TIBCO ActiveMatrix® Adapter for Files for Unix/Win (TIBCO Business Studio™) User's Guide

Software Release 1.3
February 2016

TIBC○®

Important Information

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

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

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

TIBCO, Two-Second Advantage, TIBCO ActiveMatrix Adapter for Files for Unix/Win, TIBCO ActiveMatrix Adapter for Files for Unix/Win (TIBCO Business Studio), TIBCO ActiveMatrix Adapter Framework, TIBCO ActiveMatrix BusinessWorks, TIBCO Adapter SDK, TIBCO Business Studio, TIBCO Designer, TIBCO Enterprise Message Service, TIBCO Hawk, TIBCO Rendezvous, and TIBCO Runtime Agent are either registered trademarks or trademarks of TIBCO Software Inc. in the United States and/or other countries.

Enterprise Java Beans (EJB), Java Platform Enterprise Edition (Java EE), Java 2 Platform Enterprise Edition (J2EE), and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle Corporation in the U.S. and other countries.

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

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

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

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

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

Contents

TIBCO Documentation and Support Services

Documentation for this and other TIBCO products is available on the TIBCO Documentation site. This site is updated more frequently than any documentation that might be included with the product. To ensure that you are accessing the latest available help topics, please visit:

https://docs.tibco.com

Product-Specific Documentation

Documentation for TIBCO products is not bundled with the software. Instead, it is available on the TIBCO Documentation site. To directly access documentation for this product, double-click the following file:

TIBCO_HOME/release_notes/TIB_adfilesbs_*version*_docinfo.html

where *TIBCO_HOME* is the top-level directory in which TIBCO products are installed. On Windows, the default *TIBCO_HOME* is C:\Program Files\tibco. On UNIX systems, the default *TIBCO_HOME* is /opt/tibco.

The following documents for this product can be found on the TIBCO Documentation site:

* *TIBCO ActiveMatrix Adapter for Files for Unix/Win (TIBCO Business Studio) Installation*

* *TIBCO ActiveMatrix Adapter for Files for Unix/Win (TIBCO Business Studio) User's Guide*

* *TIBCO ActiveMatrix Adapter for Files for Unix/Win (TIBCO Business Studio) Examples*

* *TIBCO ActiveMatrix Adapter for Files for Unix/Win (TIBCO Business Studio) Release Notes*

How to Contact TIBCO Support

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

* 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.

How to Join TIBCOmmunity

TIBCOmmunity is an online destination for TIBCO customers, partners, and resident experts. It is a place to share and access the collective experience of the TIBCO community. TIBCOmmunity offers forums, blogs, and access to a variety of resources. To register, go to the following web address:

https://www.tibcommunity.com

Product Overview

TIBCO ActiveMatrix® Adapter for Files for Unix/Win (TIBCO Business Studio™) provides a design-time environment to configure adapter configurations, Publication Service, Subscription Service, and schemas in TIBCO Business Studio™.

The following figure shows how TIBCO ActiveMatrix Adapter for Files for Unix/Win (TIBCO Business Studio) works with other products.
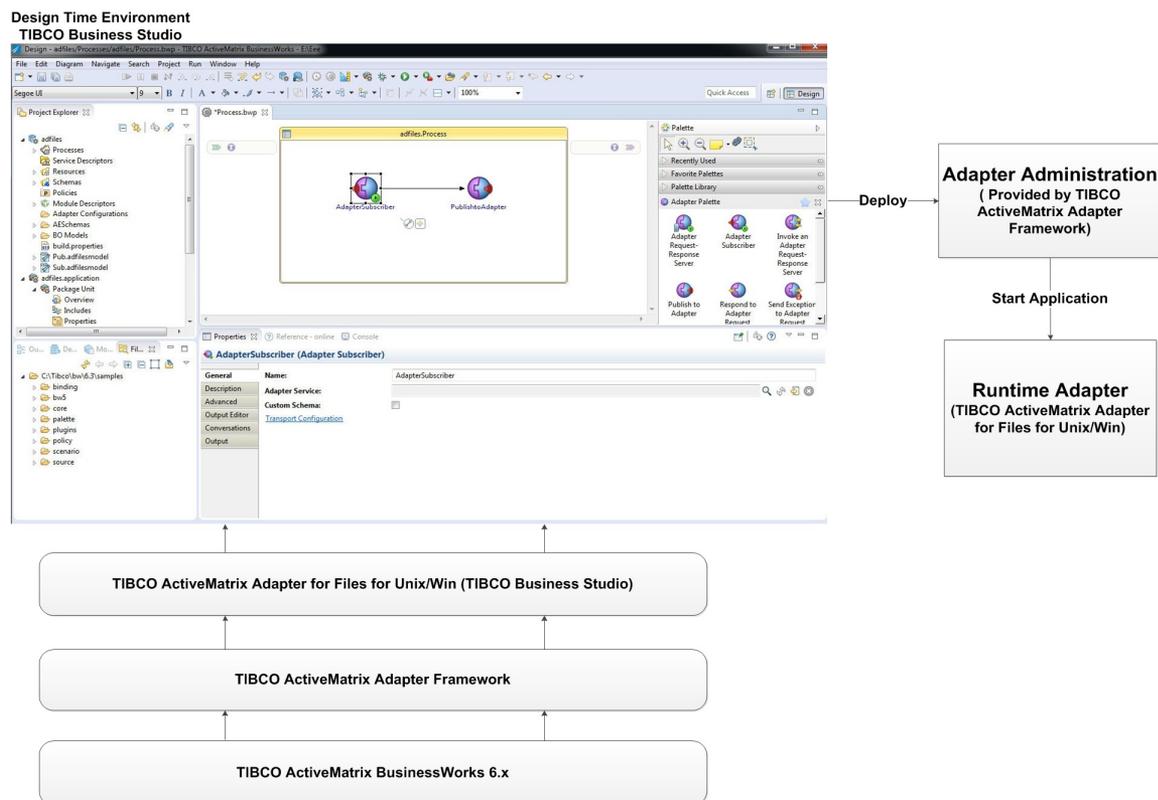
TIBCO ActiveMatrix BusinessWorks™

TIBCO ActiveMatrix BusinessWorks is an integration product suite for enterprise, web, and mobile applications. You can create services and integrate applications using a visual, model-driven development environment, and then deploy them in the TIBCO ActiveMatrix BusinessWorks run time.

