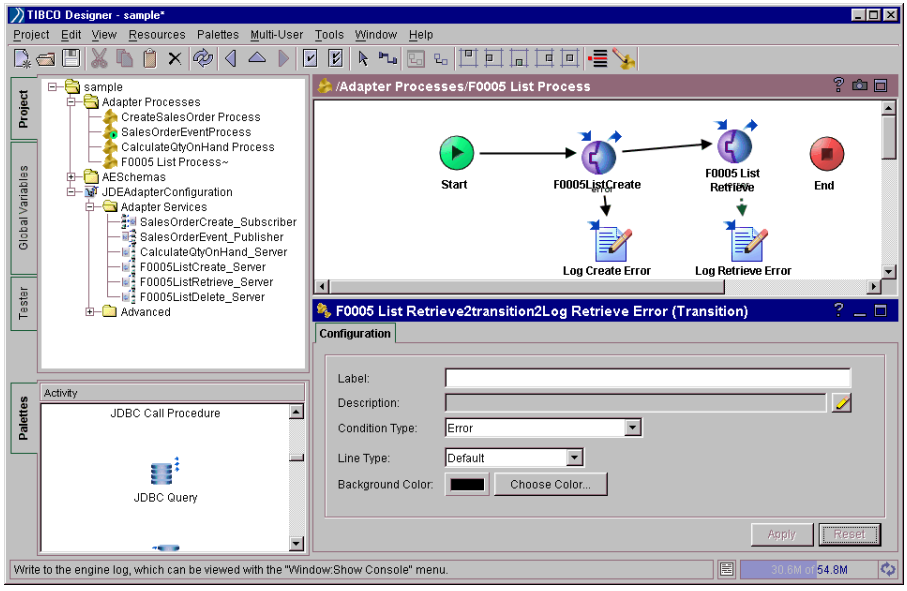
24. From the palette panel, drag and drop the **Write to Log** task icon to the design panel area below the **F0005 List Retrieve** task.

25. Select the **Log** task in the design panel. In the `Configuration` tab, rename the task to **Log Retrieve Error**. Click **Apply**.



26. Click the **Create transition** button on the TIBCO Designer toolbar. Connect the **F0005 List Create** task with the **F0005 List Retrieve** task. Then connect the **F0005 List Retrieve** task with the **Log Retrieve Error** task.

27. In the `Configuration` tab, select **Condition Type** value as **Error**. Click **Apply.**



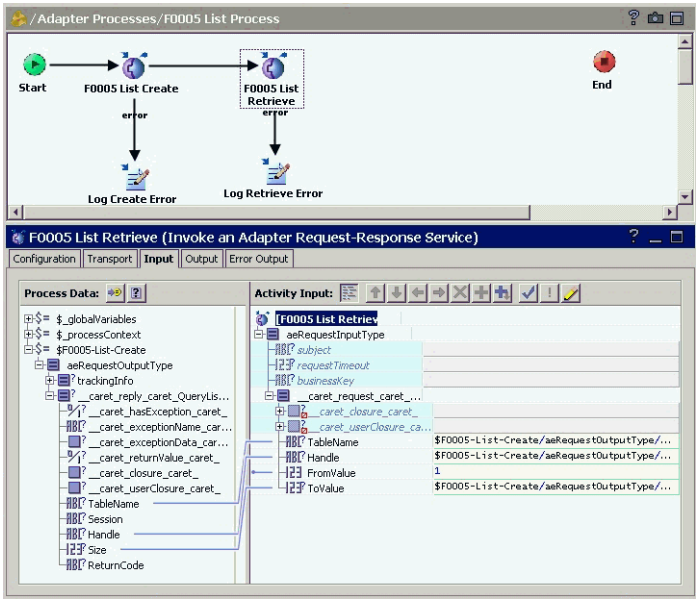
28. Click the **Select** button on the TIBCO Designer toolbar. Select the **F0005 List Retrieve** task. Select the `Input` tab.

    a. In the Process Data panel, expand the tree for **$F0005-List-Create -> ListCreate**

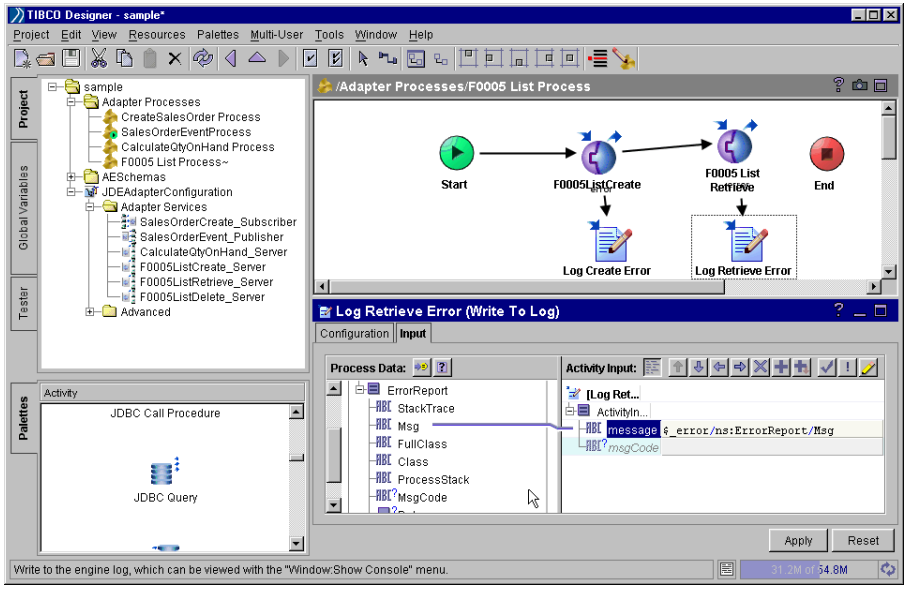
    b. In the Activity Input panel, expand the tree for **ListRetrieve**.

    c. Drag and drop the data element **TableName** from the Process Data panel to the data element **TableName** in the Activity Input panel.

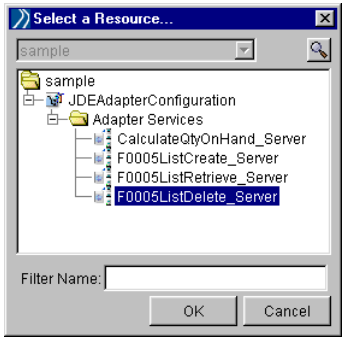
    d. Similarly, map **Handle** to **Handle** and **Size** to **ToValue**. Click **Apply**.



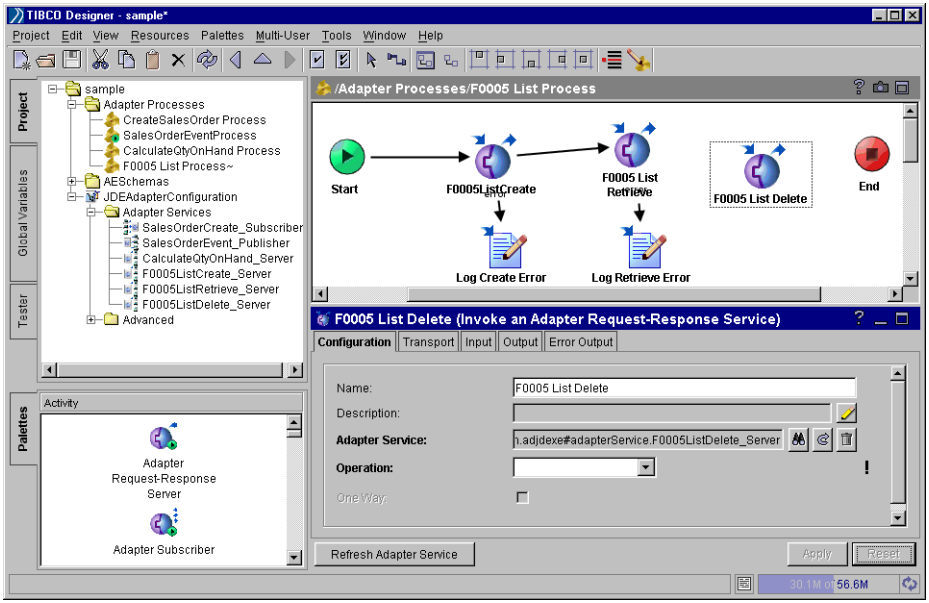
29. Select the **Log Retrieve Error** task. Select the `Input` tab.
    a. In the Process Data panel, expand the **$_error** tree.
    b. In the Activity Input panel, drag and drop the **Msg** data element to the **message** field. Click **Apply**.



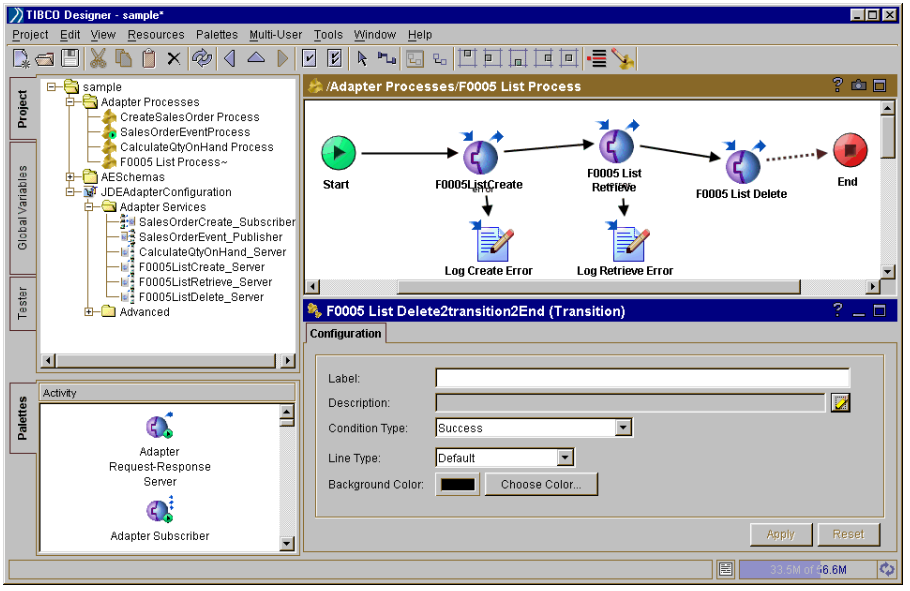
30. From the palette panel, drag and drop an `Invoke an Adapter Request-Response Service` icon between the **F0005 List Retrieve** and **End** tasks, in the design panel.
31. Select the **Invoke an Adapter Request-Response Service** task in the design panel. In the `Configuration` tab, rename this task to **F0005 List Delete**.
32. Click the **Browse Resources** button near the Adapter Services text box. The Select a Resource dialog is displayed. Select **F0005ListDelete_Server** in the Adapter Service box. Click **OK**.



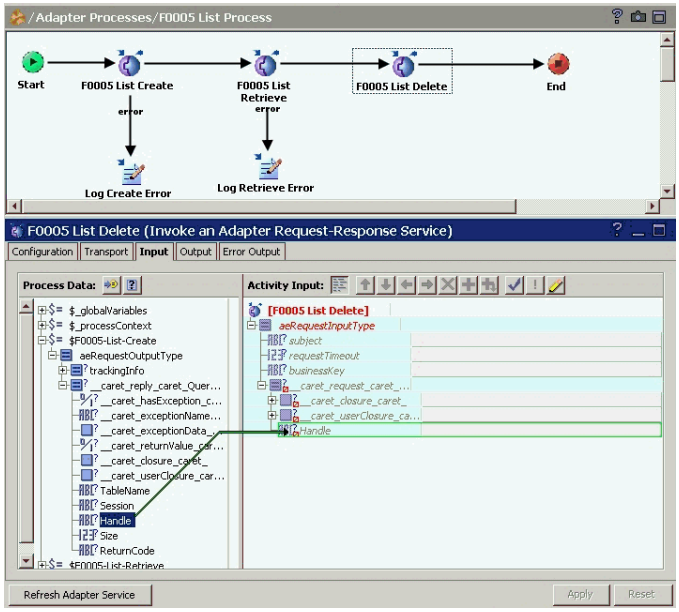
33. Click **Apply**.



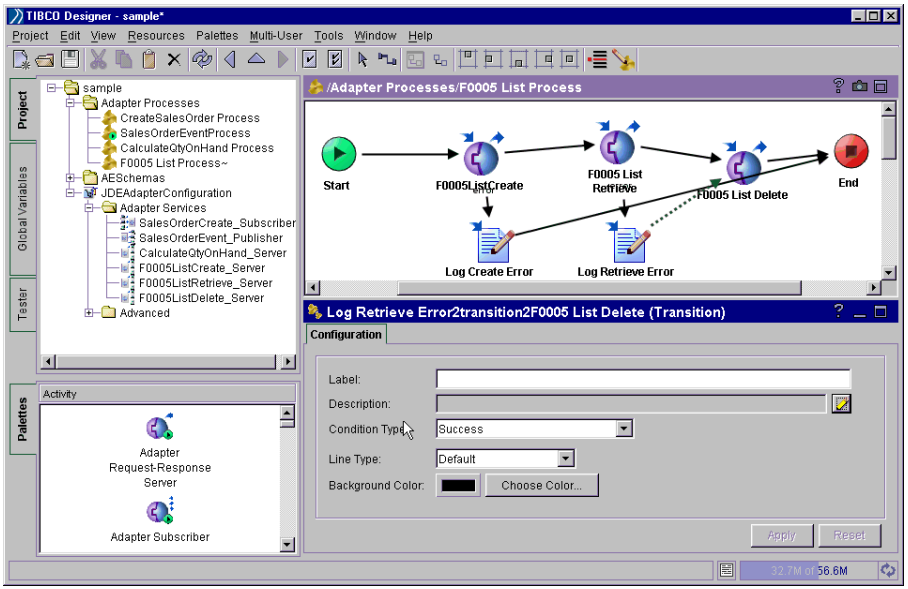
34. Click the **Create transition** button on the TIBCODesigner toolbar.
    - Connect the **F0005 List Retrieve** task with the **F0005 List Delete** task.
    - Connect the **F0005 List Delete** task with the **End** task.



35. Click the **Select** button on the TIBCO Designer toolbar. Select the **F0005 List Delete** task.
36. Click the `Input` tab.
    a. In the Process Data panel, expand the tree for $**F0005-List-Create -> ListCreate**.
    b. In the Activity panel, expand the tree for **ListDelete**.
    c. Drag and drop the data element **Handle** from the Process Data panel to the data element **Handle** in the Activity Input panel.
37. Click **Apply**.



38. Click the **Create transition** button on the TIBCO Designer toolbar.
    a. Connect the **Log Create Error** task with the **End** task.
    b. Connect the **Log Retrieve Error** task with the **F0005 List Delete** task.



39. Select **Project>Save**.

Task E Run the Test Process and View Results

In Tasks A, B, and C, you created a set of request-response services. In Task D, you configured a process flow and used these services in it. In this task, you will test the process and adapter service interaction by running the process diagram in the debug mode from TIBCO BusinessWorks.

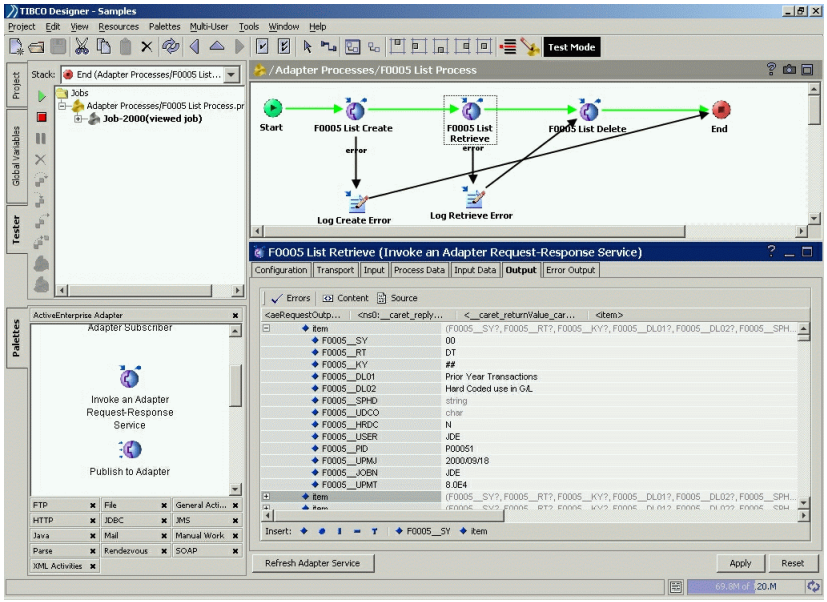
Once configured, the process can be run on the command line by running the TIBCO BusinessWorks in the engine mode. For more information, refer to TIBCO BusinessWorks documentation.

1. In TIBCO Designer, select **Tools>Show Adapter Tester**. The Adapter Tester is displayed. Click `JDEAdapterConfiguration` in the project panel. Specify the **Working Directory** and the **Adapter Executable**. Click **Start** to run the adapter.
2. Click the **Tester** tab in TIBCO Designer and click the **Start testing viewed process** button in the **Tester** panel toolbar, or select **Tools>Tester>Start**. The Select Processes to Load dialog is displayed. Expand the tree for Adapter Processes. Select the required processes and click **Start Selected**.

The connectors in the process will turn green as each step is executed.

3. Select the **F0005 List Retrieve** task in process. Click the `Output` tab and expand the trees for **ListRetrieve>returnValue**.
4. Expand each occurrence of **returnValue** in the list.

   The column attributes for the table will contain values from the database table rows.



5. Using a J.D. Edwards OneWorld Xe client, log on to J.D. Edwards OneWorld Xe. Launch the Universal Table Browser and confirm that the records displayed in the process are correct.
6. In the Adapter Tester window, click the **Stop** button to stop the adapter.

Chapter 5 Adapter Instance Options

This chapter explains how to create an adapter instance and assign it services by configuring standard settings. All configuration tasks are performed in TIBCO Designer and the information is stored in a project that is later used by the run-time adapter.

Topics

Configuration Tasks for an Adapter

To configure an adapter instance, you must do the following:

- Configure the jdeinterop.ini
- Configure an Adapter Instance

Configure the jdeinterop.ini

The section lists the method to configure `jdeinterop.ini` file of the supported J.D. Edwards applications.

- If you use J.D. Edwards OneWorld Xe, copy the connectivity configuration file `jdeinterop.ini` from the `Interoperability/Java/Doc` directory on the deployment server. Modify this configuration file to specify the name of your J. D. Edwards OneWorld Xe enterprise server and the JDE port number.
- If you use PeopleSoft EnterpriseOne 8.9 or above (with Tools Release below 8.96), copy the `jdeinterop.ini.templ` file from from `System/Classes/Samples` directory to the machine that runs the adapter and rename it to `jdeinterop.ini`. Modify this configuration file to specify the name of your J. D. Edwards enterprise server and the JDE port number.