TIBCO ActiveMatrix® Adapter Framework

TIBCO ActiveMatrix Adapter Framework works as a bridge between TIBCO ActiveMatrix Adapter for Files for Unix/Win (TIBCO Business Studio) and TIBCO ActiveMatrix BusinessWorks 6.x. TIBCO ActiveMatrix Adapter Framework provides the development environment for configuring supported TIBCO adapters in TIBCO Business Studio and working with TIBCO ActiveMatrix BusinessWorks 6 applications and processes. It also provides the following components:

- Adapter Administration

  Adapter Administration provides a centralized administrative command-line console to manage and monitor the adapter applications deployed in an enterprise.

- Adapter Palette

  Adapter Palette provides a set of activities to communicate with the configured TIBCO adapters.

TIBCO Business Studio

TIBCO Business Studio is the Eclipse graphical user interface (GUI) used by TIBCO ActiveMatrix BusinessWorks and TIBCO ActiveMatrix Adapter for Files for Unix/Win (TIBCO Business Studio) to design adapter configurations and business processes, run processes and generate deployable artifacts in the form of archive files.

TIBCO ActiveMatrix® Adapter for Files for Unix/Win

TIBCO ActiveMatrix Adapter for Files for Unix/Win is the runtime component of TIBCO ActiveMatrix Adapter for Files for Unix/Win (TIBCO Business Studio). Therefore TIBCO ActiveMatrix Adapter for Files for Unix/Win (TIBCO Business Studio) has all the runtime features of TIBCO ActiveMatrix Adapter for Files for Unix/Win.

As shown in the figure, you can configure the adapter through TIBCO ActiveMatrix Adapter Framework in TIBCO Business Studio. You can also use TIBCO ActiveMatrix BusinessWorks to develop a business process so that the adapter can be integrated into a data flow.

After installing TIBCO ActiveMatrix Adapter Framework, the Adapter palette is available for TIBCO adapters. The Adapter palette contains activities that communicate with configured TIBCO Adapter services. You can add activities and design business processes in TIBCO Business Studio. The configurations of all the features are similar to TIBCO ActiveMatrix Adapter for Files for Unix/Win. When you run adapter configurations, TIBCO ActiveMatrix Adapter for Files for Unix/Win (TIBCO Business Studio) uses the runtime properties in TIBCO ActiveMatrix Adapter for Files for Unix/Win.

Getting Started

This tutorial is designed for the beginners who want to use TIBCO ActiveMatrix Adapter for Files for Unix/Win (TIBCO Business Studio) in TIBCO Business Studio.

All the operations are performed in TIBCO Business Studio. See TIBCO Business Studio Overview to get familiar with TIBCO Business Studio.

A basic procedure of using TIBCO ActiveMatrix Adapter for Files for Unix/Win (TIBCO Business Studio) includes:

1. Creating a Project
2. Creating an Adapter Configuration
3. Configuring Adapter Preferences
4. Creating a Schema
5. Adding an Adapter Service
6. Testing an Adapter Configuration

Creating a Project

The first task using the adapter is creating a project. After creating a project, you can create adapter configurations and add services.

An Eclipse project is an application module configured for TIBCO ActiveMatrix BusinessWorks. An application module is the smallest unit of resources that is named, versioned, and packaged as part of an application.

Procedure

1. Start TIBCO Business Studio using one of the following ways:

    - Microsoft Windows: click **Start** > **All Programs** > **TIBCO** > *TIBCO_HOME* > **TIBCO Business Studio** *version_number* > **Studio for Designers**.

    - Unix: run the TIBCO Business Studio executable located in the *TIBCO_HOME*/studio/ *version_number*/eclipse directory.

2. On the Workspace Launcher page, accept the default workspace, or click **Browser** to create a new workspace, and then click **OK**.

3. From the main menu, click **File** > **New** > **BusinessWorks Resources**.

4. In the BusinessWorks Resource window, click the **BusinessWorks Application Module** resource, and then click **Next**.

    > TIBCO BusinessWorks provides several ways to launch this wizard. For detailed information, see the TIBCO ActiveMatrix BusinessWorks documentation.

5. In the Project window, enter a project name in the **Project name** field.

    To use the default location, keep the **Use default location**, **Create empty process**, and **Create Application** check boxes selected.

6. Click **Finish**.

Result

Two folders are created with the given project name. One is a project folder, and the other is an application folder that packages the application module, named as *application_module_name*.application.

The project folder contains the **Processes**, **Service Descriptors**, **Resources**, **Schemas**, **Module Descriptors** folders, which are created automatically in the project.

Creating an Adapter Configuration

After creating a project, you can create an adapter configuration. An adapter configuration contains all configuration information required by the runtime adapter.

The concept of adapter configuration in this book is the same as adapter instance in TIBCO ActiveMatrix Adapter for Files for Unix/Win.

Prerequisites

A project is created.

Procedure

1. Right-click the project, and then click **New** > **Other**.

2. In the "Select a wizard" dialog, select **TIBCO Adapters** > **Adapter for Files for Unix/Win** > **Adapter for Files Configuration**, and then click **Next**.

3. In the Create new Adapter for Files Configuration dialog, keep the default name or enter a new name for the adapter configuration in the **File name** field. Click **Finish**.

Result

An adapter configuration is created, and the **AESchemas**, **BO Models**, and **Adapter Configurations** folders are created automatically in the project.

Configuring Adapter Preferences

After creating an adapter configuration and before adding a service to the adapter configuration, It is good practice to configure adapter preferences to set up general information and the default transport type for the adapter.

The preference configurations apply to all the adapter services to be added in the same workspace. If you want to change settings for all adapter services, you can change settings in the Preference window, and then click the **Update Preferences** link.

Procedure

1. From the menu, click **Window** > **Preferences**.

2. In the Preference window, click **TIBCO Adapter** > **Adapter for Files for Unix/Win**.
   The Adapter for Files for Unix/Win panel is displayed on the right.