  At the end of the `jdeinterop.ini` file, add the contents of the `jdbj.ini` file of your JAS server (HTML Web Server) installation.
- If you use PeopleSoft EnterpriseOne with Tools Release 8.96 or higher, configure the following section in the `jdeinterop.ini` file:

```
[EVENTS]
UseGuaranteedEventsSystem = True
Transport = HTTP
eventServiceURL=http://TRANSACTION_SERVER_NAME:TRANSACTION_SERVER_HTTP_PORT/e1events/EventClientService
```

- `TRANSACTION_SERVER_NAME is the name of` machine on which Transaction Server is installed.
- `TRANSACTION_SERVER_HTTP_PORT` is the port on the Transaction Server which listens to the HTTP requests. By default this port is `9082`.

```
port=7013
ListenerMaxConnection = 100
ListenerMaxQueueEntry = 100
outbound_timeout = 6000000
```

Configure an Adapter Instance

Use the following sequence to create and configure an adapter instance.

1. Start TIBCO Designer and open a multi-file project. See the *TIBCO Designer User's Guide* for details.
1. Drag the TIBCO Adapter for J. D. Edwards OneWorld Xe icon from the palette panel to the design panel. This creates an adapter named, by default, `JDEAdapterConfiguration`.
2. Define the adapter instance by assigning a new name and optionally change logging options. See Logging Tab on page 126 for details.
3. Enter information on connection options. See Design-time Connection Tab on page 123 and g for details.
4. Optionally, change the Inbound Thread count. See Multithreading Tab on page 126 for details.
5. Add a service to the adapter instance. To add and configure the:
   - Publication service, see Configuration Tasks for a Publication Service on page 116.
   - Subscription service, see Configuration Tasks for a Subscription Service on page 117.
   - Request-Response service, see Configuration Tasks for a Request-Response Service on page 120.

Configuration Tasks for a Publication Service

After configuring the adapter, as described in Configuration Tasks for an Adapter on page 114, you can configure a publication service.

1. Open the `Adapter Services` folder in the project panel to display the services palette.
2. Add a publication service to the adapter configuration by dragging the service icon from the palettes panel and dropping it in the design panel. Under the C.`onfiguration` tab, set the combination of options required for your service. For information the Configuration Tab, refer to Configuration Tab on page 131.
3. Click **Apply** to save the configuration.
4. In the `Real-Time Event` Tab, click the **Select Real Time Event** button.
5. A drop-down list of events is displayed. In general, the events in the list are those which have been configured in J. D. Edwards OneWorld Xe. For further information on configuring J. D. Edwards OneWorld Xe, see Chapter 6, Deploying and Starting an Adapter Using TIBCO Administrator, on page 149.
6. Choose the appropriate event from the drop down list. An event structure appears in the configuration panel. Expand the event structure.
7. By default, all attributes in a data structure are selected. To unselect a particular attribute, clear the **Use Attributes?** check box.

If a data structure in the event can have multiple occurrences in the event, select the **Use Looping** check box.

8. Click **Apply** to save the configuration of this event.

You can configure the adapter to publish other events available in the J. D. Edwards OneWorld Xe system by repeating steps 2 through 9 of this procedure.

9. Configure the `Schema` tab setting. See Schema Tab on page 135 for details.
10. Save the project and exit TIBCO Designer.
11. Create the run-time adapter property file and add the project name and adapter configuration name. See Chapter 6, Deploying and Starting an Adapter Using TIBCO Administrator, on page 149 for details.

Configuration Tasks for a Subscription Service

After configuring the adapter, you can add subscription groups of business functions or business function services to it. For information on configuring the adapter, see Configuration Tasks for an Adapter on page 114.

Configuring the Subscription Service for a Business Function

Before configuring the subscription service for a business function, generate the GenJava wrapper for the business function and make sure that the `GenJava JAR` file is in the `interop` directory specified in the `Design-time Connection` and `Run-time connection` tab.

1. Open the `Adapter Services` folder in the project panel of the adapter.
2. Add a subscription service to the adapter configuration by dragging the service icon from the palette panel and dropping it in the design panel. Use the `Configuration` Tab to rename the service and set transfer parameters. Refer to Configuration Tab on page 136.
3. In the `BusinessFunctions` Tab, select a Category, either Business Function or Business Function Group. In this case, choose Business Function. Click the **Add Business Function** button.
4. Select a Business Function from the Select a Business Function dialog and click **OK**.
5. Expand the Business Function to see the attributes.

   By default, all attributes in a data structure are selected. To unselect a particular attribute, clear the **Use Attributes?** check box. Click **Apply**.
6. Configure the `Schema` Tab Setting. See Schema Tab on page 135 for details.
7. Save the project and exit TIBCO Designer.
8. Create the run-time adapter property file and add the project name and adapter configuration name. See Chapter 6, Deploying and Starting an Adapter Using TIBCO Administrator, on page 149 for details.

Configuring the Subscription Service for a Business Function Group

Configuring the Subscription Service for a Business Function Group is similar to configuring it for a Business Function. This section describes how to add a Business Function Group to the adapter.

Before configuring the subscription service for a business function group, generate the GenJava wrappers for all the business functions in the group and make sure that the `GenJava JAR` files are in the interop directory specified in the `Design-time Connection` and `Run-time Connection` tab.

1. Open the `Adapter Services` folder in the project panel of the adapter.
2. Add a subscription service to the adapter configuration by dragging the service icon from the palettes panel and dropping it in the design panel
3. Use the `Configuration` Tab to rename the service and set transfer parameters. Refer to Configuration Tab on page 136.
4. In the `BusinessFunctions` tab, select a Category, either Business Function or Business Function Group. In this case choose Business Function Group.
5. Name the Business Function Group in the GroupName text box.
6. Click the **Add Business Function** icon.
7. Select a Business Function from the drop-down and click **OK**. Since this is a Business Function Group, you will be able to add more than one business function. You can add the functions one at a time.
8. Expand the Business Function that you just added, to see the attributes.

   By default, all attributes in a data structure are selected. To unselect a particular attribute, clear the **Use Attributes?** check box.

In a Business Function group, if a business function could be executed multiple times (for example a business function to create sales order lines) check the **Use Looping?** check box.

9. Click the business function icon again (as in step 6 above) and choose another business function. You can select multiple business functions to belong to the group.
10. Expand the attributes for the business function and select them as required.

    Click **Apply** to save the configuration of this Business Function Group.

- To rearrange the order of the Business Functions within the Business Function Group, use the small icons with the yellow up and down arrows to the right of the data structure symbol.
- To delete a business function, highlight the function to be deleted and click the icon with the red delete mark.

11. Next, configure the `Schema` tab setting. See Schema Tab on page 141 for details.

12. Save the project and exit TIBCO Designer.
13. Create the run-time adapter property file and add the project name and adapter configuration name. See Chapter 6, Deploying and Starting an Adapter Using TIBCO Administrator, on page 149 for details.

Configuration Tasks for a Request-Response Service

After configuring the adapter, you can add request-response services to it. For information on configuring the adapter, see Configuration Tasks for an Adapter on page 114.

1. Open the `Adapter Services` folder in the project panel of the adapter.
2. Add a Request-Response service to the adapter configuration by dragging the service icon from the palettes panel and dropping it in the design panel.
3. Use the `Configuration` tab to rename the service and set the transfer parameters. Refer to Configuration Tab on page 142.

When you rename a service, you will be prompted to regenerate the endpoints to ensure connection with the J. D. Edwards OneWorld Xe enterprise server.

4. In the BusinessFunctions Tab, select a Category, either Business Function, Business Function Group or XML Query.

If you are choosing either Business Function or Business Function Group, see Configuring the Subscription Service for a Business Function on page 117 or Configuring the Subscription Service for a Business Function Group on page 117.

5. If you have chosen XML query, you must choose a query type.
6. The three types of queries are:
   - `ListCreate`: You specify selection criteria, and ListCreate retrieves records from the database tables or views, formats the records, and stores them in temporary work files on the J. D. Edwards OneWorld Xe enterprise server.
   - `ListRetrieve`: With ListRetrieve, you read the formatted records from the work files created by the ListCreate operation. You can specify the number of records to be retrieved at a time. This feature is useful when a large number of records exist in the work files. By specifying only a limited number of records to be retrieved at a time, you can limit the message size that is transmitted on the network.
   - `ListDelete`: Deletes the temporary work file created by the ListCreate operation.
7. Choose either **Table**, **View** or **TC (Table Conversion)** settings.
8. Click the pen icon to see a list of the tables, views, or Table Conversions available in J. D. Edwards OneWorld Xe.
9. Select a Table/View/TC from the drop-down and click **OK**.

You can only add one table or view or TC. All attributes will be selected by default.

10. Click **Apply** to save the configuration of this XML query.
11. Next, configure the `Schema` tab setting. See Schema Tab on page 147 for details.
12. Save the project and exit TIBCO Designer.
13. Create the run-time adapter property file and add the project name and adapter configuration name. See Chapter 6, Deploying and Starting an Adapter Using TIBCO Administrator, on page 149 for details.

Adapter Instance Fields

The following tabs can be used to define an adapter instance:

Configuration Tab

Instance Name

Use the default name or replace it with a name of your choice.

- An instance name must use alphanumeric characters. An underscore (_) character can be used. The entire instance name must be less than 80 characters. The space character cannot be used in an instance name.
- An instance name cannot use global variables.
- An instance name must be unique with respect to other adapter instances in the project.
- Each instance name must be unique per adapter within a project even if each instance is defined in a different folder. That is, configuring same-named adapter instances in different folders will not make their names unique.

When you create an adapter instance, the palette automatically creates several resources for it. The names of these resources derive from the name of the instance they belong to. Changing the adapter instance name results in an automatic regeneration of the resources names. If you manually modify any resource name, that particular name will *not* be automatically regenerated next time your rename the adapter instance.

Description

Provide information about the adapter instance that you want stored in the project. The field is optional.

Version

The version string indicates the ActiveEnterprise (AE) format in which the adapter instance is saved. An adapter instance can be saved in AE 4.0 or AE 5.1 format.

When a new adapter instance is created in TIBCO Designer 5.x, the version string is set to `AE Version 5.1`. When a 4.x adapter instance is opened in Designer 5.x, the `Version` field is set to `AE Version 4.0`.

- If a 4.x adapter instance is to be run against a 4.x run-time adapter, the instance must be saved with the `Version` field set to `AE Version 4.0`.

  If you are using TIBCO Designer 5.x to modify 4.x adapter instances, change only features supported by the 4.x. run-time adapter and use the validation utility to verify the instance before deploying the project. The validation utility scans the project and returns warnings if any 5.x features are defined for 4.x adapter instances. Invoke the utility from the `Project>Validate Project for Deployment` menu command in Designer.

- If a 4.x adapter instance is to be run against a 5.x run-time adapter, the `Version` field should be set to `AE Version 5.1`.

  To change versions, click the `Change Version` button.

Message Filter

Specify a message filter, if you have configured a message filter resource for use with the adapter. The filter allows you to manipulate incoming and outgoing data before sending it on the network or handing it to the target application. Filters can be written using the TIBCO Adapter SDK. See the *TIBCO Adapter SDK Programmer's Guide* for information about writing a message filter.

Show All Tabs

Check this box to display additional tabs for configuring advanced options.

Design-time Connection Tab

Many of the following fields make use of global variables. Click the `Global Variables` tab in the project panel to enter a value for a global variable.

Server Name

Specify the name of the machine on which the J. D. Edwards OneWorld Xe application (enterprise) server is running.

Port Number

Specify the port that will be used for connection between J. D. Edwards OneWorld Xe and TIBCO Adapter for J. D. Edwards OneWorld Xe.

Environment

Specify the environment name in J. D. Edwards OneWorld Xe that will be used by the adapter instance.'

Username

User account name that is used by the adapter to connect to J. D. Edwards OneWorld Xe.

Password

Specify the password for the account used by the adapter to connect to J. D. Edwards OneWorld Xe.

Remember Password

If this field is not checked, the password must be entered each time the project is opened. If it is checked, the password will be stored in the project repository.

InterOp Jar Directory

Specify the directory where GenJava jar files for business functions are installed. These are the class files used by the adapter palette to build metadata information. You build these files manually, and copy them to the InterOp Jar Directory.

Configuration File

Specify the location of the `jdeinterop.ini` file.

Use Design-time Connection for Run-time

Select this check box to copy all the design-time connection parameters as runtime connection parameters.

Test Connection

Click to test the validity of the connection information that you specified for the adapter instance.

Run-time Connection Tab

Initial Number of Connections

Specify the number of connections that should be made to the adapter when the adapter starts up.

Maximum Number of Connections

Specify the maximum number of connections that can be made to the adapter.

Connection Idle Time (milliseconds)

Specify the time, in milliseconds, for which a connection is allowed to be idle.

Maximum Number of Reconnect Attempts

Specify the total number of reconnection attempts to make before the run-time adapter or adapter service is stopped. A value of -1 means reconnection attempts will continue indefinitely.

Number of Reconnect Attempts Before Suspending Impacted Service(s)

Specify the number of reconnection attempts to make before suspending a run-time adapter or adapter service. The `Adapter Termination Criteria` field value determines whether the adapter or an adapter service is stopped.

Interval between Reconnect Attempts (milliseconds)

Specify the time interval in milliseconds to elapse between each reconnection attempt.

Adapter Termination Criteria (after max number of reconnect attempts)

If your adapter supports only a single connection to a vendor application or database, either choice results in the same behavior. The adapter stops if the service loses its connection.

If your adapter supports multiple connections, select from the drop-down list:

- `When Any Service is Suspended` stops the adapter if any one service has been unable to re-establish connection after the specified number reconnection attempts.
- `When All Services Are Suspended` stops the adapter only when all services have been suspended. That is, only the adapter service that cannot reconnect

is stopped. Other adapter services that are connected continue to function normally.

General Tab

This tab has no fields.

Encoding for the adapter can be changed by modifying the `[SERVER]/codePage` property in the configuration file specified in the Run-time Connection tab.

Multithreading Tab

Inbound Thread Count

Specifies the number of inbound threads the application will have at runtime. If there are no Inbound services configured, that is, no subscription or request-response services, the value of this field is ignored.

Logging Tab

Use Advanced Logging

When `Use Advanced Logging` is not selected (the default), you can set two standard output destinations (sinks) for trace messages and set the tracing level for the roles selected.

When `Use Advanced Logging` is selected, you have complete control on selecting the destinations and associating desired roles with each of the destinations.

To create and configure the sinks, select the log sinks folder under the `Advanced` folder in the project panel.

To create sinks, drag and drop the `Generic log sink` icon from the palette panel into the design panel. From the configuration panel, select the sink type. The following are the sink types available:

- File
- Hawk
- Network
- STDIO

When File and STDIO sinks are created from the `Generic log sink` they offer further configuration options. For the File sink, the file limit, file count, and the option to append or overwrite can be specified. When created by default, this is set to 30000 bytes, 3 and append mode respectively. For the STDIO sink, the option to write to `stdout` or `stderr` can be selected. When created by default, `stdout` is selected.

The Hawk sink uses the hawk session, created and used by the adapter for monitoring purposes, to send tracing messages to the TIBCO Hawk monitor or Display. For details on Hawk sessions, see Using Global Variables on page 160. The configuration for the Hawk sink involves specifying the `MicroAgent Name` that must be specified in the configuration panel.

The Network sink is used to publish tracing message on TIBCO Rendezvous. The configuration for the network sink involves specifying the session, and the subject on which the trace messages needs to be published.

For all the sinks, optionally the name and description for the sinks can be provided.

Log to Standard I/O

(STDIO Sink) When selected, trace messages are displayed in the command prompt window where the adapter is started. When not selected, trace messages do not display in the window.

Log File

Specify the name of the log file (log sink) to which trace messages are written. Global variables can be used to specify the location of the log file. See Using Global Variables on page 160 for more information.

The roles available are Info, Debug, Warning, and Error messages. The trace message generated depends on the roles selected. Turning on the roles can affect the performance of the adapter. Therefore, it is recommended that you turn on the required roles only.

Log Info/Debug/Warning/Error Messages

Trace messages of the selected level(s) will be collected in the named log sink. You can configure what levels of trace messages you want logged, and where trace messages are sent. There are three types of logs (log sinks) that you can configure to hold trace messages, corresponding to three levels (roles) of trace messages, Information, Warning and Error. A fourth level of trace messages, Debug, is reserved and should not be enabled unless requested by the TIBCO Product Support Group. This option writes a lot of information to the log file and significantly reduces the speed of the adapter.

Startup Tab

Show Startup Banner

Select this check box to display the startup banner. The startup banner displays the run-time adapter version, the infrastructure version on which the adapter is built, and copyright information in the console window when the adapter is started.

Monitoring Tab

Many of the following fields make use of global variables. Click the `Global Variables` tab in the project panel to enter a value for a global variable.

Enable Standard Microagent

Allows you to turn on or off the standard TIBCO Hawk Microagent. The way to turn it on or off is also configurable. By clicking the `globe` icon, a standard check box or text value (true or false) can be used to turn the standard microagent on or off.

Standard Microagent Name

This field is predefined and cannot be changed.

Enable Class Microagent

Allows you to turn on or off the instance or class specific standard TIBCO Hawk Microagent. The way to turn it on or off is also configurable. By clicking the `globe` icon, a standard check box or text value (true or false) can be used to turn the class microagent on or off.

Class Microagent Name

This is the name for the class microagent that will be registered with the TIBCO Hawk system. In most cases the default value is used.

Default Microagent Session

Specifies the name of the TIBCO Rendezvous session that will be used by the standard, class, and custom microagents.

The session name and the corresponding session is automatically generated by TIBCO Designer. However, you can modify the session parameters if required. Navigate to the `Sessions` folder under the `Advanced` folder to modify the session parameters.

Make sure you have set the correct parameter value for the global variables that correspond to the TIBCO Hawk configuration. If the session parameters are not set properly, the microagents will not display in the TIBCO Hawk Display.

Adapter Services

After configuring an adapter instance, select one or multiple adapter services for the instance. The following sections describe the services and fields that are available to the adapter.

- Publication Service Fields on page 131
- Subscription Service Fields on page 136
- Request-Response Service Fields on page 142

Publication Service Fields

The following tabs are available:

Configuration Tab

Name

You can use the default name or replace it with a name of your choice.

- A service name must use alphanumeric characters. An underscore (_) character can be used. The entire instance name must be less than 80 characters. The space character cannot be used in an instance name.
- A service name cannot use global variables.