3. In the **General** tab, specify the working directory for Adapter Launcher, and specify the preferred settings for Publication Service and Subscription Service.
   When running Adapter Launcher, runtime files are generated in this directory.
   The following figure is an example of the settings.
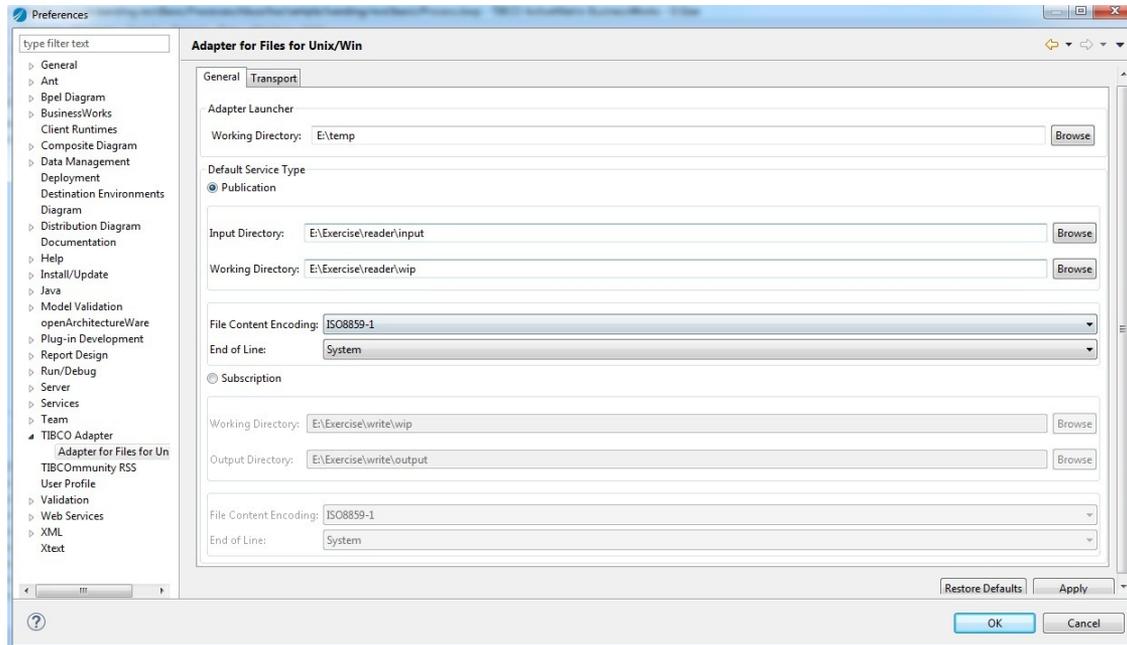
4. In the **Transport** tab, specify the default transport settings based on your requirements.

5. Click **Apply**, and then click **OK**.

Creating a Schema

After configuring preferences for the adapter, you can create a read schema or write schema. A schema is used when you add an adapter service to the adapter configuration.

Prerequisites

An adapter configuration is created in the project.

Procedure

1. Double-click the adapter configuration you have created. In the Adapter for Files Configuration panel, click the **Business Object Editor** link.

2. Create a read schema or write schema and configure it:

    *   If the adapter service you want to create is Publication Service, you can create a delimited or positional read schema. For details, see Defining Read Schemas.

    *   If the adapter service you want to create is Subscription Service, you can create a classic or manual write schema. For details, see Defining Write Schemas.

Adding an Adapter Service

After creating a schema, you can add an adapter service to the adapter configuration. Adapter services are abstractions that describe how adapters work together with other applications.

Prerequisites

An adapter configuration is created in the project.

Procedure

1. In the Adapter for Files Configuration panel, click the **Adapter Services** tab.

2. In the All Adapter Services panel, click **Add**.

3. In the Adapter Service General Configuration window, select the service type: Publication or Subscription. Click **Next**.
   The **Location**, **Configuration**, and **Service Name** fields are set by default automatically.

4. In the Schema window, click ⬚ to select a schema in the Select Schema Model window. Click **OK**.
   The schema you select is displayed in the **Schema** field.

5. Click **Next**.

6. In the Transport Session window, use the default transport.
   The default transport type is set in the Preferences window. If you do not want to use the default transport type, you can click ⬚ and create a new session with the wizard.

7. Click **Finish**.
   An adapter service is added in the All Adapter Services panel.

8. Configure the adapter service.
   For details on adapter services configuration, see Adapter Services.

Testing an Adapter Configuration

After configuring the adapter, you can use Adapter Launcher to start the adapter. Adapter Launcher is a tool that is used to test adapter configurations in TIBCO Business Studio.

Prerequisites

- An adapter service has been added as described in Adding an Adapter Service.

- The EMS server is running if the adapter uses the JMS transport type. For how to start the EMS Server, see the TIBCO Enterprise Message Server documentation.

When you set up an adapter configuration in Adapter Launcher, the configuration is saved for each adapter configuration in the project. If you want to run the adapter configuration in Adapter Launcher on a different machine, you have to change some settings because some directories are not valid.

Procedure

1. From the menu, click **Run** > **Run Configurations**.

2. In the left panel, click **Adapter Launcher** > **New**.

3. In the **Adapter Configuration** tab, click **Browse** to select the adapter configuration in the **Adapter Configurations** field.

4. In the Adapter Configuration selection dialog, expand the project and select the adapter configuration you have created.
   In the **Adapter Executable** field, the latest version of the adapter executable is selected by default.

5. Click **OK**.

6. In the **Working Directory** field, enter a directory path or click **Browse** to navigate to a working directory.
   When running Adapter Launcher, runtime files are generated in this directory, and all files in this directory are temporary. It is good practice not to edit files in the working directory. Ensure that the disk where the working directory is located contains enough space to save multiple copies of your project.

> With the working directory configuration, you can also click the **Update Preferences** link in the **Adapter Services** tab. And then specify the directory for the **Working Directory** field. By doing this, the specified working directory is available for all the projects in the same workspace.

7. Click **Apply**, and then click **Run**.
   TIBCO Business Studio starts the adapter. The progress of the adapter configuration is displayed in the console view.

8. Click the **Stop Adapter** icon on the tool bar of the console if you want to stop the adapter.

TIBCO Business Studio Overview

TIBCO Business Studio is an Eclipse-based integration development environment that is used to design, develop, and test ActiveMatrix BusinessWorks applications.

TIBCO Business Studio provides a workbench with which you can create, manage, and navigate resources in your workspace. A *workspace* is the central location on your machine where all data files are stored.
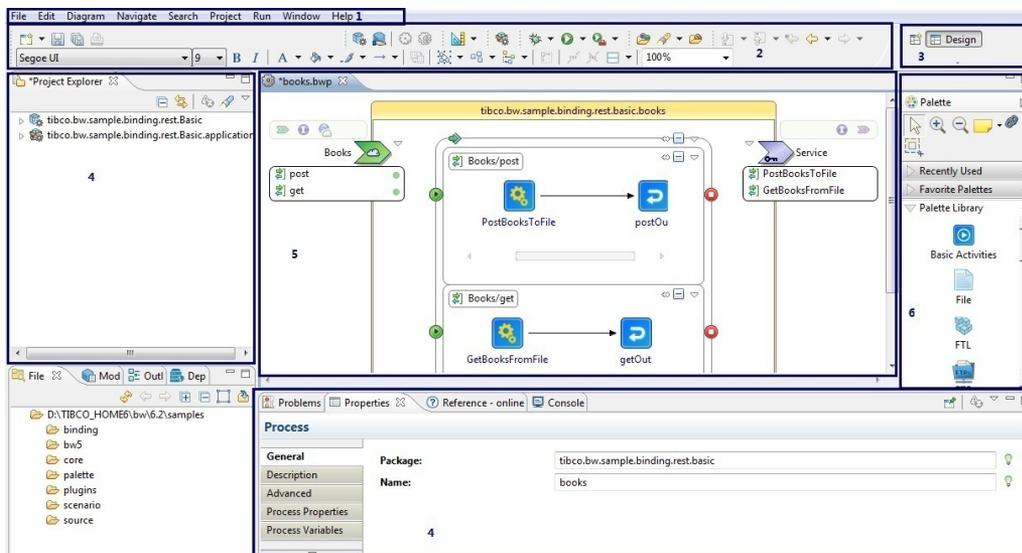
The workbench consists of:

- **Menu**: contains menu items such as File, Edit, Diagram, Navigate, Search, Project, Run, Window, and Help.

- **Toolbar**: contains buttons for frequently used commands such as New ⬚ ▾, Save 🖫, Enable/Disable Business Studio Capabilities ⬚ ▾, Create a new BusinessWorks Application Module ⬚, Create a new BusinessWorks Shared Module ⬚, Debug ⬚ ▾, Run ▶ ▾, and so on.

- **Perspective**: contains an initial set and layout of views that are required to perform a certain task. TIBCO Business Studio launches the Modeling perspective by default. You can change the perspective from the menu **Window** > **Open Perspective** > *Perspective_Name*.

- **View**: displays resources. For example, the Project Explorer view displays the ActiveMatrix BusinessWorks applications, modules, and other resources in your workspace, and the Properties view displays the properties for the selected resource. You can open a view from the menu **Window** > **Show View** > *View_Name*.

- **Editor**: provides a canvas to configure, edit, or browse a resource. Double-click a resource in a view to open the appropriate editor for the selected resource. For example, double-click an ActiveMatrix BusinessWorks process (`MortgageAppConsumer.bwp`) in the Project Explorer view to open the process in the editor.

- **Palette**: contains a set of widgets and a palette library. A *palette* groups activities that perform similar tasks and provide quick access to activities when configuring a process.

Adapter Configurations

An adapter configuration contains adapter services and all the resources required by the runtime adapter. The configured services can be wired to a business process through associated activities in the Adapter palette.

When you work with adapter configurations, you can go through the following workflow:

- Create an adapter configuration and its services

  The adapter provides a series of flexible methods of creating adapter configurations and services. For more information, see Creating an Adapter Configuration and Adding an Adapter Service.

- Configure an adapter configuration and its services

  Each adapter configuration must have its corresponding resource configured. The Adapter Configuration editor consists of several tabs. Each tab corresponds to one type of resource and has one or more configuration sections that are presented in panels. Use these tabs to configure the adapter resource.

  The Project Explorer shows a virtual outline of the adapter configuration. The outline depicts the virtual containment hierarchy of the configuration. Each node in the hierarchy is an Eclipse resource that is backed by a physical file. In most cases, a resource node in the outline maps to a tab in the Adapter Configuration editor. For more information, see Adapter Configurations and Adapter Services.

  You can set the preferred configuration settings in the Preferences dialog, and then configure each resource using the tabs in the editor. See the sections that correspond to each resource.

  > Many configuration options use module properties. On how to define and use module properties, see Module Properties.

- Validate an adapter configuration

  When you work with resources in TIBCO Business Studio, errors and warnings are logged automatically in the Problems view. Fix any error before you test the adapter configuration.

- Test an adapter configuration using the Adapter Launcher tool

  For more information, see Testing an Adapter Configuration.

Configuration

The **Configuration** tab contains the general information for an adapter configuration, and consists of the Adapter for Files Configuration panel and the Getting Started panel.

Adapter for Files Configuration Panel

The Adapter for Files Configuration panel contains the following fields:

Field	Description
Adapter Name	The name of the adapter.
Instance Id	The adapter configuration ID. The value of this field comes from the file name you specified when creating the adapter configuration. When you change the field value and save the configuration, the adapter configuration file name is also changed.
	For more information, see Guidelines for Choosing an Instance ID.

Field	Description
Description	(Optional) A short description for the adapter configuration.
Message Filter	(Optional) The filter that performs manipulations on incoming and outgoing data before sending it to the network or the target application.
	You can write filters by TIBCO Adapter SDK. For information about writing a message filter, see *TIBCO Adapter SDK Programmer's Guide*.

Getting Started Panel

The Getting Started panel contains the following links:

Field	Description
Configure Adapter Services	Click this link to add or configure adapter services directly.
	For details on configuring adapter services, see Adapter Services.
Business Object Editor	Click this link to add or define schemas using Business Object Editor directly.
	For details on how to define schemas using Business Object Editor, see Defining Schemas.

Guidelines for Choosing an Instance ID

Instance ID is the ID of an adapter configuration. The value of this field comes from the file name you specified when you created the adapter configuration.

The concept of adapter configuration in this book is the same as adapter instance in TIBCO ActiveMatrix Adapter for Files for Unix/Win.

The following list describes the guidelines for choosing an instance ID:

- An instance ID must use alphanumeric characters. You can use the underscore (_), but you cannot use the space character. The entire instance name must be less than 80 characters.

- An instance ID must be unique with respect to other adapter configurations in the same folder. You can use the same instance ID to name an adapter configuration in a different folder of the same project.

When you create an adapter configuration, TIBCO ActiveMatrix Adapter Framework automatically creates several resources for it. The names of these resources are derived from the ID of the configuration that they belong to. Changing the adapter instance ID results in an automatic regeneration of the resource names. If you manually modify any resource name, that particular name is not automatically regenerated the next time you rename the adapter configuration.

Adapter Services

The **Adapter Services** tab contains the All Adapter Services panel, the **Business Object Editor** link, the **Refresh Business Object** link, the **Update Preference** link and the service configuration panel on the right.

You can add Publication Service and Subscription Service in the All Adapter Services panel, and then configure the service in the panel displayed in the right. For detailed configuration, see Adapter Services.

Business Object Editor is used to define schemas for Publication Service and Subscription Service. When clicking the **Business Object Editor** link, the Adapter for Files Business Object Editor panel is

displayed. You can define schemas directly in the panel. For how to use Business Object Editor to define schemas, see Defining Schemas.

The **Refresh Business Object** link provides an easy way of applying configuration changes in schemas to corresponding services. If you change configuration for schemas, when you click the **Refresh Business Object** link, schema changes are applied to the corresponding services, and you do not have to manually wire the schema to the corresponding services again.