Description

Provide information about the adapter service that you want stored in the project. The field is optional.

Transport Type

Select the transport to be used by the run-time adapter, JMS or Rendezvous. After selecting the transport, the transport-specific configuration fields display.

The transport can be configured to use a trusted store and identity resource for use in SSL (Secure Sockets Layer) configurations. TIBCO Rendezvous sessions and JMS topics have an SSL configuration field which uses a dialog to perform SSL configuration.

To enable and configure SSL, in the `Project` panel, expand the `Advanced` folder, then expand the `Sessions` folder. Select the TIBCO Rendezvous session or JMS topic and click `Use SSL?`. The SSL configuration options are explained in the online help associated with the session dialog. Click the question mark to display the online help.

Transport Tab

Message Subject

This field displays only if `Rendezvous` is selected in the `Transport Type` field (under the `Configuration` tab).

By default a service uses a message subject that is generated using the `Domain` and `Deployment` global variables, the adapter acronym, the adapter instance name and the service name. If you use this default subject, make sure the values for `Domain` and `Deployment` are not empty. You can type a TIBCO Rendezvous subject name different from the default in this field. See *TIBCO Rendezvous Concepts* for information about specifying subject names.

Destination

This field displays only if `JMS` is selected in the `Transport Type` field (under the `Configuration` tab).

By default a service uses a dynamic destination that is generated using the `Domain` and `Deployment` global variables, the adapter acronym, the adapter instance name and the service name. If you use this default dynamic destination, make sure the values for `Domain` and `Deployment` are not empty. You can override the default dynamic destination by specifying the static destination in this field. The static destination must be defined on the JMS server before it can be used by the run-time adapter. See the *TIBCO Enterprise Message Service User's Guide* for information about destinations.

Quality of Service

If Rendezvous is selected as the transport type, select:

- `Certified`

  Guarantees that every certified message reaches its intended recipient in the order sent. The message can be sent across network boundaries, and if a network fails, delivery attempts continue until delivery succeeds or until the message's time limit expires. This is often called certified message delivery.

  If certified message delivery is used, data is stored in a ledger file. The size of the ledger depends on several factors, the most important of which is the retention rate of stored data. That is, the ledger grows fastest in response to the cumulative length of undeliverable messages. You must ensure that sufficient disk space is available for the expected size of the ledger.

- `Reliable`

Ensures that each multicast or broadcast message is received as long as the physical network and packet recipients are working, and that the loss of a message is detected. This choice can compensate for brief network failures because it can retransmit a message on request if the first attempt failed. This choice is appropriate when message delivery is expected but some loss can be tolerated.

Wire Format

Services must use the same wire format to exchange data.

- `Rendezvous Message` (Rendezvous only)

  Control information for validation is *not* sent in the message. If you use this format, the adapter is compatible with adapters not developed with TIBCO Adapter SDK.

- `XML Message` (Rendezvous or JMS)

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

- `ActiveEnterprise Message` (Rendezvous only)

  Control information for validation is sent in the message. If no control information is included, an exception is returned to the subscription service. ActiveEnterprise standard wire format provides class information and packing rules for the TIBCO Adapter SDK set of data types. This format allows ActiveEnterprise components to perform extra validation on messages sent or received.

  See the *TIBCO Adapter SDK Programmer's Guide* for details about the control information generated and sent with ActiveEnterprise messages.

Connection Factory Type

- `Topic` (JMS only)

  A message published to a topic is broadcast to one or more subscription services. All messages published to the topic are received by all services that have subscribed to the topic. This messaging model is known as publish-subscribe.

- `Queue` (JMS only)

  A message sent to a queue is consumed by one and only one receiver. Each message has only one receiver though multiple receivers may connect to the queue. The first receiver to access the queue gets the message. The other receivers do not. This messaging model is known as point-to-point.

Delivery Mode

- `Persistent` (JMS only)

  In general, a message marked persistent will be available to a JMS client even if the JMS server goes down.

- `Non Persistent` (JMS only)

  A message marked non persistent will not be available to a JMS client if the JMS server goes down.

Messages sent with the persistent delivery mode are always written to persistent storage, except when they are published to a topic that has no durable subscribers. When a topic has no durable subscribers, there are no subscribers that need messages resent in the event of a server failure. Therefore, messages do not need to be saved, and performance is improved because disk I/O is not required.

The semantics for these fields are somewhat more complex than the explanation given here. See the *TIBCO Enterprise for JMS User's Guide* for more information.

Session Reference

Every adapter instance can have one or more sessions configured for it. Sessions encapsulate stateful connections to TIBCO Rendezvous and other messaging sources. The session object shown in this field is initially supplied by the adapter, depending on the Quality of Service selected. You can change the session by browsing for it in the project panel.

Endpoint Reference

You can go to the referenced endpoint to edit its properties. Endpoint reference objects are explained in the TIBCO Designer *Palette Reference*.

Real-time Event Tab

For the Real-Time Event Tab, you import events and configure data objects as described in After configuring the adapter, as described in Configuration Tasks for an Adapter on page 114, you can configure a publication service. on page 116.

Event, Data Structure and Attributes

The user can select an event from the list of available events. The list of available events retrieved from the real-time events configured in J. D. Edwards OneWorld Xe system. Each real-time event has one or more data structures associated with it. Each data structure defines and groups attributes for that event. If a real-time

event contains multiple data structures, such an event is called a **container** event.

This column displays the name of the real-time event, the name data structures in that event and the name of attributes in each data structure in the event.

Use Looping

For a **container** type real-time event, a data structure in that event can have multiple occurances when the business transaction is published in real-time events. For example, in case of a sales order real-time event, the data structure for sales order lines can occur multiple times if the sales order has multiple items. In such cases, that particular data structure should be marked as **looping** when configuring the real-time event in the adapter. **The Use Looping?** check box is displayed only for a data structure in an event.

Use Attribute

Indicates whether the attribute in the data structure should be used for output. If the user unchecks this check box, the attribute will not be used/output in the ActiveEnterprise message published by the adapter.

Attribute Wanted

Displays the name of the attribute if the **Use Attribute?** check box is selected. If the **Use Attribute?** check box is not selected for a attribute, this column will be blank for that attribute.

Schema Tab

Class Reference

The class reference for a service is set automatically.

Attempting to validate the configuration by clicking **Project > Validate for Deployment** before a schema is attached to the service may result in an error.

Once the schema has been specified for a service, you can view its attributes by selecting the `AESchemas` folder in the project panel. You can expand the field names to view the data types and field attributes. For more information about schema, see the *TIBCO Designer User's Guide*.

Subscription Service Fields

The following tabs are available:

Configuration Tab

Name

You can use the default name or replace it with a name of your choice.

- A service name must use alphanumeric characters. An underscore (_) character can be used. The entire instance name must be less than 80 characters. The space character cannot be used in an instance name.
- A service name cannot use global variables.

Description

Provide information about the adapter service that you want stored in the project. The field is optional.

Transport Type

Select the transport to be used by the run-time adapter, JMS or Rendezvous. After selecting the transport, the transport-specific configuration fields display.

The transport can be configured to use a trusted store and identity resource for use in SSL (Secure Sockets Layer) configurations. TIBCO Rendezvous sessions and JMS topics have an SSL configuration field which uses a dialog to perform SSL configuration.

To enable and configure SSL, in the `Project` panel, expand the `Advanced` folder, then expand the `Sessions` folder. Select the TIBCO Rendezvous session or JMS topic and click `Use SSL?`. The SSL configuration options are explained in the online help associated with the session dialog. Click the question mark to display the online help.

Transport Tab

Message Subject

This field displays only if `Rendezvous` is selected in the `Transport Type` field (under the `Configuration` tab).

By default a service uses a message subject that is generated using the `Domain` and `Deployment` global variables, the adapter acronym, the adapter instance name and the service name. If you use this default subject, make sure the values for `Domain` and `Deployment` are not empty. You can type a TIBCO Rendezvous subject name different from the default in this field. See *TIBCO Rendezvous Concepts* for information about specifying subject names.

Destination

This field displays only if `JMS` is selected in the `Transport Type` field (under the `Configuration` tab).

By default a service uses a dynamic destination that is generated using the `Domain` and `Deployment` global variables, the adapter acronym, the adapter instance name and the service name. If you use this default dynamic destination, make sure the values for `Domain` and `Deployment` are not empty. You can override the default dynamic destination by specifying the static destination in this field. The static destination must be defined on the JMS server before it can be used by the run-time adapter. See the *TIBCO Enterprise for JMS User's Guide* for information about destinations.

Quality of Service

If Rendezvous is selected as the transport type, select:

- `Certified`

  Guarantees that every certified message reaches its intended recipient in the order sent. The message can be sent across network boundaries, and if a network fails, delivery attempts continue until delivery succeeds or until the message's time limit expires. This is often called certified message delivery.

  If certified message delivery is used, data is stored in a ledger file. The size of the ledger depends on several factors, the most important of which is the retention rate of stored data. That is, the ledger grows fastest in response to the cumulative length of undeliverable messages. You must ensure that sufficient disk space is available for the expected size of the ledger.

- `Reliable`

Ensures that each multicast or broadcast message is received as long as the physical network and packet recipients are working, and that the loss of a message is detected. This choice can compensate for brief network failures because it can retransmit a message on request if the first attempt failed. This choice is appropriate when message delivery is expected but some loss can be tolerated.

- `Distributed Queue`

  Distributed queue includes a group of cooperating transport objects, each in a separate process. Each transport object is called a member. To balance the transmission load among servers, the adapter can use distributed queues for *one-of-n* delivery of messages to a group of servers. Each member of a distributed queue listens for the same subject using the TIBCO Rendezvous Distributed Queue listener objects. Even though many members listen for each inbound message (or task), only one member processes the message. For details on distributed queues, see *TIBCO Rendezvous Concepts*.

  Load balancing for the processing of TIBCO Rendezvous certified messages is supported by using distributed queuing. The messages from TIBCO Rendezvous are distributed equally among all instances that belong to the same group. This distributes the message load over several adapter instances. However, the order in which messages are sent to the application is not guaranteed.

Wire Format

Services must use the same wire format to exchange data.

- `Rendezvous Message` (Rendezvous only)

  Control information for validation is *not* sent in the message. If you use this format, the adapter is compatible with adapters not developed with TIBCO Adapter SDK.

- `XML Message` (Rendezvous or JMS)

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

- `ActiveEnterprise Message` (Rendezvous only)

  Control information for validation is sent in the message. If no control information is included, an exception is returned to the subscription service. ActiveEnterprise standard wire format provides class information and packing rules for the TIBCO Adapter SDK set of data types. This format allows ActiveEnterprise components to perform extra validation on messages sent or received.

See the *TIBCO Adapter SDK Programmer's Guide* for details about the control information generated and sent with ActiveEnterprise messages.

Connection Factory Type

- `Topic` (JMS only)

  A message published to a topic is broadcast to one or more subscribers. All messages published to the topic are received by all services that have subscribed to the topic. This messaging model is known as publish-subscribe.

- `Queue` (JMS only)

  A message sent to a queue is consumed by one and only one receiver. Each message has only one receiver though multiple receivers may connect to the queue. The first receiver to access the queue gets the message. The other receivers do not. This messaging model is known as point-to-point.

Delivery Mode

- `Durable` (JMS only)

  If a subscription service is marked `durable`, it indicates that messages need to be resent on the configured topic or queue, if the JMS server goes down.

- `Non Durable` (JMS only)

  If a subscription service is marked `non-durable`, it indicates that messages will not be resent on the configured topic or queue, if the JMS server goes down.

Messages sent with the persistent delivery mode are always written to persistent storage, except when they are published to a topic that has no durable subscribers. When a topic has no durable subscribers, there are no subscribers that need messages resent in the event of a server failure. Therefore, messages do not need to be saved, and performance is improved because disk I/O is not required.

The semantics for these fields are somewhat more complex than the explanation given here. See the *TIBCO Enterprise for JMS User's Guide* for more information.

Session Reference

Every adapter instance can have one or more sessions configured for it. Sessions encapsulate stateful connections to TIBCO Rendezvous and other messaging sources. The session object shown in this field is initially supplied by the adapter, depending on the Quality of Service selected. You can change the session by browsing for it in the project panel.

Endpoint Reference

You can go to the referenced endpoint to edit its properties. Endpoint reference objects are explained in the TIBCO Designer *Palette Reference*.

Business Functions Tab

For the BusinessFunctions Tab, you choose BusinessFunctions as described in Configuring the Subscription Service for a Business Function on page 117.

Select a Category

Select a Business Function or a Business Function Group as the category for the service.

Business Functions

The user can select a business function from the list of available business functions. The list of available business functions is retrieved from list of all business functions deployed on the J. D. Edwards OneWorld Xe enterprise server. Each business function has a data structure associated with it. This data structure defines input/output parameters (attributes) for that business function.

This column displays the name of the business function group, the name of business functions in the group and the names of attributes for each business function in the group.

Use Looping

A business function group may contain one or more business functions. A business function in a business function group may be invoked multiple times. For example, in case of a sales order creation function, the business function for creating sales order lines may be called multiple times. When a business function will be called multiple times, it should be marked as **looping** when configuring the business function group in the adapter by checking the **Use Looping?** check box.

The **Use Looping?** check box is displayed only for a business function in a business function group

Use Attribute

Indicates whether the attribute in the data structure should be used as an input parameter. If the user unchecks this check box, the attribute will not be used as an input parameter for the business function and will not be available in the ActiveEnterprise message schema used by the adapter.

Attribute Wanted

Displays the name of the attribute if the **Use Attribute?** check box is selected. If the **Use Attribute?** check box is not selected for a attribute, this column will be blank for that attribute.

Use Bind

The adapter has functionality to pass output parameter values of any business function to subsequent business functions. If a parameter in a business function is used as a input parameter for execution of subsequent business functions, that parameter should be marked as a bind parameter by checking the **Use Bind?** check box.

Bind Identifier

If a parameter in a business function uses the value of a preceding business function parameter, that preceding business function parameter is assigned here. A drop-down box is displayed in this column. The drop-down box will have list of all parameters in preceding business functions that are marked as **bind** parameters

Schema Tab

Class Reference

The class reference for a service is set automatically.

Attempting to validate the configuration by clicking **Project > Validate for Deployment** before a schema is attached to the service may result in an error.

Once the schema has been specified for a service, you can view its attributes by selecting the `AESchemas` folder in the project panel. You can expand the field names to view the data types and field attributes. For more information about schema, see the *TIBCO Designer User's Guide*.

Request-Response Service Fields

The following tabs are available:

Configuration Tab

Name

You can use the default name or replace it with a name of your choice.

- A service name must use alphanumeric characters. An underscore (_) character can be used. The entire instance name must be less than 80 characters. The space character cannot be used in an instance name.
- A service name cannot use global variables.

Description

Provide information about the adapter service that you want stored in the project. The field is optional.

Transport Type

Select the transport to be used by the run-time adapter, JMS or Rendezvous. After selecting the transport, the transport-specific configuration fields display.

The transport can be configured to use a trusted store and identity resource for use in SSL (Secure Sockets Layer) configurations. TIBCO Rendezvous sessions and JMS topics have an SSL configuration field which uses a dialog to perform SSL configuration.

To enable and configure SSL, in the `Project` panel, expand the `Advanced` folder, then expand the `Sessions` folder. Select the TIBCO Rendezvous session or JMS topic and click `Use SSL?`. The SSL configuration options are explained in the online help associated with the session dialog. Click the question mark to display the online help.

Transport Tab

Message Subject

This field displays only if `Rendezvous` is selected in the `Transport Type` field (under the `Configuration` tab).

By default a service uses a message subject that is generated using the `Domain` and `Deployment` global variables, the adapter acronym, the adapter instance name and the service name. If you use this default subject, make sure the values for `Domain` and `Deployment` are not empty. You can type a TIBCO Rendezvous subject name different from the default in this field. See *TIBCO Rendezvous Concepts* for information about specifying subject names.

Destination

This field displays only if `JMS` is selected in the `Transport Type` field (under the `Configuration` tab).

By default a service uses a dynamic destination that is generated using the `Domain` and `Deployment` global variables, the adapter acronym, the adapter instance name and the service name. If you use this default dynamic destination, make sure the values for `Domain` and `Deployment` are not empty. You can override the default dynamic destination by specifying the static destination in this field. The static destination must be defined on the JMS server before it can be used by the run-time adapter. See the *TIBCO Enterprise for JMS User's Guide* for information about destinations.

Quality of Service

If Rendezvous is selected as the transport type, select:

- `Certified`

  Guarantees that every certified message reaches its intended recipient in the order sent. The message can be sent across network boundaries, and if a network fails, delivery attempts continue until delivery succeeds or until the message's time limit expires. This is often called certified message delivery.

  If certified message delivery is used, data is stored in a ledger file. The size of the ledger depends on several factors, the most important of which is the retention rate of stored data. That is, the ledger grows fastest in response to the cumulative length of undeliverable messages. You must ensure that sufficient disk space is available for the expected size of the ledger.

- `Reliable`

Ensures that each multicast or broadcast message is received as long as the physical network and packet recipients are working, and that the loss of a message is detected. This choice can compensate for brief network failures because it can retransmit a message on request if the first attempt failed. This choice is appropriate when message delivery is expected but some loss can be tolerated.

- `Distributed Queue`

  Distributed queue includes a group of cooperating transport objects, each in a separate process. Each transport object is called a member. To balance the transmission load among servers, the adapter can use distributed queues for *one-of-n* delivery of messages to a group of servers. Each member of a distributed queue listens for the same subject using the TIBCO Rendezvous Distributed Queue listener objects. Even though many members listen for each inbound message (or task), only one member processes the message. For details on distributed queues, see *TIBCO Rendezvous Concepts*.

  Load balancing for the processing of TIBCO Rendezvous certified messages is supported by using distributed queuing. The messages from TIBCO Rendezvous are distributed equally among all instances that belong to the same group. This distributes the message load over several adapter instances. However, the order in which messages are sent to the application is not guaranteed.

Wire Format

Services must use the same wire format to exchange data.

- `XML Message` (JMS only)

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

- `ActiveEnterprise Message` (Rendezvous only)

  Control information for validation is sent in the message. If no control information is included, an exception is returned to the subscription service. ActiveEnterprise standard wire format provides class information and packing rules for the TIBCO Adapter SDK set of data types. This format allows ActiveEnterprise components to perform extra validation on messages sent or received.

  See the *TIBCO Adapter SDK Programmer's Guide* for details about the control information generated and sent with ActiveEnterprise messages.

Connection Factory Type

- `Topic` (JMS only)

  A message published to a topic is broadcast to one or more subscribers. All messages published to the topic are received by all services that have subscribed to the topic. This messaging model is known as publish-subscribe.