When you change the default settings in the Preference window, you can click the **Update Preferences** link, changes are applied automatically to all the adapter services you have created in the same workspace.

Transports

The primary task of an adapter is to retrieve or send data. When you add an adapter service to an adapter, the adapter user interface automatically creates the corresponding session and endpoint to encapsulate transport information necessary for data communication, depending on the transport protocol and delivery mode being used.

You cannot create sessions explicitly or add endpoints to the session. However, you can create a service, and create the session and endpoint in the adapter user interface. If you cannot change the setting directly, you can change the corresponding session or endpoint in the **Transport** tab. The information is grouped in panels.

Endpoints and Sessions

Both endpoints and sessions are concepts in TIBCO Adapter SDK, the fundamental class library is used in the adapter implementation.

Endpoints send or receive the data. They represent the service provided by an adapter. Each endpoint is associated with a session that is used to communicate with the source or target application. A session encapsulates the transport information of an adapter service.

For more information about these concepts, see *TIBCO Adapter SDK Programmer's Guide*.

Supported Endpoints

Publishers, Subscribers, Clients, or Servers are the endpoints that are available in an adapter. An adapter service encapsulates both an endpoint and the corresponding session. TIBCO ActiveMatrix Adapter for Files for Unix/Win (TIBCO Business Studio) supports the following services:

- Publication Service

  A Publisher endpoint and associated session.

- Subscription Service

  A Subscriber endpoint and associated session.

Supported Sessions

Both TIBCO Rendezvous and JMS transport types are supported.

- For TIBCO Rendezvous transports, sessions of the following kinds of quality of service are available:

  - Reliable (RV)

    Reliable Message Delivery ensures that each multicast or broadcast message is received if the physical network and packet recipients are working. It also ensures that the loss of a message is detected.

    Reliable Message Delivery can compensate for brief network failures, because it can retransmit a message on request if the first attempt fails. This option is appropriate when message delivery is

expected but some loss is tolerated. When you select this quality of service, an RV session is used.

- Certified (RVCM)

  Certified Message Delivery guarantees that every certified message reaches its intended recipient in the order sent. A message can be sent across network boundaries, and if a network fails, delivery attempts continue until delivery succeeds or until the time limit of the message expires. This is often called guaranteed delivery. When you select this quality of service, an RVCM session is used.

- Distributed Queue (Subscription Service only) (RVCMQ)

  Distributed Queue delivers a message to one of many service listeners (workers). It contains features of both Certified Messaging and Fault Tolerance.

For more information about the available TIBCO Rendezvous types of quality of service, see *TIBCO Rendezvous Concepts*.

- For JMS transports, sessions of the following standard connection factory types are available:

  - Topic

    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 called publish-subscribe.

  - Queue

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

For more information about connection factories, see *TIBCO Enterprise Message Service User's Guide*.

Sessions

The adapter provides Rendezvous sessions and JMS sessions. After adding a session in the All Adapter Transport panel and its endpoints, configuration options for the session are displayed in the Configuration panel. The transport type of the session you select determines specific session options.

No matter which transport type of session you select, JMS or Rendezvous, the general information of the session is displayed in a generic configuration panel at the top of the Configuration panel. Apart from the generic configuration, most sessions have more configuration options. These options are displayed in one or more configuration panels.

Rendezvous sessions include the following options:

- Rendezvous Configuration
- Rendezvous Options: Reliable Session
- Rendezvous Options: Certified Session
- Rendezvous Options: Distributed Queue Session

For more information, see the TIBCO Rendezvous documentation.

JMS sessions include the following options:

- JMS Configuration
- More Options Reference

For more information, see the TIBCO Enterprise Message Service documentation.

Rendezvous Configuration

The Rendezvous Configuration panel contains the following fields:

Field	Description
Name	The name of the TIBCO Rendezvous transport.
Description	(Optional) A short description for the transport.
Daemon	TIBCO Rendezvous daemon for this session.
Network	The network for this transport. By default, the property is an empty string, which is interpreted as the primary network. Using this attribute only makes sense on computers with more than one network interface.
Service	The service for this transport. By default, the property is defined to be the default TIBCO Rendezvous service (`7500`).
Connection Type	The following list describes the connection types: • Reliable • Certified • Distributed Queue

Rendezvous Options: Reliable Session

No additional configuration is required for the Reliable sessions.

Rendezvous Options: Certified Session

The Certified sessions contain the following fields:

Field	Description
CM name	The name used to identify the delivery tracking session. The CM name must be unique across the entire network.
Ledger File	If the value of the property value is a valid file name, the transport uses a file-based ledger.
Sync Ledger File	This check box controls the behavior when updating the ledger file. • When the check box is selected, operations that update the ledger file do not return until the changes are written to the storage medium. • When the check box is cleared, the operating system writes changes to the storage medium asynchronously.
Relay Agent	The relay agent for this transport.

Field	Description
Require Old Message	This check box indicates whether a persistent correspondent requires delivery of messages sent to a previous transport with the same name for which delivery is not confirmed. Its value affects the behavior of other delivery-tracking senders. • When the check box is selected, if the name attribute is non-NULL, the transport requires certified senders to retain unacknowledged messages sent to this persistent correspondent. • When the check box is cleared, messages are not retained.
Message Timeout (ms)	The maximum time (in milliseconds) that this call can block while waiting for a reply.

Rendezvous Options: Distributed Queue Session

The Distributed Queue sessions contain the following fields:

Field	Description
CMQ name	The sequence of module properties that specify the name of the queue.
Worker Weight	Relative worker weights assist the scheduler in assigning tasks. When the scheduler receives a task, it assigns the task to the available listener with the greatest listener weight. The default value is 1.
Worker Tasks	Worker tasks for this session. The default value is 1.
Worker Complete Time (ms)	If the complete time is non-zero, the scheduler waits a worker member to complete an assigned task. If the complete time elapses before the scheduler receives completion from the worker member, the scheduler reassigns the task to another worker member. The default value is 0.
Scheduler Weight	The ability of this session to fulfill the role of scheduler, relative to other members in the same queue. The queue members use relative scheduler weight values to elect one member as the scheduler. Members with higher scheduler weight take precedence. The acceptable values range from 0 to 65545. The default value is 1.
Scheduler Heartbeat (ms)	The scheduler session sends heartbeat messages at this interval (in milliseconds). All member sessions in the queue must specify the same value for this parameter. The acceptable values are unsigned decimals. The default value is 1000.
Scheduler Activation (ms)	When the heartbeat signal from the scheduler has been silent for this interval (in milliseconds), the queue member with the greatest scheduler weight takes its place as the new scheduler. All member sessions in the queue must specify the same value for this parameter. The acceptable values are unsigned decimals. The default value is 3000.

JMS Configuration

The JMS Configuration panel contains the following fields:

Field	Description
Name	The name of the JMS transport.
Description	(Optional) A short description for the transport.
Connection Type	The following list describes the available connection types: • **Direct** The connection is direct. • **JNDI** A JNDI Server is used.
Provider URL	(Direct connection type only) The URL of the server.
Connection Factory	(Direct connection type only) The following list describes the available connection factory types : • **TopicConnectionFactory** 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 called publish-subscribe. • **QueueConnectionFactory** A message sent to a queue is received by one and only one receiver. Each message has only one receiver, though multiple receivers might connect to the queue. The first receiver to access the queue receives the message. The other receivers do not. This messaging model is called point-to-point.
Test Connection	(Direct connection type only) Click this button to test the connection to the server at the specified Provider URL.
JNDI Reference	(JNDI connection type only) Displays JNDI server information.

More Options Reference

The More Options panel contains the following fields:

Field	Description
Client ID	The ID of the client.
User Identity	The detailed information of the user.

You have to configure the `JMSConnection` parameters in JMSSharedResources (in the **Resources** folder) when the TIBCO ActiveMatrix BusinessWorks activities use JMS transport in TIBCO Business Studio. The `JMSConnection` parameters are configured in the **Advanced** tab in TIBCO Business Studio.

Endpoints

The adapter provides the publisher endpoint and the subscriber endpoint. A publisher sends data to TIBCO Rendezvous or TIBCO Enterprise Message Service. A subscriber specifies the data consumers in the applications.

Publisher endpoints include TIBCO Rendezvous publisher endpoint and JMS publisher endpoint. Only reliable sessions and certified sessions can be associated with a publisher. Subscriber endpoints include TIBCO Rendezvous subscriber endpoint and JMS subscriber endpoint. The session type of the endpoint you select determines endpoint specific options you have to configure.

TIBCO Rendezvous Publisher Endpoint

You can add a publisher endpoint for a TIBCO Rendezvous reliable session or certified session. The EndPoint Configuration panel of the TIBCO Rendezvous publisher endpoint contains the following fields:

Field	Description
Name	The name of the publisher.
Description	(Optional) A short description for the publisher.
Endpoint Type	The type for the current publisher. The default type for TIBCO Rendezvous Reliable Publisher is `Rv Publisher`, the default type for TIBCO Rendezvous Certified Publisher is `RvCm Publisher`.
Wire Format	The format in which messages are sent. The available options are **ActiveEnterprise Message** (by default) and **XML Message**.
Subject	The subject with which the publisher sends out messages.
Reply Subject	The reply subject for the publisher.
Message Timeout (ms)	(Publishers for TIBCO Rendezvous Certified Sessions only) The time after which the message is discarded from the ledger file. The default value is `0`, which means the timeout is infinite.
Pre-registered Listeners	(Publishers for TIBCO Rendezvous Certified Sessions only) The comma-separated list of listeners preregistered for this publisher. Refer to each listener using the `CmName` of the session.

JMS Publisher Endpoint

You can add a publisher endpoint for a JMS Topic or Queue session. The EndPoint Configuration panel of the JMS publisher endpoint contains the following fields:

Field	Description
Name	The name of the publisher.
Description	(Optional) A short description for the publisher.

Field	Description
Endpoint Type	The type for the current publisher. The only available type is `Jms Publisher`.
Delivery Mode	The delivery mode for the messages. The available options are **Persistent** (by default) and **Non-Persistent**.
Destination	The destination which the publisher sends out messages to.
Reply destination	The reply destination for the publisher.
Message Priority	The priority of the messages to be sent. The value ranges from `0` to `9`, and the default value is `4`.
isCompressed	Select this check box if you want to compress the body of a message before sending the message to the server. Setting compression ensures that messages take less memory space in storage.
Message Timeout (ms)	The time after which the message is discarded from the ledger file. The default value is `0`, which means the timeout is infinite.

TIBCO Rendezvous Subscriber Endpoint

You can add a subscriber endpoint for a TIBCO Rendezvous Reliable, Certified, or Distributed Queue session. The EndPoint Configuration panel of the TIBCO Rendezvous subscriber endpoint contains the following fields:

Field	Description
Name	The name of the subscriber.
Description	(Optional) A short description for the subscriber.
Endpoint Type	The type for the current subscriber. The default type for TIBCO Rendezvous Reliable Subscriber is `Rv Subscriber`, the default type for TIBCO Rendezvous Certified Subscriber is `RvCm Subscriber`, the default type for TIBCO Rendezvous Distributed Queue Subscriber is `RvCmq Subscriber`.
Startup State	States when starting up the endpoint. The available options are **Active** (by default) and **Inactive**.
Wire Format	The format in which messages are sent. The available options are **ActiveEnterprise Message** (by default) and **XML Message**.
Subject	The subject with which the subscriber receives messages.
Listen Timeout (ms)	If no message is received after the specified amount of time, the adapter performs any actions specified in the program for that case. The default value is `0`.

JMS Subscriber Endpoint

You can add a subscriber endpoint for a JMS Topic or Queue session. The EndPoint Configuration panel of the JMS subscriber endpoint contains the following fields:

Field	Description
Name	The name of the subscriber.
Description	(Optional) A short description for the subscriber.
Endpoint Type	The type for the current subscriber. The only available type is `Jms Subscriber`.
Auto Confirm	When you select this check box, TIBCO Adapter SDK confirms events for the subscriber automatically.
Destination	The destination with which the subscriber receives messages.
Message Selector	A message selector is a string. With a string, a client program can specify a set of messages, based on the values of message headers and properties. After substituting header and property values from the message into the selector string, if the string is evaluated to `true`, the selector matches the message. Consumers can request that the server only deliver messages that match a selector.
Delivery Mode	(Subscribers for JMS Topic Sessions only) The delivery mode for the messages. The available options are **Durable** (by default) and **Non-Durable**.
Durable Name	(Subscribers for JMS Topic Sessions only) The name of the durable subscriber.

Logging

The adapter uses logging to configure logs. The adapter defines traces with different roles and sends them to log sinks with the corresponding role. You can either use the default standard I/O for logging or adjust where and when different types of information are sent by defining sinks and mapping each sink to one or more roles.

You can use the **Logging** tab to configure the logging options. The information is grouped in panels.

- If you use the console for logging, select the **Log To Standard I/O** check box in the Logging panel. You can send the information to multiple locations, and choose to log one or more message types. For more information, see Logging Configuration with Standard I/O.