- `Queue` (JMS only)

  A message sent to a queue is consumed by one and only one receiver. Each message has only one receiver though multiple receivers may connect to the queue. The first receiver to access the queue gets the message. The other receivers do not. This messaging model is known as point-to-point.

Delivery Mode

- `Durable` (JMS only)

  If a subscription service is marked `durable`, it indicates that messages need to be resent on the configured topic or queue, if the JMS server goes down.

- `Non Durable` (JMS only)

  If a subscription service is marked `non-durable`, it indicates that messages will not be resent on the configured topic or queue, if the JMS server goes down.

Messages sent with the persistent delivery mode are always written to persistent storage, except when they are published to a topic that has no durable subscribers. When a topic has no durable subscribers, there are no subscribers that need messages resent in the event of a server failure. Therefore, messages do not need to be saved, and performance is improved because disk I/O is not required.

The semantics for these fields are somewhat more complex than the explanation given here. See the *TIBCO Enterprise for JMS User's Guide* for more information.

Session Reference

Every adapter instance can have one or more sessions configured for it. Sessions encapsulate stateful connections to TIBCO Rendezvous and other messaging sources. The session object shown in this field is initially supplied by the adapter, depending on the Quality of Service selected. You can change the session by browsing for it in the project panel.

Endpoint Reference

You can go to the referenced endpoint to edit its properties. Endpoint reference objects are explained in the TIBCO Designer *Palette Reference*.

Business Functions Tab

Select a Category

Select a Business Function, a Business Function Group or an XML Query List as the category for the service.

Business Functions

The user can select a business function from the list of available business functions. The list of available business functions is retrieved from list of all business functions deployed on the J. D. Edwards OneWorld Xe enterprise server. Each business function has a data structure associated with it. This data structure defines input/output parameters (attributes) for that business function.

This column displays the name of the business function group, the name of business functions in the group and the names of attributes for each business function in the group.

XML Query Lists

The Request-Response operations for an XML Query List can be `ListCreate`, `ListRetrieve` or `ListDelete`. These operations can retrieve data from J. D. Edwards OneWorld Xe tables or views. These operations can also run the selected Table Conversions and return the Table Conversion output. You can select the table, view or TC(Table Conversion) name from the list of all the tables/views/TCs available on the J. D. Edwards OneWorld Xe system.

Use Looping

A business function group may contain one or more business functions. A business function in a business function group may be invoked multiple times. For example, in case of a sales order creation function, the business function for creating sales order lines may be called multiple times. When a business function will be called multiple times, it should be marked as **looping** when configuring the business function group in the adapter by checking the **Use Looping?** check box.

The **Use Looping?** check box is displayed only for a business function in a business function group.

Use Attribute

Indicates whether the attribute in the data structure should be used as an input parameter. If the user unchecks this check box, the attribute will not be used as an input parameter for the business function and will not be available in the ActiveEnterprise message schema used by the adapter.

Attribute Wanted

Displays the name of the attribute if the **Use Attribute?** check box is selected. If the **Use Attribute?** check box is not selected for a attribute, this column will be blank for that attribute.

Use Bind

The adapter has functionality to pass output parameter values of any business function to subsequent business functions. If a parameter in a business function is used as a input parameter for execution of subsequent business functions, that parameter should be marked as a bind parameter by checking the **Use Bind?** check box.

Bind Identifier

If a parameter in a business function uses the value of a preceding business function parameter, that preceding business function parameter is assigned here. A drop-down box is displayed in this column. The drop-down box will have list of all parameters in preceding business functions that are marked as **bind** parameters.

Schema Tab

Class Reference

The class reference for a service is set automatically.

Attempting to validate the configuration by clicking **Project > Validate for Deployment** before a schema is attached to the service may result in an error.

Once the schema has been specified for a service, you can view its attributes by selecting the `AESchemas` folder in the project panel. You can expand the field names to view the data types and field attributes. For more information about schema, see the *TIBCO Designer User's Guide*.

Chapter 6 Deploying and Starting an Adapter Using TIBCO Administrator

This chapter provides an overview about deploying, starting, stopping, and monitoring adapter services using the TIBCO Administrator web interface.

Topics

Create an EAR File in TIBCO Designer

Generate an Enterprise Archive file (EAR) that contains information about the adapter services to deploy.

The EAR file contains information on what you wish to deploy. This could be one or more adapter services, one or more TIBCO BusinessWorks process engines, or both.

Building an archive creates the EAR file, which you can then deploy from TIBCO Administrator. If you make changes to the business processes or adapter services included in the archive, you need to rebuild the archive. Saving the project does not affect the archive.

In TIBCO Designer, follow these steps to create an EAR:

1. Configure the adapter services.
2. Drag and drop the `Enterprise Archive` resource from the palette panel to the design panel. If there are any configured adapter services in your project, an `Adapter Archive` resource becomes available in the palette panel.
3. Drag the `Adapter Archive` into the design panel and specify information in the `Configuration` tab, then click **Apply**.
4. Go to the `Enterprise Archive` and click **Build Archive** to create the archive file.

See Also

See the *TIBCO Designer User's Guide* for more information about this procedure. The guide is available from the Designer `Help` menu.

Deploy the Project

Before deploying a project, the machine on which the adapter is installed must be part of a TIBCO administration domain. After you have installed the TIBCO Administration Server, any machine on which you install TIBCO Runtime Agent (required by an adapter) can be added to the administration domain. The TIBCO software installed on the machine is then visible and accessible via the TIBCO Administrator GUI.

When you deploy a project, startup scripts and other information about the different components are sent to the machines to which the components were assigned. The project data store and TIBCO Administration Server are updated with the deployed components.

To deploy a project:

1. Import the EAR file into TIBCO Administrator.
2. Assign adapter archives in the EAR file to adapters installed in the administration domain and likewise assign process archives to process engines.
3. Specify startup options for each adapter service.

Password Handling

At design-time, the adapter uses a password to connect to the backend application and fetch metadata. At run-time, the adapter uses a password to connect to the back-end application and interoperate with it. If you create a 4.x configuration using TIBCO Designer <version_num>, and use the configuration against a 4.x adapter version, some special considerations are required for security.

When deploying the adapter check the `Service` property of the global variable in the global variables section, then go to the `Advanced` tab of the adapter archive and set the password value under the `Run-Time Variables` section.

Do not set the password to type `Password` in the global variables section for adapter configurations that are set to `AE Version 4.0` or `AE Version 5.0` (in the `Configuration` tab `Version` field) or any intermediate version.

See Also

See the *TIBCO Administrator User's Guide* for an introduction to the TIBCO administration domain and detailed information about the above steps.

See the *TIBCO Administrator Server Configuration Guide* for fault tolerance information.

Start or Stop the Adapter

The TIBCO Administrator `Application Management` module allows you to start, and stop deployed applications.

To start an adapter service from the module:

1. In the Administrator GUI left pane, expand **Application Management >** *Application-Name* **> Service Instances**.
2. In the `Service Instance` panel, select the check box next to the adapter service.
3. Click the **Start Selected** button.

   The status changes from `Stopped` to `Starting up` to `Started`.
4. To stop the adapter service, click the **Stop Selected** button.

Do not use a TIBCO Rendezvous message to stop the run-time adapter. Instead use TIBCO Hawk.

See Also

See the *TIBCO Administrator User's Guide* for more information.

Monitor the Adapter

TIBCO Administrator offers a number of monitoring options.

- Specify alerts and TIBCO Hawk rulebases for each machine in the domain.
- Specify alerts and Hawk rulebases for each adapter service.
- View the log for each adapter service.

See Also

See the *TIBCO Administrator User's Guide* for information about configuring the above monitoring options.

Chapter 7 Advanced Topics

This chapter explains advanced topics for the TIBCO Adapter for J. D. Edwards OneWorld Xe.

Topics

Using the Adapter with a Revision Control System

TIBCO Designer supports revision control systems such as MicroSoft Visual SourceSafe and Perforce. If you are using a revision control system, you must manually add some configured resources to the revision control system and check in the resources when completing the instance configuration.

As part of service configuration, the adapter creates schema files in `root/AESchemas/ae/PeopleSoft8`. For example, if you configure a service in an adapter configuration `Instance1`, the following files are created:

```
Project_root/AESchemas/ae/adjdexe/Instance1.aeschema
Project_root/AESchemas/ae/adjdexe/Instance1/businessObjects.aesche
ma
```

When the project is saved and a revision control system has been specified, the adapter displays a warning that additional files were created and should be added to the revision control system. This warning appears only when the files are created for the first time. The warning displays a `Go To Resource` button that helps in navigating to the resource. You should use the `Multi-User>Add Resources to RCS` menu command to add these files to the revision control system.

For information about how to use the Multi-User feature in TIBCO Designer, refer to the *TIBCO Designer User's Guide*.

Copy, Cut, Paste and Move Operations

To successfully copy and paste a service from adapter *Instance1* to *Instance2*, the adapter configuration and schema files for the *Instance2* must be checked out.

To successfully cut and paste a service from adapter *Instance1* to *Instance2*, the adapter configuration and schema files for both *Instance1* and *Instance2* must be checked out.

To successfully move a service from adapter *Instance1* to *Instance2*, the adapter configuration and schema files for both *Instance1* and *Instance2* must be checked out.

Regeneration When Moving, Copying and Pasting

- Default subjects are not regenerated to reflect the new instance name when a service is moved.
- Manually changed certified messaging and certified messaging queue ledger file names are regenerated to defaults when a service is moved, or copied and pasted to a new instance.

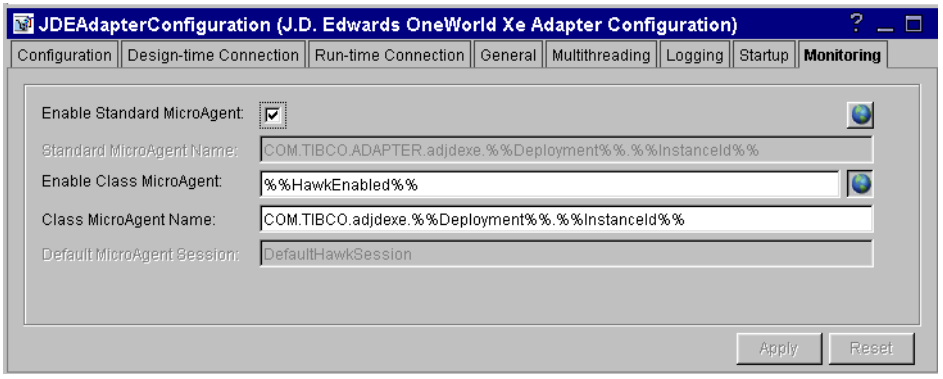
- If a service associated with a custom session is moved, or copied and pasted, the custom session is not moved, or copied and pasted. The session is regenerated as a default session.

Defining a TIBCO Hawk Session

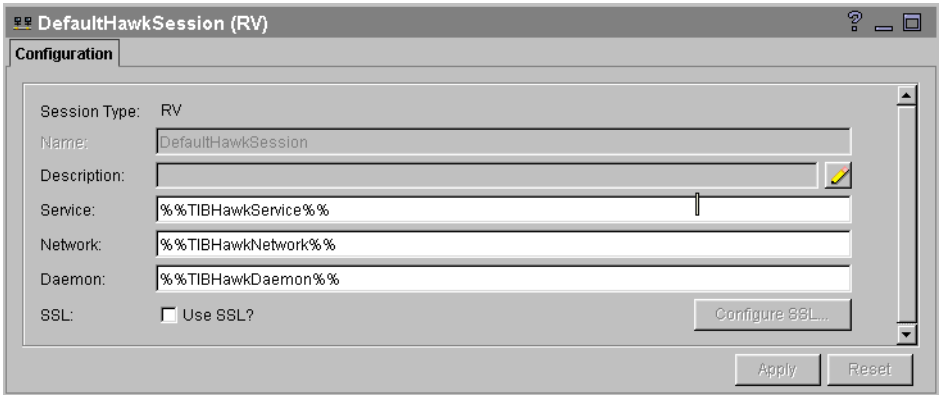
To use TIBCO Hawk to monitor the adapter you must first define a TIBCO Rendezvous session. Use the following steps to create the session.

1. In the project tree panel, click the `JDEA Adapter Configuration` icon defined for your adapter instance.
2. Select the `Show All Tabs` check box, then click the `Monitoring` tab.
3. The `Default MicroAgent Session` contains the name of the Hawk session:`DefaultHawkSession`. Use default setting for the other fields.

   The next figure shows the definition for a `Monitoring` tab.



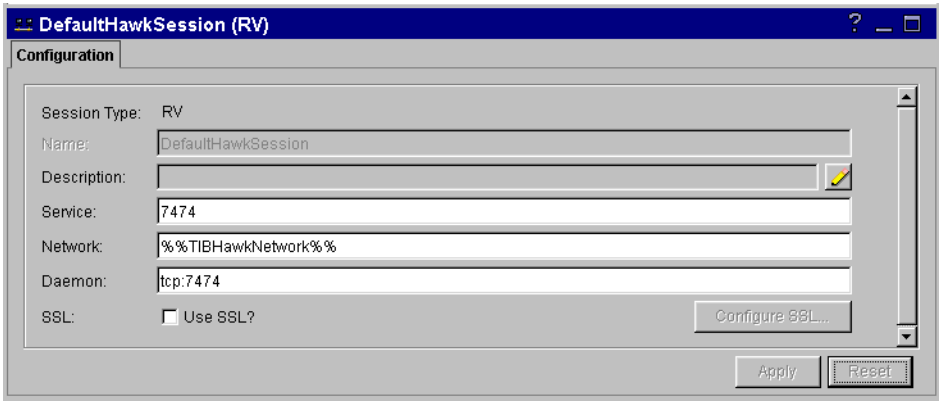
4. Open the `Advanced` folder for the adapter instance. Double-click the `Sessions` folder.
5. Double-click the `DefaultHawkSession` icon.



6. In `Service`, type **`7474`** (the default used by TIBCO Hawk) or modify the global variable by clicking the `Global Variables` tab.
7. In `Daemon`, type **`tcp:7474`** (the default used by TIBCO Hawk) or modify the global variable by clicking the `Global Variables` tab.

   Click **Apply** and save the project.

   The next figure shows the hawk session defined.



Using Global Variables

The variable substitution mechanism can override global variables predefined in the project in a restricted manner. Predefined variables can be viewed and set in TIBCO Designer. Variables are specified as %%VARNAME%% and cannot contain any white space.

Variable substitution allows you to accomplish the following.

- Substitute string variables specified in the project at startup time.
- Locally define the value for a variable for a specific project. The local value takes precedence over any global value.
- Specify the value for a variable in a properties file. This overrides the project repository and values set in code, but not variables set on the command line.
- Enforce the pre-defined variables listed in Predefined Global Variables on page 162.

  Variables can be used anywhere in the configuration and will be replaced by the locally-defined adapter instance.

Specifying Variables Using TIBCO Designer

Global variables provide an easy way to set defaults for use throughout your project. There are several ways in which they can be used:

- Define a variable using TIBCO Designer, then override the value for individual applications at deployment time using TIBCO Administrator. You can also override values for predefined variables, unless the GUI does not allow you to set them later.
- Predefine a variable using TIBCO Designer, then override the value for individual services (for example, publication service or TIBCO BusinessWorks process) at deployment time using TIBCO Administrator. The values you specify are then used at runtime. You can also override values for predefined variables, unless the GUI does not allow you to set them later.

For example, you could assign the value 7474 to the predefined global variable `RvDaemon`. You can then use the variable in different sessions in your adapter. If you wish to change the TIBCO Rendezvous daemon for your adapter, you can globally set it to a different value or override it from the command line.

To use global variables in your project, follow these steps:

1. In the project panel, select the `Global Variables` tab.

The project panel is updated to display all currently defined global variables. You now have these choices:

— To assign or change a variable value, select that region and triple-click the variable. The variable expands so you can change either the variable name or the variable value. Press Enter when you are done.

— To add a new global variable group, click the leftmost icon. Specify the name of the group, then press Enter. With the group icon selected, you can click the `abc` icon to add variables to the group.

— To add a global variable, click the `abc` icon. A new global variable item is added to the bottom of the list. Supply the variable name and, optionally, the value. Press Enter when you're done.

The global variable is now displayed in the global variables list.

2. When you want to use the global variable in the fields of a resource, enter the variable name surrounded by `%%` on both sides.

   When the project is deployed and the configured components are run, all occurrences of the global variable name are replaced with the global variable value (unless it was overridden in a way that had higher precedence).

A number of global variables are predefined. See Predefined Global Variables on page 162 for information. You may add definitions of any variables you need to the predefined variables.

Changing Global Variable Values at Runtime

You can change the value of a global variable when you deploy your project in TIBCO Administrator. See the section on modifying runtime variables in the *TIBCO Administrator User's Guide* for more information on using TIBCO Administrator.

You can also specify values for global variables when starting a process engine on the command line. To do this, specify the following as a command line argument when starting the process engine:

`-tibco.clientVar.`*<variablePathAndName>* *<value>*

where *variablePathAndName* is the name of the variable you wish to set, including the path to the variable if it is contained in a folder. *value* is the value you wish to set the variable to. For example, if you have a global variable named `item1` contained in a folder named `myGroup` and you wish to set its value to `500`, add the following argument to the command line when starting the process engine:

```
-tibco.clientVar.myGroup/item1 500
```

Predefined Global Variables

The next table lists and explains the predefined global variables. Some global variables are automatically used within the system when an adapter instance is configure. For example, the RV Session shown above uses the value defined for the `RvService`, `RvNetwork` and `RvDaemon` global variables

For further information on the connection variables, see Connection Settings on page 165.

Table 8 Predefined Global Variables

Variable	Description
`Deployment`	Defaults to the TIBCO Designer project name. This global variable is used by the system to partially define the subject name defined for a service.
`DirLedger`	Specifies the path name of the TIBCO Rendezvous certified messaging ledger file. The default is the root installation directory.
`DirTrace`	Specifies the path name for log file used by the adapter. The default is the root installation directory.
`Domain`	The default value for file-based local projects is `MyDomain`. The value for server-based projects is the domain to which the project was saved.
`Env`	Included for backward compatibility. Do not use.
`HawkEnabled`	Indicates whether TIBCO Hawk is used to monitor the adapter. True indicates that a Hawk microagent is defined for the adapter. False indicates the microagent is not to be used.
JmsSslProviderUrl	Specifies where the JMS SSL daemon is located.