- If you are using custom roles, use the All Log Sinks panel to add or remove log sinks and roles, and then configure the logging options for the selected sinks in the Configuration panel. For more information, see Log Sinks and Roles and Log Sinks Configuration.

Logging Configuration with Standard I/O

When you log to standard I/O, you can send information to multiple locations and log to one or more message types.

To configure the logging through standard I/O, you can use the Log Level panel. The Log Level panel contains the following fields:

Field	Description
Log to Standard I/O	Use this check box to turn on or turn off sending logging information to the console when the adapter is started. When you select this check box, logging information is displayed.
Log Info Messages	Use this check box to turn on or turn off sending all INFO type messages to the specified location(s).
Log Debug Messages	Use this check box to turn on or turn off sending all DEBUG type messages to the specified location(s).
Log Warning Messages	Use this check box to turn on or off sending all WARNING type messages to the specified location(s).
Log Error Messages	Use this check box to turn on or off sending all ERROR type messages to the specified location(s).

For when to turn on the `Info`, `Debug`, `Warning` or `Error` options, see Log Sinks and Roles.

Log Sinks and Roles

The adapter supports the file sink, the standard I/O sink, the network sink and the hawk sink. Each sink has the debug, error, information and warning roles.

Supported Log Sinks

The adapter supports the following log sinks at run time:

- File sink

  A file sink sends messages to a file.

- Standard I/O sink

  A standard I/O sink (stdioSink) sends messages to standard I/O.

- Network sink

  A network sink sends messages over the network.

- Hawk sink

  A hawk sink sends messages to TIBCO Hawk.

Supported Log Roles

For a selected log sink, you can add one or more log roles that decide the log levels of the log sink. The following list describes the available log roles:

- Debug

- Error

- Information

- Warning

The generated logging messages depend on the roles you add to the log sink.

Turning on the log role level affects the performance of the adapter. It is good practice to turn on the required levels only. By default, the Info, Warning, and Error levels are enabled, and the Debug level is disabled. Do not enable the Debug level unless you are requested by the TIBCO Product Support group. This option writes a great deal of debugging information to the log file, and thus reduces the speed of the adapter.

Adding Log Sinks

When you use custom roles, you can add or remove log sinks and roles in the All Log Sinks panel.

When adding a log sink, the infoRole, warnRole, and errorRole roles are added simultaneously. You can add or remove roles when needed.

Procedure

1. Open the adapter configuration.
2. Click the **Logging** tab.
3. In the All Log Sinks panel, click **Add**.
4. In the New Log Sinks dialog, select the log sink, and click **OK**.

Adding Roles

After adding a log sink, you can map one or more log roles to the log sink.

Procedure

1. Open the adapter configuration.
2. Click the **Logging** tab.
3. In the All Log Sinks panel, click the sink you want to add a role.
4. Click **Add Role**.
5. In the Configuration panel, select the role you want to add from the list.

Result

Roles are added one by one under the selected log sink in the All Log Sinks panel.

Log Sinks Configuration

The adapter supports the file sink, the standard I/O sink, the network sink and the hawk sink. Each sink has its own configuration options.

When you select a log sink from the All Log Sinks panel, the configuration options for the sink are displayed in the Configuration panel on the right. The log sink you select determines the options you have to configure.

File Sink Configuration

The Configuration panel of the file sink contains the following fields:

Field	Description
Name	The name of the sink.
Description	(Optional) A short description for the sink.
File Name	Includes the path and name of the trace file. The file name uses a `.log` extension. The default file name is *%%DirTrace%%/%%Deployment%%.%%InstanceId%%.log*.
File Limit (bytes)	The maximum size of the file, in bytes. The default value is `30000`. The maximum value is `2147483647` bytes.
File Count	The number of rollover files. The default value is `3`.
Append Mode	Use this check box to control whether to add traces to the existing file at startup. When you select the check box, traces are added to the existing file at startup. When you clear the check box, the existing file is overwritten at startup if one with the same name exists.

Standard I/O Sink Configuration

The Configuration panel of the standard I/O sink contains the following fields:

Field	Description
Name	The name of the sink.
Description	(Optional) A short description for the sink.
Output Stream	Output the information that is logged. Two options are available: • **stout**: general output information. • **stderr**: error information.

Network Sink Configuration

The Configuration panel of the network sink contains the following fields:

Field	Description
Name	The name of the sink.
Description	(Optional) A short description for the sink.
Subject	The subject of TIBCO Rendezvous messages to be sent.
Session Reference	Click **Browse** and select one of the sessions.

Hawk Sink Configuration

The Configuration panel of the hawk sink contains the following fields:

Field	Description
Name	The name of the sink.
Description	(Optional) A short description for the sink.
MicroAgent Name	The name of the microagent to trace from the hawk sink.

Monitoring

TIBCO Hawk monitors the runtime adapter. You can configure the monitoring options in the **Monitoring** tab.

The Monitoring panel contains the following fields:

Field	Description
Enable Standard MicroAgent	(Optional) Use this check box to turn on or turn off the standard TIBCO Hawk microagent. The check box is selected by default.
Standard MicroAgent Name	(Optional) The name for the standard microagent that is registered with the TIBCO Hawk system. In most cases, keep the default value. You do not have to specify the `InstanceId` property, because it is automatically set at run time by the runtime adapter.
Standard MicroAgent Timeout (ms)	(Optional) The timeout value for the standard microagent in milliseconds. The default value is `10000`. Normally you do not have to change this value. However, when machines are under extreme stress where method invocations are timing out, you can use this option to increase the timeout value.
Enable Class MicroAgent	(Optional) Use this check box to turn on or turn off the instance-specific or class-specific standard TIBCO Hawk microagent.
Class MicroAgent Name	(Optional) The name for the class microagent that is registered with the TIBCO Hawk system. In most cases, keep the default value. You do not have to specify the `InstanceId` property, because it is automatically set at run time by the runtime adapter.
Class MicroAgent Timeout (ms)	(Optional) The timeout value for the class microagent in milliseconds. The default value is `10000`. Normally you do not have to change this value. However, when machines are under extreme stress where method invocations are timing out, you can use this option to increase the timeout value.

Field	Description
Default Microagent Session	(Optional) TIBCO Rendezvous session to be used by the TIBCO Hawk microagents by default. This field is disabled and you cannot change it. The session name and the corresponding session are automatically generated. However, you can modify the session parameters if required in the **Transports** tab.

Advisories

The adapter contains TIBCO Rendezvous advisory messages and TIBCO Adapter SDK advisory messages. You can configure two types of advisory messages in the **Advisories** tab.