Table 8 Predefined Global Variables

Variable	Description
`JmsProviderUrl`	Specifies where the JMS server is located. Setting this value mostly makes sense in early stages of a project, when only one JMS server is used.
`RemoteRvDaemon`	TIBCO Rendezvous routing daemon (`rvrd`) to be used. See *TIBCO Administrator Server Configuration Guide* for details about setting up a domain using `rvrd`.
`RvDaemon`	TIBCO Rendezvous daemon. Sessions use this daemon to establish communication. The default value is `7500`.
`RvNetwork`	TIBCO Rendezvous network. This variable need only be set on computers with more than one network interface. If specified, the TIBCO Rendezvous daemon uses that network for all outbound messages. In most cases, you can leave the default.
`RvService`	TIBCO Rendezvous service. The Rendezvous daemon divides the network into logical partitions. Each transport communicates on a single service. A transport can communicate only on the same service with other transports. Unless you are using a non-default TIBCO Rendezvous configuration, you should leave the default (7500).
`RvaHost`	Computer on which the TIBCO Rendezvous agent runs. This variable is only relevant if you are using the TIBCO Rendezvous Agent (`rva`) instead of the TIBCO Rendezvous daemon, and if you have configured a non-default setup. See *TIBCO Rendezvous Administration* for details about specifying the rva parameters.

Table 8 Predefined Global Variables

Variable	Description
`RvaPort`	TCP port where the TIBCO Rendezvous agent (rva) listens for client connection requests. See *TIBCO Rendezvous Administration* for details about specifying the rva parameters. Defaults to 7501.
`TIBHawkDaemon`	TIBCO Rendezvous daemon used in the TIBCO Hawk session. See the *TIBCO Hawk Installation and Configuration* manual for details about this parameter.
`TIBHawkNetwork`	TIBCO Rendezvous network used by the TIBCO Hawk session. See the *TIBCO Hawk Installation and Configuration* manual for details about this parameter.
`TIBHawkService`	TIBCO Rendezvous service used by the TIBCO Hawk session. See the *TIBCO Hawk Installation and Configuration* manual for details about this parameter.

Connection Settings

You can save your connection settings to the J. D. Edwards OneWorld Xe system, then reuse them when creating another adapter configuration. You save connection settings through a menu in the Connection Settings Toolbar.

The connection settings menu has the following choices:

- Save Connection Settings

  After filling the information in the Connection, use this choice to save the information in a file.

- Delete Connection Setting

  Use to delete a saved file.

- User Connections

  Choose a stored setting to fill in the values in the Connection tab.

For information on using global variables in the Connection tab, see Connection Settings on page 165.

Setting Encoding Options

TIBCO Adapter for J. D. Edwards OneWorld Xe only supports ISO-LATIN-1.

The Values column contains the value to use in the `System Encoding Other` or `File Encoding Other` field.

Table 9 Common Encoding Aliases

Values	Common Alias Name	Comments
LATIN_1	iso-8859-1 iso_8859-15 ibm-819 ibm819 cp819 latin1 latin-1 ascii ascii-7 us-ascii 8859-1 csisolatin1 iso-ir-100 iso_8859-1:1978 ansi_x3.4-1968 ansi_x3.4-1986 iso_646.irv:1991 iso646-us us cp367 csASCII ibm367 iso-8859-1:1987 l1 iso_8859_1 ansi_x3.110-1983 iso8859-1 (and others)	Western Europe

See the *TIBCO Adapter Concepts* book for an introduction to Internationalization topics such as Unicode and how adapters handle it.

Chapter 8 Monitoring the Adapter Using TIBCO Hawk

This chapter explains how to use TIBCO Hawk microagents to monitor and manage the adapter.

Topics

Overview

TIBCO Hawk is a sophisticated tool for enterprise-wide monitoring and managing of all distributed applications and systems. System administrators can use it to monitor adapters in a wide area network of any size. TIBCO Hawk can be configured to monitor system and adapter parameters and to take actions when predefined conditions occur. These actions include: sending alarms that are graphically displayed in the TIBCO Hawk display, sending email, paging, running executables, or modifying the behavior of a managed adapter.

Unlike other monitoring applications, TIBCO Hawk relies on a purely distributed intelligent agent architecture using publish or subscribe to distribute alerts. TIBCO Hawk uses TIBCO Rendezvous for all messaging and thus gains the benefits and scalability from the TIBCO Rendezvous features of publish/subscribe, subject name addressing, interest-based routing, and reliable multicast.

TIBCO Hawk is a purely event-based system that uses alerts. The agents are configured with rules that instruct them on everything from what and how to monitor to what actions to take when problems are discovered. Thus the workload is fully distributed throughout the enterprise. Every agent is autonomous in that it does not depend on other components to perform its functions.

The TIBCO Hawk Enterprise Monitor consists of these components:

- **Display**—GUI front end that displays alarms and provides editors to create rule bases, create tests, view messages, and invoke microagents to request information or initiate an action.
- **Agents**—Intelligent processes that perform monitoring and take actions as defined in rules.
- **Rulebases**—Rules that are loaded by agents to determine agent behavior.
- **Application Management Interface (AMI)**—Manages network applications via TIBCO Rendezvous and supports communication between a network application and monitoring TIBCO Hawk agents, including the ability to examine application variables, invoke methods, and monitor system performance.
- **Microagents**—Feed information back to TIBCO Hawk and expose action methods to rulebases.

  For more information, see the TIBCO Hawk documentation.

Starting TIBCO Hawk Software

The TIBCO Hawk agent can be configured to start automatically during the system boot cycle. See the *TIBCO Hawk Installation and Configuration* guide for information about starting TIBCO Hawk.

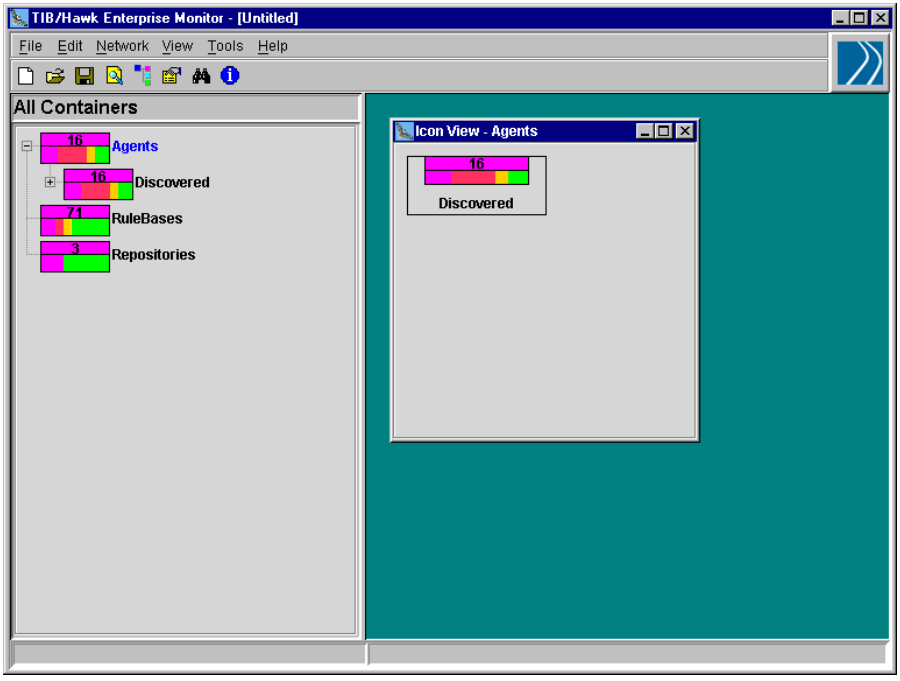
The *TIBCO Hawk Administrator's Guide* explains how to start the TIBCO Hawk Display.

The guides are included in your TIBCO Hawk software installation area.

The Auto-Discovery Process

After you start an instance of TIBCO Hawk Display, it continually discovers machines running TIBCO Hawk Agents on your network. Container icons are created for each agent, and arranged hierarchically in clusters. By default, agent icons are clustered according to subnets.

At first, the Agents container is empty. Its counter displays a value of zero and, on the right, the Discovered counter is also at zero. Both icons are initially green in color to show that no alerts, or warning messages, are in effect. As agents are discovered, the counters increment to reflect the current number of discovered agents:



Monitored network nodes are arranged in a hierarchical tree of containers. Clicking a container in the left panel displays nested items on the right.

Icon colors change to reflect the highest level of alert found on discovered agents. For explanations of icon elements and characteristics, see your *TIBCO Hawk Administrator's Guide*.

Invoking Microagent Methods

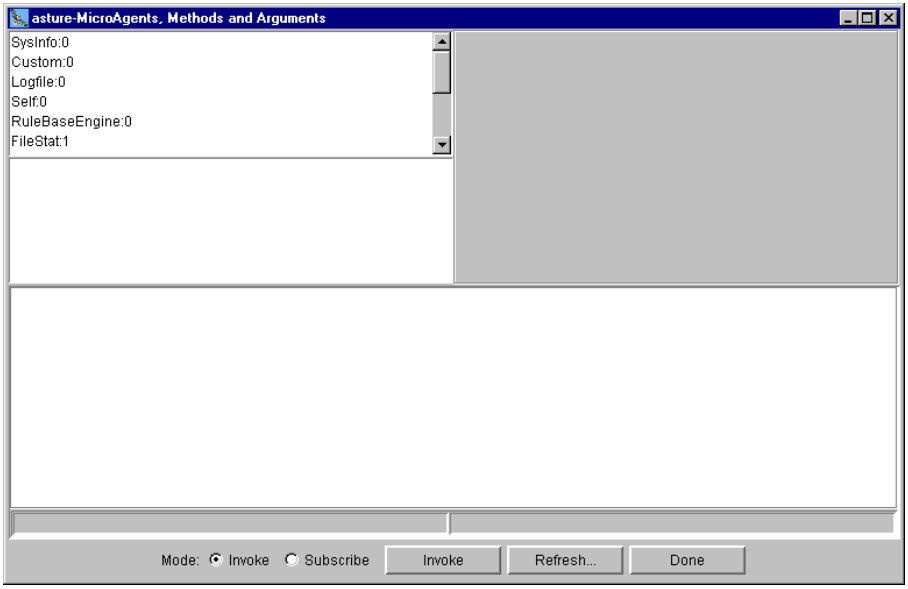
A set of default microagents is loaded when a TIBCO Hawk Agent is started. When you install and start the adapter, its microagents are dynamically added to the local agent.

To invoke a microagent method:

1. Start TIBCO Hawk Display, then right-click on the agent icon and select **Get Microagents**.

   If TIBCO Hawk security is implemented on your system and you do not have access to microagents on this agent, an error dialog displays. Select another agent, or contact your system administrator to obtain access.

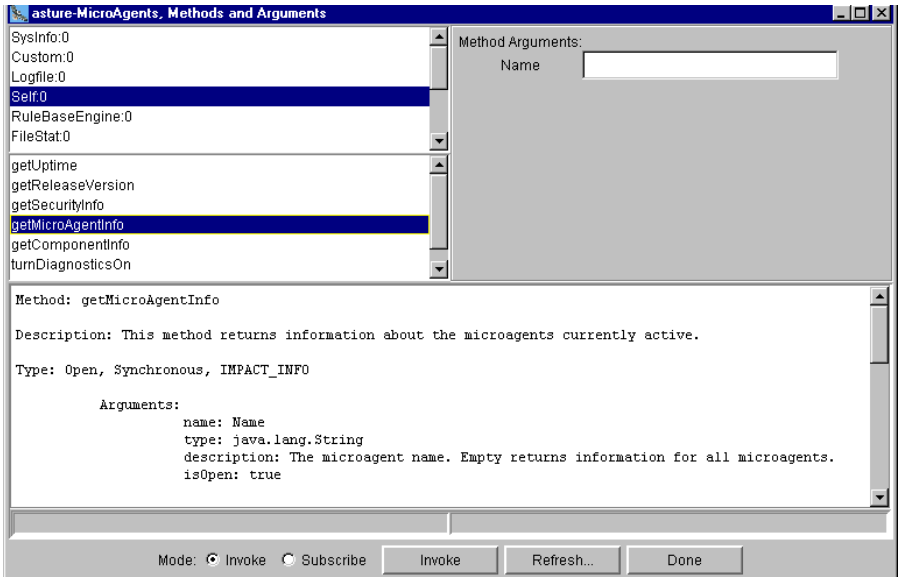
   The Microagents, Methods and Arguments dialog displays. The panel on the upper left lists microagents you can access on the current agent.



   This dialog has two modes, `Invoke` and `Subscribe`. Invoking a method immediately returns a single set of current results. Subscribing provides updates of current results at regular intervals. Radio buttons at the bottom of the dialog control these modes.

2. Click a microagent name, such as **Self**, to display a list of associated methods and text descriptions in the panels below.

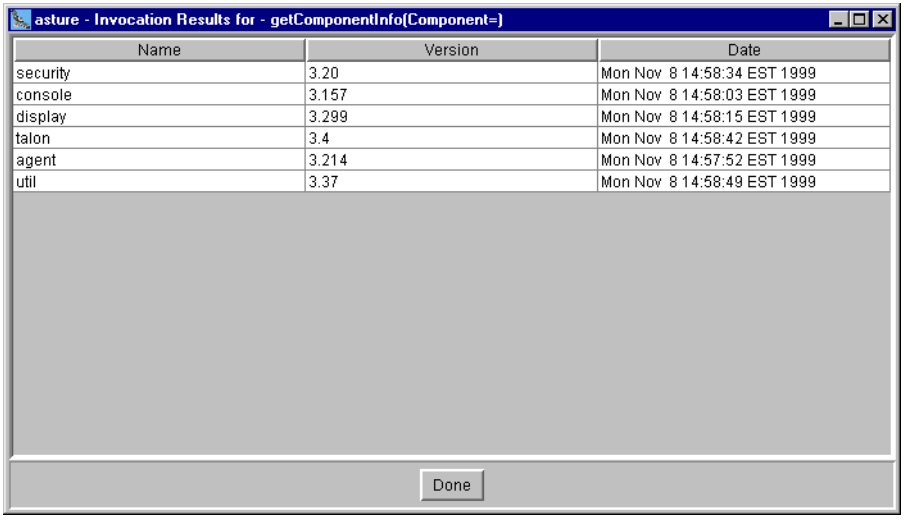
3. Click the name of the method to invoke, such as **getComponentInfo**.



If the method accepts arguments, fields for each argument display in the upper right panel. Detailed help text displays in the lower panel.

4. Specify any arguments for the method invocation.
5. Verify that the `Invoke` radio button is selected.
6. Click the **Invoke** button to invoke the selected method.

   The Invocation Results dialog displays the results returned by the method.



7. Click **Done** to close the dialog.

These steps describe how to interactively invoke a microagent method and receive a single set of results in TIBCO Hawk Display. You can also use a microagent method as the data source of a TIBCO Hawk rule. Rules automatically receive method results, apply tests to evaluate them, then take action if necessary. For more information on building TIBCO Hawk rules and rule bases, see your *TIBCO Hawk Administrator's Guide*.

Available Microagents

Each adapter has two microagents, a standard TIBCO Hawk microagent named `COM.TIBCO.ADAPTER.`*xyz* where *xyz* is the adapter configuration name and a 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. It does not make a permanent change of the setting in either the repository or the .tra file.

By default both the standard and custom microagents are available at runtime. You can disallow adding custom methods to the standard microagent when deploying the adapter by changing the `addCustomHawkMethodstoStdMAgent` property value in the adapter's property file.

The following table lists each method available for the adapter and page on which the method is explained.

Table 10 Microagent Methods

Method	Description	Page
`activateTraceRole()`	Activates a mapping of a role to a sink at runtime.	177
`deactivateTraceRole()`	Deactivates a mapping of a roles to sinks at runtime.	178
`getActivityStatistics()`	Returns the total number of objects processed for all the schemas.	179

Table 10 Microagent Methods (Cont'd)

Method (Cont'd)	Description (Cont'd)	Page
`getActivityStatisticsByOperation()`	Returns the total number of objects processed for all the schemas by each service that is associated with a specified operation.	180
`getActivityStatisticsByService()`	Returns information about the services implemented by this adapter.	181
`getAdapterServiceInformation()`	Returns information about the services implemented by this adapter.	182
`getComponents()`	Returns information about the publisher, subscriber and IODescriptor.	183
`getConfig()`	Returns basic configuration information. More specific information is accessed by the more specific methods.	184
`getConfigProperties()`	Returns all attributes and elements for the given repository object.	185
`getConnectionStatistics()`	Returns the state and statistics for all the current connections used by the adapter.	186
`getHostInformation()`	Returns standard and extended application information.	187
`getQueueStatistics()`	Returns the current count of elements in any internal queue used by the adapter.	188
`getRvConfig()`	Returns information about all Rendezvous sessions defined.	189
`getStatus()`	Returns general status information, such as the number of Rendezvous messages received and published, the number of errors since the last call, the PID of the application, and more.	190
`getThreadStatistics()`	Returns the operation counts of the current threads	191

Table 10 Microagent Methods (Cont'd)

Method (Cont'd)	Description (Cont'd)	Page
`getTraceSinks()`	Returns information about sinks to which traces currently go.	192
`getVersion()`	Returns the configuration ID, application name, version, and date for this adapter instance.	193
`_onUnsolictedMsg()`	Displays alert messages sent to the current adapter.	194
`preRegisterListener()`	Preregisters an anticipated listener.	195
`resetActivityStatistics()`	Resets all the counts for the activity statistics.	196
`resetConnectionStatistics()`	Resets all the counts for the connection statistics.	197
`resetThreadStatistics()`	Resets all the counts for the thread statistics.	198
`reviewLedger()`	Returns information retrieved from the ledger file of a certified messaging session for a publisher adapter.	199
`setTraceSinks()`	Adds a role or changes the file limit of a previously specified sink.	201
`stopApplicationInstance()`	Stops the running adapter instance.	202
`unRegisterListener()`	Unregisters a currently preregistered listener.	203

activateTraceRole()

Activates a mapping of a role to a sink at runtime. This replaces the now-deprecated `setTraceSink()` TIBCO Hawk method.

Input Parameters	Type	Description
Role Name	string	Name of the role to activate.
Sink Name	string	Name of the sink for which to activate the role.

deactivateTraceRole()

Deactivates a mapping of a roles to sinks at runtime.

Input Parameters	Type	Description
Role Name	string	Name of the role to activate.
Sink Name	string	Name of the sink for which to activate the role.

getActivityStatistics()

Returns the total number of objects processed for all the schemas, based on the request type. Also, returns the number of success and error objects.

Input Parameter	Type	Description
GetSubTotalBy	string	Indicates how to group the subtotals, by `Service` or `Operation`.

Returns	Type	Description
Name	string	Service name or `All Services` which represents the final tally of all the services
Total	integer	Total number of objects processed including both success and failures.
Success	integer	Total number of objects successfully processed.
Failure	integer	Total number of objects that caused an error during processing.
MeasurementInterval	integer	Displays the time (in seconds) since last time the adapter was reset, or if never reset, since the adapter started.

getActivityStatisticsByOperation()

Returns statistics about one operation.

Input Parameter	Type	Description
Operation	string	Name of the operation.

Returns	Type	Description
Operation	string	Name of the operation.
Service Name	string	Name of the service.
Total	integer	Total number of objects processed, both success and failures.
Success	integer	Total number of objects successfully processed.
Failure	integer	Total number of objects that caused an error during processing.
MeasurementInterval	integer	Displays the time (in seconds) since last time the adapter was reset, or if never reset, since the adapter started.
LineIndex	string	Concatenated string of `Service Name` and `Operation` separated by a comma.

getActivityStatisticsByService()

Returns statistics about the data handled by a given adapter service or all adapter services since the time the adapter was started.

Input parameter	Type	Description
Service Name	string	Name of service to get the statistics for. If no service name is given, performance statistics for all services is returned.

Returns	Type	Description
Service Name	string	Service name
Schema Name	string	Name of top level schema processed by this service.
Operation	string	Type of operation this service provides.
Total	integer	Total number of objects processed, both success and failures.
Success	integer	Total number of objects successfully processed.
Failure	integer	Total number of objects that caused an error during processing.
MeasurementInterval	integer	Displays the time (in seconds) since last time the adapter was reset, or if never reset, since the adapter started.
LineIndex	string	Concatenated string of `Service Name` and `Operation` separated by a comma.

getAdapterServiceInformation()

Returns information about the services implemented by this adapter.

Input Parameter	Type	Description
Service Name	string	Name of the service from which to get information. Default is `ALL`.

Returns	Type	Description
Line	integer	Sequential row number.
Service Name	string	Name of the service as defined at design-time.
Endpoint Name	string	Name of the endpoint used for this service.
Type	string	Type of the endpoint, for example, publisher or subscriber.
Quality of Service	string	Quality of service for the endpoint. For example RVCM or JMS Persistent.
Subject	string	Subject defined for this endpoint.
Class	string	Class associated with the endpoint.
Number of Messages	integer	Number of messages processed for this endpoint.

getComponents()

Returns information about the currently active TIBCO Hawk components such as publishers, subscribers, or timers.

Input Parameters	Type	Description
Component Name	string	Name of the component. If no value is enter, all components display.
Component Type	string	Any of `Publisher`, `Subscriber`, `Timer`, or `IODescriptor`. The default value is `All`.

Returns	Type	Description
Instance ID	string	Name of this adapter instance as defined at design-time.
Adapter Name	string	Name of the adapter.
Component Name	string	Name of the component.
Component Type	string	The name of the TIBCO Adapter SDK class for this component, such as `Publisher`, `Subscriber`, or `IODescriptorSource`. For more information about the class, see your TIBCO Adapter SDK documentation.
Session Name	string	Name of the session.
Description	string	Information about this component, for example, time interval, signal type, and validating the publisher or subscriber.

getConfig()

Retrieves generic configuration information. More specific configuration information is accessed through separate methods.

Returns	Type	Description
Instance ID	string	Configuration ID of this adapter.
Adapter Name	string	Name of the adapter.
Repository Connection	string	URL of the repository used for adapter instance.
Configuration URL	string	Location of the adapter project; either a file name or configuration URL.
Command	string	Command line arguments used to start the adapter.

getConfigProperties()

Returns all attributes and elements for the given repository object.

Input Parameter	Type	Description
Property	string	Name of the property for which elements (tags) and attributes are desired. For example, `agentone/startup`. If no value is given, all properties are returned.

Returns	Type	Description
Element Name	string	Repository directory for the property.
Attribute Name	string	Name of the repository object attribute.
Attribute Value	string	Value of the repository object attribute.
Line	integer	Line number in which this property is defined in the project file.

getConnectionStatistics()

Returns the state and statistics for all the current connections used by the adapter.

Returns	Type	Description
Connection ID	string	Unique identification of a particular connection.
Connection Type	string	Type or key that will match this connection to a thread or queue.
State	string	Current state: CONNECTED or DISCONNECTED.
NumRetries	integer	Total number of times this connection had to be reestablished.
TotalNumOperations	integer	Total number of operations processed by this connection since the adapter started.
CurrentNumOperations	integer	Total number of operations processed by this connection since the last reconnection.
NumLostConnections	integer	Total amount of time that this connection has been lost.
MeasurementInterval	integer	Displays the time (in seconds) since last time the adapter was reset, or if never reset, since the adapter started.

getHostInformation()

Return standard and extended application information set. It returns the following information.

Returns	Type	Description
Name	string	Name of the property.
Value	string	Value of the property.

getQueueStatistics()

Return the current count of elements in any internal queue used by the adapter. This includes the TIBCO Rendezvous event queues automatically spawned by Rendezvous for each adapter.

Returns	Type	Description
QueueID	string	Unique identification of a particular queue.
QueueType	string	Type or key that will match this queue to a thread or connection.
QueueCount	integer	Current number of elements in the queue.
MaxQueueSize	integer	Maximum number of elements in the queue.
MeasurementInterval	integer	Displays the time (in seconds) since last time the adapter was reset, or if never reset, since the adapter started.

getRvConfig()

Returns information about the Rendezvous session defined by this adapter. Information about all currently defined sessions is returned if no `sessionName` is provided.

Input Parameter	Type	Description
Session Name	string	Name of the Rendezvous session for which configuration is required. If not given, information about all sessions is returned. The default is all.

Returns	Type	Description
Instance ID	string	Configuration ID of this adapter.
Adapter Name	string	Name of the adapter.
Session Name	string	Name of the session.
Service	string	Service parameter for this session.
Daemon	string	Daemon parameter for this session.
Network	string	Network parameter for this session.
Synchronous?	boolean	Returns `1` if this is a synchronous session, `0` otherwise.
Session Type	string	Type of session; one of `M_RV`, `M_RVCM`, or `M_RVCMQ`.
Certified Name	string	Name of this certified session.
Ledger File	string	Ledger file for this certified messaging session. Returns the empty string for sessions that are not certified messaging sessions.
CM Timeout	string	Timeout for this certified messaging session. Returns the empty string for sessions that are not certified messaging sessions.

getStatus()

Retrieves basic status information about the adapter.

This information is fairly limited; for more detail, additional methods are provided (getConfig() on page 184 and getRvConfig() on page 189).

Returns	Type	Description
Instance ID	string	Configuration ID for this adapter instance.
Adapter Name	string	Name of the adapter.
Uptime	integer	Number of seconds since startup.
Messages Received	integer	Number of Rendezvous messages received.
Messages Sent	integer	Number of Rendezvous messages published.
New Errors	integer	Number of errors since the last call to this method.
Total Errors	integer	Total number of errors since startup.
Process ID	integer	Process ID of the application.
Host	string	Name of host machine on which this adapter is running.

getThreadStatistics()

Return the operation counts of the current threads.

Returns	Type	Description
ThreadID	string	Unique identification of a particular thread.
ThreadType	string	Type that tells what part of the adapter this thread belongs. Valid types include "Publisher", "Subscriber", "RPC", or "Connection".
TaskType	string	One-word description of the tasks this thread processes.
TaskCount	integer	Number of tasks processed by this thread.
MeasurementInterval	integer	Displays the time (in seconds) since last time the adapter was reset, or if never reset, since the adapter started.

getTraceSinks()

Returns information about sinks to which traces currently go.

Input Parameters	Type	Description
Sink Name	string	Name of the sink for which you need information. If no name is specified, information about all sinks is returned. Default is all.
Role Name	string	Name of the role for which you need information for the specified sink or sinks. Default is all.

Returns	Type	Description
Instance ID	string	Name of this adapter instance as a string.
Adapter Name	string	Name of the application for this sink.
Sink Name	string	Name of the sink
Sink Type	string	Type of this sink. One of fileSink, rvSink, hawkSink, stderrSink.
Roles	string	Roles this sink supports, as a string. For example warning, error, debug.

getVersion()

Retrieves version information for the current application. Two lines may be returned, one for the TIBCO Adapter SDK, one for the adapter.

Returns	Description
Instance ID	Configuration ID as a string, for example `SDK`.
Adapter Name	Name of the adapter as a string, for example `agentone`.
Version	Version number as a string, for example `5.2`.

_onUnsolictedMsg()

Displays all alert messages sent from the adapter or an error if not successful.

preRegisterListener()

Preregister an anticipated subscription service. Some sending applications can anticipate requests for certified delivery even before the listening applications start running. In such situations, the publication service can preregister subscription services, so TIBCO Rendezvous software begins storing outbound messages in the publication service ledger. If the listening correspondent requires old messages, it receives the backlogged messages when it requests certified deliver.

Input Parameters	Type	Description
Session Name	string	Name of the session that anticipates the listener.
Publisher Name	string	Name of the component for which the listener should be preregistered.
Listener Session Name	string	Name of the subscription service to preregister.

Returns OK if the subscription service was preregistered successfully, false otherwise.

resetActivityStatistics()

Resets all the counts for the activity statistics.

resetConnectionStatistics()

Resets all the counts for the connection statistics.

resetThreadStatistics()

Resets all the counts for the thread statistics.

reviewLedger()

Returns information retrieved from the ledger file of a Rendezvous certified messaging session.

Before invoking this method, ensure that the certified messaging publisher adapter has established a certified delivery agreement with its subscriber agents.

Input Parameters	Type	Description
Session Name	string	Name of the Rendezvous session for which ledger information is desired (default is all).
Subject	string	Name of the subject for which ledger information is desired.

Returns	Type	Description
Session Name	string	Name of the Rendezvous CM session to which this information applies.
Subject	string	Subject name for this session.
Last Sent Message	integer	Sequence number of the most recently sent message with this subject name.
Total Messages	string	Total number of pending messages with this subject name.
Total Size	integer	Total storage (in bytes) occupied by all pending messages with this subject name. If the ledger contains ten messages with this subject name, then this field sums the storage space over all of them.
Listener Session Name	string	Within each listener submessage, the Listener Session Name field contains the name of the delivery-tracking listener session.
Last Confirmed	string	Within each listener submessage, the Last Confirmed field contains the sequence number of the last message for which this listener session confirmed delivery.

Returns (Cont'd)	Type	Description
Line	integer	Row number in ledger file.
Unacknowledged Messages	integer	Number of RVCM messages pending for this listener. The value is computed by subtracting the last sent sequence number from the last acknowledged sequence number.

setTraceSinks()

Adds a role or changes the file limit of a previously specified sink.

Input Parameters	Type	Description
Sink Name	string	Name of the sink for which you want to add a role or change the file limit.
Role Name	string	Name of the role you want to add to this sink (warning, error, debug, or user defined). Default is all.
File Size	integer	Maximum file size for this sink. This parameter is ignored if the sink specified by sinkName is not a file sink.

Returns OK if successful or an error if not successful.

stopApplicationInstance()

Stops the specified adapter by calling the internal `stop()` method. This method returns OK if successful or an error if not successful.

unRegisterListener()

Unregister a currently preregistered subscription service.

Input Parameters	Type	Description
Session Name	string	Name of the session that anticipates the subscription service.
Publisher Name	string	Name of the publication service to which the subscription service is preregistered.
Listener Session Name	string	Name of the subscription service to unregister.

This method returns true if the subscription service was unregistered successfully, false otherwise.

Appendix A Trace Messages

This appendix explains the error messages that are logged to the location specified when the adapter was configured.

Topics

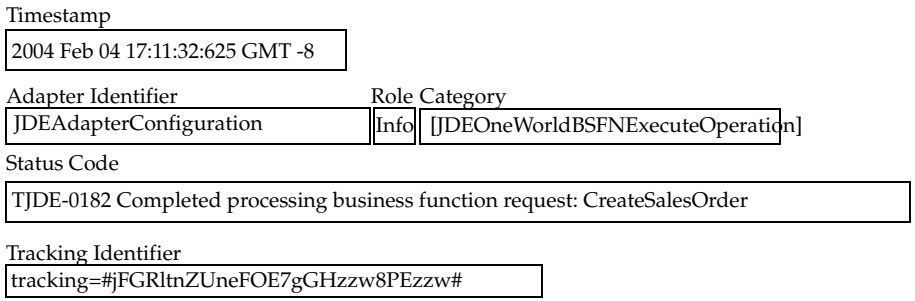
Overview

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

Each trace message can include the following fields:

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

The above fields are explained in Trace Message Fields on page 208. The following diagram shows an example trace message and calls out the fields.



Example Trace Messages

The following trace messages were written during a session where TIBCO Adapter for J. D. Edwards OneWorld Xe received a request for creating a sales order, then processed the object.

The first message indicates that TIBCO Adapter for J. D. Edwards OneWorld Xe has started. The timestamp indicates when the adapter started, and the role indicates that the trace message is informational, which means the activity is normal for the adapter. The category is identified, and the corresponding status code is displayed. The status code indicates that the adapter started successfully.

```
2004 Feb 04 17:10:28:328 GMT -8 JDEAdapterConfiguration Info
[Adapter] TIS-00001 "Adapter JDEAdapter started successfully"
```

The next set of trace messages indicates the adapter received a request for creating a sales order. The tracking identifier `#jFGRltnZUneFOE7gGHzzw8PEzzw#` included in the trace message uniquely identifies the message. The process (TIBCO BusinessWorks process) from which the message originated provided the identifier.

```
2004 Feb 04 17:11:17:968 GMT -8 JDEAdapterConfiguration Info
[Adapter] TIS-30028 "Obtained connection from pool"

2004 Feb 04 17:11:32:625 GMT -8 JDEAdapterConfiguration Info
[JDEOneWorldBSFNExecuteOperation] TJDE-0182 Completed processing
business function request: CreateSalesOrder
tracking=#jFGRltnZUneFOE7gGHzzw8PEzzw#
```

The final trace message indicates that the request for creating a sales order in J. D. Edwards OneWorld Xe was completed successfully.

Trace Message Fields

Each trace message includes the following fields:

Table 11 Tracing Fields

Field Name	Description
Timestamp	Timestamp of occurrence. For example, `2003 Feb 22 20:14:51:718 GMT -8`.
Adapter Identifier	Name of the adapter that wrote the trace message. This is a combination of the adapter acronym and adapter configuration name. For example, the application identifier, `ADB.publisher1` identifies a TIBCO Adapter for ActiveDatabase service named publisher1.
Role	A role can be: • Info. Indicates normal adapter operation. No action is necessary. A tracing message tagged with Info indicates that a significant processing step was reached and has been logged for tracking or auditing purposes. Only info messages preceding a tracking identifier are considered significant steps. • Warn. An abnormal condition was found. Processing will continue, but special attention from an administrator is recommended. • Error. An unrecoverable error occurred. Depending on the error severity, the adapter may continue with the next operation or may stop altogether. • Debug. A developer-defined tracing message. In normal operating conditions, debug messages should not display. When configuring the adapter you define what roles should or should not be logged. For example, you may decide not to log Info roles to increase performance.
Category	Name of the adapter java class that is printing this message.
Status Code	Unique code for the message and description. Status codes are identified by a unique number and description. If a trace message includes an error or warn role, the status code documentation includes a resolution. See Status Messages on page 210 for details.

Table 11 Tracing Fields

Field Name	Description
Tracking Identifier	A unique identifier that is "stamped" on each message by the originating adapter. The tracking identifier remains in effect from a message's beginning to its completion as it is exchanged by TIBCO applications. If the adapter is the termination point of the message, the tracking identifier is not displayed in the trace message. You cannot modify the tracking identifier format or configure what information is displayed.
Application Information	Application-specific information added to the tracking info to trace the message back to its source. Set initially by the originating adapter and carried forward. It is augmented by each intermediate component.

Status Messages

Status Code	Role	Category	Resolution
TJDE-0004	**Adapter startup complete. Ready to process requests . . .**		
	infoRole	JDEOneWorldAdapterCore	Indicates normal adapter information. No action necessary.
TJDE-0151	**Completed processing real-time event <real-time event name>**		
	infoRole	JDESender	Indicates normal adapter information. No action necessary.
TJDE-0181	**Received business function request: <business function request schema>**		
	infoRole	JDEOneWorldBSFNExecuteOperation	Indicates normal adapter information. No action necessary.
TJDE-0182	**Completed processing business function request: <business function request schema>**		
	infoRole	JDEOneWorldBSFNExecuteOperation	Indicates normal adapter information. No action necessary.
TJDE-0183	**Setting data structure parameter values for business function <business function name>.**		
	infoRole	JDEOneWorldBSFNExecuteOperation	Indicates normal adapter information. No action necessary.
TJDE-0184	**Executing business function <business function name>.**		
	infoRole	JDEOneWorldBSFNExecuteOperation	Indicates normal adapter information. No action necessary.
TJDE-0185	**Checking for errors for business function <business function name>.**		
	infoRole	JDEOneWorldBSFNExecuteOperation	Indicates normal adapter information. No action necessary.
TJDE-0186	**Retrieving data structure parameter values for business function <business function name>.**		
	infoRole	JDEOneWorldBSFNExecuteOperation	Indicates normal adapter information. No action necessary.

Status Code	Role	Category	Resolution
TJDE-0601	**Connection to JDE OneWorld successful. User: <user id>, Environment: <environment name>.**		
	infoRole	JDEOneWorldEventClient	Indicates normal adapter information. No action necessary.
TJDE-0602	**Adding subscription for event <real-time event name>.**		
	infoRole	JDEOneWorldEventClient	Indicates normal adapter information. No action necessary.
TJDE-0603	**Successfully subscribed to event <real-time event name>.**		
	infoRole	JDEOneWorldEventClient	Indicates normal adapter information. No action necessary.
TJDE-0381	**Received list create request for <table/view name>.**		
	infoRole	JDEOneWorldListCreateOperation	Indicates normal adapter information. No action necessary.
TJDE-0382	**Completed processing list create request for <table/view name>.**		
	infoRole	JDEOneWorldListCreateOperation	Indicates normal adapter information. No action necessary.
TJDE-0581	**Received list retrieve request for: Table=<table/view name>, Handle=<list handle>, From Value=<from record>, To Value=<to record>**		
	infoRole	JDEOneWorldListRetrieveOperation	Indicates normal adapter information. No action necessary.
TJDE-0582	**Completed processing list retrieve request for: Table=<table/view name>, Handle=<list handle>, From Value=<from record>, To Value=<to record>**		
	infoRole	JDEOneWorldListRetrieveOperation	Indicates normal adapter information. No action necessary.
TJDE-0481	**Received list delete request for handle <list handle>.**		
	infoRole	JDEOneWorldListDeleteOperation	Indicates normal adapter information. No action necessary.

Status Code	Role	Category	Resolution
TJDE-0482	**Completed processing list delete request for handle <list handle>.**		
	infoRole	JDEOneWorldListDeleteOperation	Indicates normal adapter information. No action necessary.
TJDE-0483	**Error Occurred while creating JDE COnnection <error message>.**		
	errorRole	JDEOneWorldListDeleteOperation	Please check J.D.Edwards debug log for more information about this error. Also check that the connection parameters are correct.
TJDE-0001	**Error occurred when retrieving JDE OneWorld Xe Connection parameters.**		
	errorRole	JDEOneWorldAdapterCore	Please make sure that connection parameters are defined in Repository.
TJDE-0002	**Error occurred when connecting to JDE OneWorld server. Reason: <error message>**		
	errorRole	JDEOneWorldAdapterCore	Please make sure that JDE OneWorld server and network services are running. Also, check that network connectivity can be established between adapter machine and JDE OneWorld server.
TJDE-0003	**Error occurred when creating class loader for InterOp JAR files.**		
	errorRole	JDEOneWorldAdapterCore	Please make sure that directory name for GenJava JAR files is correct.
TJDE-0102	**Error occurred when trying JDEOneWorldBSFNExecuteServer:onInvoke method.**		
	errorRole	JDEOneWorldBSFNExecuteServer	Please check the error log for additional information.
TJDE-0101	**Business function request contains no data.**		
	errorRole	JDEOneWorldBSFNExecuteOperation	Please specify a correct business function request schema. Then try the request again.

Status Code	Role	Category	Resolution
TJDE-0103	**Error occurred in the constructor of JDEOneWorldBSFNExecuteOperation.**		
	errorRole	JDEOneWorldBSFNExecuteOperation	Please check the error log for additional information.
TJDE-0104	**Adding error in the ErrorList. {<error message>}**		
	errorRole	JDEOneWorldBSFNExecuteOperation	Please check previously listed error messages and J.D.Edwards debug log for more information about the error.
TJDE-0105	**Adding error in the ErrorList from GroupCall. . {<error message>}**		
	errorRole	JDEOneWorldBSFNExecuteOperation	Please check previously listed error messages and J.D.Edwards debug log for more information about the error.
TJDE-0106	**Error occurred when calling business function {<business function name>}. Error: {<error message>}**		
	errorRole	JDEOneWorldBSFNExecuteOperation	Please check the errors returned by JDE OneWorld server. Correct the business function data accordingly and try the request again.
TJDE-0107	**Field type must be JDEDate for Date field {<field name>} in {<business function name>} function.**		
	errorRole	JDEOneWorldBSFNExecuteOperation	Please check the type definition for the field in request/reply schema.
TJDE-0108	**Field type must be JDEMathNumeric for short field {<field name} in {<business function>} function.**		
	errorRole	JDEOneWorldBSFNExecuteOperation	Please check the type definition for the field in request/reply schema.
TJDE-0109	**Field type must be JDEMathNumeric for int field {<field name>} in {<business function name>} function.**		
	errorRole	JDEOneWorldBSFNExecuteOperation	Please check the type definition for the field in request/reply schema.

Status Code	Role	Category	Resolution
TJDE-0110	**Field type must be JDEMathNumeric for long field {<field name>} in {<business function name>} function.**		
	errorRole	JDEOneWorldBSFNExecuteOperation	Please check the type definition for the field in request/reply schema.
TJDE-0111	**Field type must be JDEMathNumeric for float field {<field name>} in {<business function name>} function.**		
	errorRole	JDEOneWorldBSFNExecuteOperation	Please check the type definition for the field in request/reply schema.
TJDE-0112	**Field type must be JDEMathNumeric for double field {<field name>} in {<business function name>} function.**		
	errorRole	JDEOneWorldBSFNExecuteOperation	Please check the type definition for the field in request/reply schema.
TJDE-0113	**Field type must be char for a char field {<field name>} in {<business function name>} function.**		
	errorRole	JDEOneWorldBSFNExecuteOperation	Please check the type definition for the field in request/reply schema.
TJDE-0118	**Business function {<business function name>} in group definition {<business function group name>} not found!**		
	errorRole	JDEOneWorldBSFNExecuteOperation	Please define the business function using Services GUI and try the request again.
TJDE-0119	**Error in setting parameter {<field name>) value in business function {<business function name>}.**		
	errorRole	JDEOneWorldBSFNExecuteOperation	Please check the type definition for the field in business function request schema.
TJDE-0120	**Error in creating XML request for business function {<business function name>}.**		
	errorRole	JDEOneWorldBSFNExecuteOperation	Please check the error log for additional information.

Status Code	Role	Category	Resolution
TJDE-0121	**Error in creating XML request for {<business function name>} business function from list of business functions.**		
	errorRole	JDEOneWorldBSFNExecuteOperation	Please check the error log for additional information.
TJDE-0122	**Error in instantiating {<business function name>} business function.**		
	errorRole	JDEOneWorldBSFNExecuteOperation	Please make sure that the GenJava JAR file for business function exists and the business function class name is correct. Please check the error log for additional information.
TJDE-0123	**Error in instantiating {<business function name>} business function and {<business function data structure name>} data structure.**		
	errorRole	JDEOneWorldBSFNExecuteOperation	Please make sure that the GenJava JAR file for business function exists and the business function class name is correct. Please check the error log for additional information.
TJDE-0124	**Error in Business Function Execute operation implementation for request : <business function request schema>**		
	errorRole	JDEOneWorldBSFNExecuteOperation	Please check previously listed error messages for business functions to resolve the errors. Error message numbers for business function messages are TJDE-0101 through TJDE-0123.
TJDE-0125	**Error(s) occurred when executing business function!{<business function name>}**		
	errorRole	JDEOneWorldBSFNExecuteOperation	Please check the errors returned by JDE OneWorld server. Correct the business function data accordingly and try the request again.

Status Code	Role	Category	Resolution
TJDE-0126	**Error occured in getting reply from JDE OneWorld server.**		
	errorRole	JDEOneWorldBSFNExecuteOperation	Please check the errors returned by JDE OneWorld server. Correct the business function data accordingly and try the request again.
TJDE-0127	**Field type must be Date for JDEDate field {<field name>} in data structure {<business function data structure name>}.**		
	errorRole	JDEOneWorldBSFNExecuteOperation	Please check the type definition for the field in business function request schema.
TJDE-0128	**Field type must be short, int, long, float or double for a JDEMathNumeric field {<field name>} in data structure {<business function data structure name>}.**		
	errorRole	JDEOneWorldBSFNExecuteOperation	Please check the type definition for the field in business function request schema.
TJDE-0129	**Field type must be short, int, long, float or double for a Double field {<field name>} in data structure {<business function data structure name>}.**		
	errorRole	JDEOneWorldBSFNExecuteOperation	Please check the type definition for the field in business function request schema.
TJDE-0130	**Error while setting JDEdwards DS element {<field name>} in data structure {<business function data structure name>}.**		
	errorRole	JDEOneWorldBSFNExecuteOperation	Please check the type definition for the field in business function request schema.
TJDE-0150	**Error while parsing incoming data for Business Function Execute Subscriber. Error: <error message>**		
	errorRole	JDEOneWorldBSFNExecuteSubscriber	Please specify a correct business function request schema. Also, please check the debug log for additional information.

Status Code	Role	Category	Resolution
TJDE-0606	**JDE OneWorld event type <type of received event> not handled by the adapter.**		
	errorRole	JDEOneWorldEventThread	The event subscribed by the adapter is not a Real-Time Event.
TJDE-0607	**Error occurred when retrieving the Real Time Event data structure <real-time event data structure name>.**		
	errorRole	JDEOneWorldEventThread	Please make sure that Real-Time Event data structures are defined on JDE OneWorld server.
TJDE-0608	**Error occurred while processing real-time Event <real-time event name>. Error: <error message>**		
	errorRole	JDEOneWorldEventThread	Please check the debug log for additional information.
TJDE-0301	**Error occurred when creating list for request: Table=<table/view name>, Type=<table/view>, Data Selection=<data selection criteria>, Data Sequence=<data sequencing criteria>**		
	errorRole	JDEOneWorldListCreateOperation	Please check the debug log for additional information.
TJDE-0331	**Table Name and Table Type must be defined in the server request <query list request schema>.**		
	errorRole	JDEOneWorldListCreateOperation	Please specify a table name and its type in the Query List request schema and try the request again.
TJDE-0332	**Query {<table/view name>} is not defined for the adapter.**		
	errorRole	JDEOneWorldListCreateOperation	Please define the JDE OneWorld table or view using Services GUI and try the request again.
TJDE-0333	**Error in converting input parameters into XML Call List Request {<query list request schema>}.**		
	errorRole	JDEOneWorldListCreateOperation	Please correct the input parameters in the Query List request schema and try the request again.

Status Code	Role	Category	Resolution
TJDE-0334	**Error executing JDE OneWorld Request. {<error message>}**		
	errorRole	JDEOneWorldListCreateOperation	Please check the JDE OneWorld enterprise server logs for additional information.
TJDE-0335	**JDE OneWorld returned Error Code: {<error code>}. Error Description: {<error message>}.**		
	errorRole	JDEOneWorldListCreateOperation	Please correct the Query List request schema and try the request again.
TJDE-0336	**Error while parsing JDE OneWorld response XML {<;error message>}.**		
	errorRole	JDEOneWorldListCreateOperation	Please check the debug log for additional information.
TJDE-0355	**Error in JDEOneWorldListCreateServer : onInvoke method.**		
	errorRole	JDEOneWorldListCreateServer	Please check the debug log for additional information.
TJDE-0501	**Error occurred when processing list retrieve request for: Table=<table/view name>, Handle=<list handle>, From Value=<from record>, To Value=<record>**		
	errorRole	JDEOneWorldListRetrieveOperation	Please check error returned by the service to determine the problem.
TJDE-0502	**Error occurred when processing list retrieve request.**		
	errorRole	JDEOneWorldListRetrieveOperation	Please check the debug log for additional information. Then correct the Query List request schema and try the request again.
TJDE-0537	**XML List Handle required for retrieving list.**		
	errorRole	JDEOneWorldListRetrieveOperation	Please specify a correct list handle value in the Query List Retrieve schema and try the request again.

Status Code	Role	Category	Resolution
TJDE-0538	**Error executing JDE OneWorld Request. {<error message>}**		
	errorRole	JDEOneWorldListRetrieveOperation	Please check the JDE OneWorld enterprise server logs for additional information.
TJDE-0539	**JDE OneWorld returned Error Code: {<error code>}. Error Description: {<error message>}**		
	errorRole	JDEOneWorldListRetrieveOperation	Please correct the Query List request schema and try the request again.
TJDE-0540	**Error while parsing JDE OneWorld Response XML {<error message>}**		
	errorRole	JDEOneWorldListRetrieveOperation	Please check the debug log for additional information.
TJDE-0556	**Error in JDEOneWorldListRetrieveServer : onInvoke method.**		
	errorRole	JDEOneWorldListRetrieveServer	Please check the debug log for additional information.
TJDE-0441	**XML List Handle required for deleting list.**		
	errorRole	JDEOneWorldListDeleteOperation	Please specify a correct list handle value in the Query List Delete schema and try the request again.
TJDE-0442	**Error executing JDE OneWorld Request. {<error message>}**		
	errorRole	JDEOneWorldListDeleteOperation	Please check the JDE OneWorld enterprise server logs for additional information.
TJDE-0443	**JDE OneWorld returned Error Code: {<error code>}. Error Description: {<error message>}**		
	errorRole	JDEOneWorldListDeleteOperation	Please correct the Query List request schema and try the request again.
TJDE-0444	**Error while parsing JDE OneWorld Response XML {<error message>}**		
	errorRole	JDEOneWorldListDeleteOperation	Please check the debug log for additional information.

Status Code	Role	Category	Resolution
TJDE-0445	**Error executing JDEOneWorldListDeleteOperation : <error message>.**		
	errorRole	JDEOneWorldListDeleteOperation	Please check the JDE OneWorld enterprise server logs for additional information.
TJDE-0457	**Error in JDEOneWorldListDeleteServer : onInvoke method.**		
	errorRole	JDEOneWorldListDeleteServer	Please check the debug log for additional information.
TJDE-0690	**Error occurred while subscribing to real-time event. Error: <error message>**		
	errorRole	JDEOneWorldEventClient	Please check J.D.Edwards debug log for more information about this error
TJDE-0692	**Could not open real-time event port. Real-time event dispatching is disabled!**		
	errorRole	JDEDispatcherImpl	Other processes might be using TCP/IP port for real-time events. Please end other processes and try request again.

Appendix B Data Type Mapping

This table describes the mapping between J. D. Edwards OneWorld Xe data types, TIBCO ActiveEnterprise data types, and Java data types.

J. D. Edwards OneWorld Xe Data Type	Java Data Type	TIBCO ActiveEnterprise Data Type
Character	Char	char
JDEDate	java.util.Date	date
JDEMathNumeric	double	r8
String	java.lang.String	string
All other data types	java.lang.String	string

TIBCO Software Inc. End User License Agreement

READ THIS END USER LICENSE AGREEMENT CAREFULLY. BY DOWNLOADING OR INSTALLING THE SOFTWARE, YOU AGREE TO BE BOUND BY THIS AGREEMENT. IF YOU DO NOT AGREE TO THESE TERMS, DO NOT DOWNLOAD OR INSTALL THE SOFTWARE AND RETURN IT TO THE VENDOR FROM WHICH IT WAS PURCHASED.

Upon your acceptance as indicated above, the following shall govern your use of the Software except to the extent all or any portion of the Software (a) is subject to a separate written agreement, or (b) is provided by a third party under the terms set forth in an Addenda at the end of this Agreement, in which case the terms of such addenda shall control over inconsistent terms with regard to such portion(s).

License Grant. The Software is the property of TIBCO or its licensors and is protected by copyright and other laws. While TIBCO continues to own the Software, TIBCO hereby grants to Customer a limited, non-transferable, non-exclusive, license to use the Number of Units set forth in the Ordering Document, in machine-readable, object code form and solely for Customer's internal business use.

Restrictions. Customer agrees not to (a) make more copies than the Number of Units plus a reasonable number of backups; (b) provide access to the Software to anyone other than employees, contractors, or consultants under written contract with Customer agreeing to be bound by terms at least as protective of TIBCO as those in this End User License Agreement ("Authorized Users"); (c) sublicense, transfer, assign, distribute to any third party, pledge, lease, rent, or commercially share the Software or any of Customer's rights under this Agreement (for the purposes of the foregoing a change in control of Customer is deemed to be an assignment); (d) use the Software for purposes of providing a service bureau, including, without limitation, providing third-party hosting, or third-party application integration or application service provider-type services, or any similar services; (e) use the Software in connection with ultrahazardous activities, or any activity for which failure of the Software might result in death or serious bodily injury to Customer or a third party; or (f) directly or indirectly, in whole or in part, modify, translate, reverse engineer, decrypt, decompile, disassemble, make error corrections to, create derivative works based on, or otherwise attempt to discover the source code or underlying ideas or algorithms of the Software. Customer may engage in such conduct as is necessary to ensure the interoperability of the Software as required by law, provided that prior to commencing any decompilation or reverse engineering of any Software, Customer agrees to it shall observe strict obligations of confidentiality and provide TIBCO reasonable advance written notice and the opportunity to assist with and/or conduct such activity on Customer's behalf and at Customer's expense.

Beta and Evaluation Licenses. Notwithstanding the foregoing, if the Software is being provided for demonstration, beta testing, or evaluation purposes, then Customer agrees (a) to use the Software solely for such purposes, (b) that the Software will not be used or deployed in a production or development environment, and (c) that such use shall automatically terminate upon the earlier of thirty days from the date Customer receives the right to install the Software, or Customer's receipt of notice of termination from TIBCO.

Maintenance. Provided Customer has paid applicable support fees (not included with Software fees unless separately listed), TIBCO shall provide support for generally available Software ("Maintenance") on an annual basis commencing on the Purchase Date. Thereafter Maintenance will be automatically renewed for successive one (1) year terms at the then current rates. Customer shall designate at TIBCO's support website https://support.tibco.com/esupport/newuser.html, the number of authorized contacts as permitted by TIBCO based on the level of Maintenance purchased (contacts are changeable upon 48-hours prior written notice to TIBCO). Each contact may contact TIBCO for problem resolution during TIBCO's published support hours corresponding to the level of Maintenance purchased.

Upon notice from a contact of a Software problem which can be reproduced at a TIBCO Maintenance facility, TIBCO shall use reasonable efforts to correct or circumvent the problem according to its published maintenance objectives. TIBCO reserves the right to make changes only to the most currently available version. TIBCO will use reasonable efforts to support the previously released version of the Software for a maximum of six months.

TIBCO shall have no obligation to provide Maintenance for the Software if (i) used on any computer system running other than the operating system software for which the Software is approved (as set forth in the Software documentation) and licensed hereunder, or (ii) if Customer has modified or authorized a third party to modify the Software. TIBCO shall have no obligation to modify any version of the Software to run with any new versions of any operating system, or any other third party software or hardware. If Customer purchases Maintenance for any Software, Customer must purchase the same level of Maintenance for all copies of the Software for which it is licensed.

Maintenance Fees for lapsed Maintenance or the changing of the level of Maintenance shall be mutually agreed upon between Customer and TIBCO. Upgrades, patches, enhancements, bug fixes, new versions and/or new releases of the Software provided from time to time under Maintenance shall be used only as replacements to existing copies, and shall not be deemed to increase the Number of Units, and use thereof shall be governed by the terms of this Agreement, except for the first paragraph of the Limited Warranty and any right of return or refund.

Services. Customer may request additional services ("Services") either in an Ordering Document, or by a separate mutually executed work order, statement of work or other work-request document incorporating the term of this End User License Agreement (each, a "Work Order"). Unless otherwise expressly agreed to in a Work Order, all Services and any work product therefrom shall be (a) performed on a time and materials basis, plus meals, lodging, travel, and other expenses reasonably incurred in connection therewith, (b) deemed accepted upon delivery, and (c) exclusively owned by TIBCO (except for Confidential Information of Customer), including all right, title and intellectual property or other right or interest therein. Each Work Order is intended to constitute an independent and distinct agreement of the parties, notwithstanding that each shall be construed to incorporate all applicable provisions of this End User License Agreement. Fees for Services shall be due and payable in United States dollars net 30 from the date of TIBCO's invoice.

Limited Warranty. If Customer obtained the Software directly from TIBCO, then TIBCO warrants that for a period of thirty (30) days from the Purchase Date: (i) the media on which the Software is furnished will be free of defects in materials and workmanship under normal use; and (ii) the Software will substantially conform to its Documentation. This limited warranty extends only to the original Customer hereunder. Customer's sole and exclusive remedy and the entire liability of TIBCO and its licensors under this limited warranty will be, at TIBCO's option, repair, replacement, or refund of the

Software and applicable Maintenance fees, in which event this End User License Agreement shall terminate upon refund thereof.

This warranty does not apply to any Software which (a) is licensed for beta, evaluation, testing or demonstration purposes for which TIBCO does not receive a license fee, (b) has been altered or modified, except by TIBCO, (c) has not been installed, operated, repaired, or maintained in accordance with instructions supplied by TIBCO, (d) has been subjected to abnormal physical or electrical stress, misuse, negligence, or accident, or (e) is used in violation of any other term of this End User License Agreement. Customer agrees to pay TIBCO for any Maintenance or Services provided by TIBCO related to a breach of the foregoing on a time, materials, travel, lodging and other reasonable expenses basis. If Customer obtained the Software from a TIBCO reseller or distributor, the terms of any warranty shall be as provided by such reseller or distributor, and TIBCO provides Customer no warranty with respect to such Software.

EXCEPT AS SPECIFIED IN THIS LIMITED WARRANTY, THE SOFTWARE, MAINTENANCE AND SERVICES ARE PROVIDED "AS IS", ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS, AND WARRANTIES INCLUDING, WITHOUT LIMITATION, ANY IMPLIED WARRANTY OR CONDITION OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NONINFRINGEMENT, SATISFACTORY QUALITY OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE, ARE HEREBY EXCLUDED TO THE EXTENT ALLOWED BY APPLICABLE LAW. CERTAIN THIRD PARTY SOFTWARE MAY BE PROVIDED TO CUSTOMER ALONG WITH CERTAIN TIBCO SOFTWARE AS AN ACCOMMODATION TO CUSTOMER. THIS THIRD PARTY SOFTWARE IS PROVIDED "AS IS". CUSTOMER MAY CHOOSE NOT TO USE THIRD PARTY SOFTWARE PROVIDED AS AN ACCOMMODATION BY TIBCO. NO WARRANTY IS MADE REGARDING THE RESULTS OF ANY SOFTWARE, MAINTENANCE OR SERVICES OR THAT THE SOFTWARE WILL OPERATE WITHOUT ERRORS, PROBLEMS OR INTERRUPTIONS, OR THAT ERRORS OR BUGS IN THE SOFTWARE WILL BE CORRECTED, OR THAT THE SOFTWARE'S FUNCTIONALITY, MAINTENANCE OR SERVICES WILL MEET CUSTOMER'S REQUIREMENTS. NO TIBCO DEALER, DISTRIBUTOR, AGENT OR EMPLOYEE IS AUTHORIZED TO MAKE ANY MODIFICATIONS, EXTENSIONS OR ADDITIONS TO THIS WARRANTY.

Indemnity. If Customer obtained the Software from TIBCO directly, then TIBCO agrees at its own expense to defend or, at its option, to settle, any claim or action brought against Customer to the extent it is based on a claim that the unmodified Software infringes any patent issued by the United States, Canada, Australia, Japan, or any member of the European Union, or any copyright, or any trade secret of a third party; and TIBCO will indemnify and hold Customer harmless from and against any damages, costs and fees reasonably incurred (including reasonable attorneys' fees) that are attributable to such claim or action and which are assessed against Customer in a final judgment; provided that TIBCO is promptly notified in writing of such claim, TIBCO has the exclusive right to control such defense and/or settlement, and Customer shall provide reasonable assistance (at TIBCO's expense) in the defense thereof. In no event shall Customer settle any claim, action or proceeding without TIBCO's prior written approval. In the event of any such claim, litigation or threat thereof, TIBCO, at its sole option and expense, shall (a) procure for Customer the right to continue to use the Software or (b) replace or modify the Software with functionally equivalent software. If such settlement or modification is not commercially reasonable (in the reasonable opinion of TIBCO), TIBCO may cancel this End User License Agreement upon sixty days prior written notice to Customer, and refund to Customer the unamortized portion of the license fees paid to TIBCO by Customer based on a five-year straight-line depreciation. This Section states the entire liability of TIBCO with respect to the infringement of any intellectual property rights, and Customer hereby expressly waives any other liabilities or obligations of TIBCO with respect thereto. The foregoing indemnity shall not apply to the extent any infringement could have been avoided by use of the then-current release.

Limitation of Liability. EXCEPT AS PROVIDED UNDER INDEMNITY OR RESULTING FROM A BREACH OF CONFIDENTIALITY (THE "EXCLUDED MATTERS"), IN NO EVENT WILL EITHER PARTY OR TIBCO'S LICENSORS BE LIABLE FOR ANY LOST DATA, LOST REVENUE, LOST PROFITS, DAMAGE TO REPUTATION, BUSINESS INTERRUPTION, OR ANY OTHER INDIRECT, INCIDENTAL, CONSEQUENTIAL, SPECIAL, PUNITIVE, EXEMPLARY OR ANY SIMILAR TYPE DAMAGES ARISING OUT OF THIS AGREEMENT, THE USE OR THE INABILITY TO USE THE SOFTWARE, OR THE PROVISION OF ANY MAINTENANCE OR SERVICES, EVEN IF A PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. EXCEPT FOR THE EXCLUDED MATTERS, IN NO EVENT SHALL A PARTY BE LIABLE TO THE OTHER, WHETHER IN CONTRACT, TORT (INCLUDING ACTIVE OR PASSIVE NEGLIGENCE), BREACH OF WARRANTY, CLAIMS BY THIRD PARTIES OR OTHERWISE, EXCEED THE PRICE PAID BY CUSTOMER UNDER THE APPLICABLE ORDERING DOCUMENT.

THE FOREGOING LIMITATIONS SHALL APPLY EVEN IF THE ABOVE-STATED REMEDY OR LIMITED WARRANTY FAILS OF ITS ESSENTIAL PURPOSE. BECAUSE SOME STATES OR JURISDICTIONS DO NOT ALLOW LIMITATION OR EXCLUSION OF CONSEQUENTIAL OR INCIDENTAL DAMAGES, THE ABOVE LIMITATION MAY NOT APPLY TO CUSTOMER.

Confidentiality. "Confidential Information" means the terms of this End User License Agreement; all information marked by the disclosing party as proprietary or confidential; any provided software, related documentation or related performance test results derived by Customer; and any methods, concepts or processes utilized in provided software or related documentation. Confidential Information shall remain the sole property of the disclosing party and shall not be disclosed to any non-Authorized User of either TIBCO or Customer without the prior written consent of the disclosing party. If Confidential Information is communicated orally, such communication shall be confirmed as "Confidential" in writing within thirty days of such disclosure. The parties agree to protect the Confidential Information of the other in the same manner it protects the confidentiality of similar information and data of its own (and at all times exercising at least a reasonable degree of care). Except with respect to the Software, items will not be deemed Confidential Information if (i) available to the public other than by a breach of an agreement with TIBCO, (ii) rightfully received from a third party not in breach of any obligation of confidentiality, (iii) independently developed by one party without use of the Confidential Information of the other; (iv) known to the recipient at the time of disclosure (other than under a separate confidentiality obligation); or (v) produced in compliance with applicable law or court order, provided the other party is given reasonable notice of the same. Both parties agree to indemnify the other for any damages the other may sustain resulting from their unauthorized use and/or disclosure of the other's Confidential Information. Such damages shall include reasonable expenses incurred in seeking both legal and equitable remedies. To the extent required by law, at Customer's request, TIBCO shall provide Customer with the interface information needed to achieve interoperability between the Software and another independently created program, on payment of TIBCO's applicable fee. Customer agrees to observe obligations of confidentiality with respect to such information.

Export. Software, including technical data, is subject to U.S. export control laws, including the U.S. Export Administration Act and its associated regulations, and may be subject to export or import regulations in other countries. Customer agrees to comply strictly with all such regulations and agrees to obtain all necessary licenses to export, re-export, or import Software.

Government Use. If the Customer is an agency, department, or other entity of the United States Government ("Government"), the use, duplication, reproduction, release, modification, disclosure or transfer of the Software, or any related documentation of any kind, including technical data or manuals, is restricted in accordance with Federal Acquisition Regulation ("FAR") 12.212 for civilian agencies and Defense Federal Acquisition Regulation Supplement ("DFARS") 227.7202 for military agencies. The Software is commercial computer software and commercial computer software documentation. Use of the Software and related documentation by the Government is further restricted in accordance with the terms of this Agreement, and any modification thereto.

Orders. An Ordering Document shall be deemed accepted only by issuance of a TIBCO invoice and solely for purposes of administrative convenience. None of the terms of the Ordering Document (other than the Software product name, Number of Units, level of Maintenance, description of Services, and fees due in connection therewith) shall apply for any reason or purpose whatsoever, regardless of any statement on any Ordering Document to the contrary, unless countersigned by an officer of TIBCO. This Agreement constitutes the entire agreement between the parties with respect to the use of the Software, Maintenance and Services, and supersedes all proposals, oral or written, and all other representations, statements, negotiations and undertakings relating to the subject matter hereof. All orders of Software, Maintenance or Services by Customer to TIBCO shall be deemed to occur with or without reference to, under the terms of this End User License Agreement, unless expressly superseded by a signed written agreement between the parties. Neither the license to use the Software granted in this Agreement nor the obligation to pay the license fees set forth above are dependent upon the performance by any party of any Services or the supply of any other software program or product. Software shall be delivered electronically, and such delivery shall be deemed complete when the Software is made available for download by Customer.

Term and Termination. Maintenance or Services may be terminated: (a) by either party upon a default of the other, such default remaining uncured for fifteen days from written notice from the non-defaulting party; (b) upon the filing for bankruptcy or insolvency of the other party, (c) by either party upon prior written notice at least ninety (90) days prior to the end of any annual Maintenance term; or (d) by Customer (for Services), upon ten days prior written notice. Termination of Maintenance or Services shall not terminate this End User License Agreement. Customer may terminate this End User License Agreement in its entirety at any time by destroying all copies of the Software. Upon termination of this End User License Agreement in its entirety, for any reason, Customer must cease using and return or destroy all copies of the Software. Customer's obligation to pay accrued charges and any fees due as of the date of termination, as well as the sections entitled "Confidentiality", "Limited Warranty" and "Limitation of Liability" shall survive any such termination.

Authority. You hereby represent and warrant that you have full power and authority to accept the terms of this End User License Agreement on behalf of Customer, and that Customer agrees to be bound by this End User License Agreement.

General. Fees on the Ordering Document (all to be paid on the latter of thirty days from Invoice by TIBCO or the date set forth in the Ordering Document) do not include sales, use, withholding, value-added or similar taxes, and Customer agrees to pay all sales, use, value-added, goods and services, consumption, withholding, excise and any other similar taxes or government charges, exclusive of TIBCO's income tax. Customer agree to pay all reasonable costs incurred (including reasonable attorneys' fees) in collecting past due amounts. Except as set forth in the Section entitled "Limited Warranty" all fees paid under or in connection with this End User License Agreement are non-refundable and no right of set-off exists. All payments of fees due shall be made in U.S. dollars, net 30 from Purchase Date, or, for any other amounts coming due hereafter, net 30 from TIBCO's invoice. A service charge of one and one-half percent per month will be applied to all invoices that are not paid on time. No delay in the performance of any obligation by either party, excepting all obligations to make payment, shall constitute a breach of this End User License Agreement to the extent caused by force majeure. Customer hereby grants TIBCO and its independent auditors the right to audit Customer's compliance with this End User License Agreement. If any portion of this End User License Agreement is found to be void or unenforceable, the remaining provisions shall remain in full force and effect. This End User License Agreement shall be governed by and construed in accordance with the laws of the State of California, United States of America, as if performed wholly within the state and without giving effect to the principles of conflict of law. The United Nations Convention on Contracts for the International Sale of Goods is excluded from application hereto. If any portion hereof is found to be void or unenforceable, the remaining provisions of this Agreement shall remain in full force and effect.

Definitions. In connection with this End User License Agreement the following capitalized terms shall have the following meaning: "CICS Region" means a subdivided mainframe address space managed by CICS for resource allocation, resource sharing, and transaction execution, of which the resource definitions include the TIBCO EMS Client for z/OS; "Connection" means for TIBCO SmartSockets and TIBCO SmartMQ, any network protocol link established with such Software (directly or indirectly) to any other entity, including but not limited to software, firmware or hardware; "Connected Processor" means a Processor that produces information or messages consumed by the relevant Software (excluding Processors on devices such as routers, switches, proxies, HTTP or application servers configured to substantially pass-through information or messages to TIBCO Software); "Customer" means the original purchaser or licensee of the Software and any permitted successors and assigns; "Developer" means a Named User of a TIBCO Software product for use only in Non-Production; "Documentation" means text material that accompanies the TIBCO Software on delivery; "Enterprise" means an unlimited Number of Units of the TIBCO Software set forth in an Ordering Document, deployed by Customer for a period of **one (1) year** (unless otherwise set forth in an Ordering Document) from the Effective Date (the "Enterprise Term"), at which time, the Number of Units then deployed in Production and Non-Production use by Customer becomes fixed and Customer may not thereafter deploy additional Units. During the Enterprise Term, Customer's right to deploy an unlimited Number of Units does not extend to any entity which acquires, is acquired by, merged into, or otherwise combined with Customer. Customer hereby agrees to provide TIBCO, within sixty (60) days after the end of the Enterprise Term, with written notice of the Number of Units deployed at the end of the Enterprise Term by License Type, Platform and Unit; "License Type" means the environment in which the TIBCO Software may be used (including without limitation, Production, Non-Production); "Managed Endpoints" means the number of processors running instances of TIBCO BusinessWorks™ that are being managed by TIBCO ActiveMatrix™ Policy Manager; "MSU" means Millions of Service Units per hour, based on the then current MSU rating established by IBM for IBM and IBM compatible hardware which is used for software pricing (not

necessarily a direct indication of relative processor capacity) as set forth in IBM's generally available Large System Performance Reference; "Named User" means an identifiable individual, not necessarily named at the time of a license grant, designated by Customer to access the TIBCO Software, regardless of whether or not the individual is actively using the TIBCO Software at any given time; "Non-Production" means a non-operational environment into which the TIBCO Software may be installed, which is not processing live data, which is not running any operations of the Customer and which has not been deployed to permit any users to access live data. Non-Production environments include development, cold back-up, high availability, hot standby, and test environments; "Number of Units" means the cumulative number of copies of TIBCO Software licensed for use by type of Unit as set forth in this Agreement, or in an Ordering Document, and including, if applicable, as reported by Licensee upon expiration of a Project or Enterprise Term; "Ordering Document" means any purchase order or similar document or agreement requesting Software, Maintenance or Services; "Platform" means the operating system set forth in an Ordering Document; "Processor" means a central processing unit ("CPU") on which the TIBCO Software is licensed to run and which for the purposes of counting Processors on multicore chips, the number of Processors is the number of CPUs times the number of cores multiplied by .75.; "Processor Source Locked" means the number of connections to a single database made possible by (or licensed for) the relevant TIBCO Software regardless of how many Processors are used by the system or environment which is accessing the database; "Production" means an operational environment into which the licensed TIBCO Software has been installed, which is processing live data and which has been deployed so that the intended users of the environment are able to access the live data; "Project" means an unlimited Number of Units for the License Type listed in this Agreement, to be deployed by Customer solely in connection with the undertaking described under an Ordering Document for a period of **one (1) year** (unless otherwise set forth in this Agreement) from the Effective Date (the "Project Term"), at which time, the Number of Units then deployed in Production and Non-Production use by Customer becomes fixed and Customer may not thereafter deploy additional Units. During the Project Term, Customer's right to deploy an unlimited Number of Units does not extend to any entity which acquires, is acquired by, merged into, or otherwise combined with Customer. Customer hereby agrees to provide TIBCO, within sixty (60) days after the end of the Project Term, with written notice of the Number of Units deployed at the end of the Project Term by Unit and License Type. "Purchase Date" means the date of the Ordering Document is accepted by TIBCO; "Server Instance" means a computer with 1 CPUs (unless otherwise specified in the Agreement) performing common services for multiple other machines; "Software" means the most current generally available object code version (as of the Purchase Date) of the software products listed in an Ordering Document (except as provided for beta or evaluation licenses), in whole and in part, including its Documentation; "Third Party Software" means third-party software identified by its company and/or product name, the provision of which by TIBCO is made solely as an accommodation and in lieu of Customer purchasing a license to Third Party Software directly from the third party vendor; "Trading Partner" means an entity or individual with which the Licensee engages in electronic commerce by means of TIBCO Software in accordance with this Agreement; "Unit" means a license restriction describing the manner in which a copy (or multiple copies) of the TIBCO Software may be deployed (including, without limitation, Processor, Named User, Connected Processor, and Processor Source Locked) and is the mechanism used to determine the Number of Units licensed pursuant to an Ordering Document.

Special Product Provisions. TIBCO BusinessPartner: Customer may sublicense to third parties ("Partners") up to the total Number of Copies of TIBCO BusinessPartner, provided that for every such sublicense, the Number of Copies Customer is licensed to use shall be reduced by the same number, and provided further that prior to delivery of TIBCO BusinessPartner to a Partner, such Partner agrees in writing (a) to be bound by terms and conditions at least as protective of TIBCO as the terms of this Agreement, (b) that TIBCO BusinessPartner be used solely to communicate with Customer's implementation of TIBCO BusinessConnect, and (c) for such Partner to direct all technical support and Maintenance questions directly to Customer. Customer agrees to keep records of the Partners to which it distributes TIBCO BusinessPartner, and to provide TIBCO the names thereof (with an address and contact name) within sixty days of the end of each quarter. Embedded/Bundled Products. Some TIBCO Software embeds or bundles other TIBCO Software (e.g., TIBCO InConcert bundles TIBCO Rendezvous). Use of such embedded or bundled TIBCO Software is solely to enable the functionality of the TIBCO Software licensed on the Cover Page, and may not be used or accessed by any other TIBCO Software, or for any other purpose. Open Source Software: If Customer uses Open Source software in conjunction with the TIBCO Software, Customer must ensure that its use does not: (i) create, or purport to create, obligations of use with respect to the TIBCO Software; or (ii) grant, or purport to grant, to any third party any rights to or immunities under TIBCO's intellectual property or proprietary rights in the TIBCO Software. You also may not combine the TIBCO Software with programs licensed under the GNU General Public License ("GPL") in any manner that could cause, or could be interpreted or asserted to cause, the TIBCO Software or any modifications thereto to become subject to the terms of the GPL.

Version 5.5, April 2007

Index

A

B

C

G

H

I

L

M

P

R

S

T

U

V

X