The **Advisories** tab consists of the All Advisories panel and the Configuration panel. The All Advisories panel lists all Rendezvous and SDK advisories, you can add or remove advisories. The Configuration panel is used to configure the selected advisory in the All Advisories panel. The Configuration panel contains the following fields:

Field	Description
Name	The name of the advisory message.
Description	(Optional) A short description for the advisory message.

Subject	TIBCO Rendezvous advisory messages have the following structure:
	
	`_RV.<class>.<source>.<category>.<role>.<condition>.<name>`
	
	For details, see *TIBCO Rendezvous Concepts*.
	TIBCO Adapter SDK advisory messages have the following structure:
	
	`_SDK.<class>.<category>.<name>` or
	`_SDK.<class>.<category>.<subject suffix>`
	
	For details, see *TIBCO Adapter SDK Programmer's Guide*.

Advanced

The **Advanced** tab contains general information about the adapter, you can specify a termination subject or topic in this tab.

Termination Subject or Topic

If a message is sent on a termination subject (if Rendezvous is the transport) or topic (if JMS is the transport), the adapter stops. The default value of the termination subject or topic is `%%Domain%%.%%Deployment%%.adfiles.%%InstanceId%%.exit`.

For information about specifying subject names, see *TIBCO Rendezvous Concepts*.

For information about publishing on a topic, see the TIBCO Enterprise Message Service product documentation.

Adapter Services

Adapter services are abstractions that describe how adapters work together with other applications. TIBCO ActiveMatrix Adapter for Files for Unix/Win (TIBCO Business Studio) supports Publication Service and Subscription Service. Each service has different configuration options.

Publication Service

Publication Service is used to process data from text files and publish the contents in real-time to the TIBCO environment.

Each Publication Service consists of the **Configuration**, **Processing**, **Schema** (exclusive for Record Transfer mode), **SFT** (exclusive for Simple File Transfer mode), **Encoding**, and **Advanced** (exclusive for Record Transfer mode) tabs.

Publication Service - Configuration Tab

The **Configuration** tab contains the Configuration, Life Cycle, Input File, Transport, and RV/JMS panels. Go over the listed fields in each panel.

Configuration

The Configuration panel contains the following fields:

Field	Description
Name	The name of Publication Service. The name is unique among all publishers in the adapter configuration. The name only contains alphanumeric characters, including the underscore (_) , and has 80 characters at most. The name cannot contain any space character, or use module properties.
Description	(Optional) A short description for Publication Service.
Transfer Mode	Determines the operation mode for Publication Service. The available options are **Record Transfer** and **Simple File Transfer**.

Life Cycle

The Life Cycle panel contains the following fields:

Field	Description
Preserve Undelivered	Determines whether an undelivered JMS message is preserved in the system queue.
Delta Publishing Mode	Only supports the Timer polling method. When Delta Publishing mode is enabled, Publication Service checks the input file on a preconfigured timer interval, and copies any new data to a work file in the WIP directory, and then processes and publishes the new data.

Field	Description
Life Cycle	Specifies if Publication Service operates continuously (**Repeating**) or just one time (**Once-only**). The **Once-only** option means to start, process the files, and then exit.
	If more than one service is defined for the adapter configuration, you cannot select **Once-Only** for the first service.
Polling Method	Specifies how Publication Service is triggered to start processing the files When you select **Repeating** from the **Life Cycle** list. Publication Service can be triggered using a timer or a message. The Timer method of triggering is available for both transport types. Triggering Publication Service using a message depends on the transport type. If the transport type is TIBCO Rendezvous, a TIBCO Rendezvous message can be used to trigger Publication Service. If the transport type is JMS, a JMS message sent on a topic can be used to trigger Publication Service.
Polling Interval (seconds)	Specifies the amount of time in seconds until the next file scan is repeated. This is available when selecting **Timer** from the **Polling Method** list.
Polling Subject	Specifies the subject or topic name on which the TIBCO Rendezvous message or the JMS message is sent to trigger Publication Service. If the transport type is TIBCO Rendezvous, the name of this field is **Polling Subject**. If the transport type is JMS, the name of this field is **Polling Destination**.

Input File

The Input File panel contains the following fields:

Field	Description
Input Directory	Publication Service searches and processes the files in this directory, and then publishes the files. This directory can be different from the directories specified for the **Working Directory** and **Completion Directory** fields. The input, working, and completion directories can have an absolute path name or a relative path name. When a relative path name is used, it is relative to the starting directory of the runtime adapter.
	On Unix, the processing directories such as the input, working, done or output directories are specified on the same file system. Only the input directory is scanned for files that match the criteria. It is not recursively traversed.

Field	Description
Recognition Method	Specifies the mechanism for finding the desired input file(s). The following options are available: • **By file name** Processes the file that exactly matches the value given in the **File Name** field. • **By Wildcard via ICU Regular Expressions** Processes the file that matches the ICU regular expression specified in the **File Name** field. • **By prefix + extension** Processes the files that match the criteria that you have defined in the **File Prefix** and **File Extension** fields. • **By trigger** Processes the files that match the criteria that you have defined in the **File Prefix**, **File Extension**, and **Trigger File Extension** fields. When selecting the **By trigger** option, the adapter processes the input files only after they are ready. Without this, the adapter might process the files in the input directory before files are created, written, or closed by the third-party applications. The trigger file recognition method helps to avoid this situation. With the trigger recognition method, the adapter only processes the input files after the trigger files are created. The filename or file prefix cannot contain path information.

Field	Description
File Name	This field is available in the following cases: • When you select **By file name** from the **Recognition Method** list. In this case, the adapter processes the file that matches exactly the value given in this field. • When you select **By Wildcard via ICU Regular Expressions** from the **Recognition Method** list. In this case, you can use ICU regular expressions in the **File Name** field. The following list describes two examples of using ICU regular expressions: • Prepare the following files in the input directory: `text0.txt`, `text1.txt`,..., to `text10.txt`. If the input filename is `text\d\.txt`, the input files named from `text0.txt`, `text1.txt`,..., to `text9.txt` are published. • Prepare the following files in the input directory: `A6.0.0.txt`, `A6.1.0.txt`, `A6.2.0.txt`, `A6.8.0.txt`, `A6.0.0.log`, and `A6.1.0.log`. If the input filename is `A6\.[01]\.0\.(txt|log)`, the input files named `A6.0.0.txt`, `A6.1.0.txt`, `A6.0.0.log`, and `A6.1.0.log` are published. Wildcard is different from regular expressions and is not supported. For example, `*.txt` must be specified as `.*\.txt` in the regular expressions format.
File Prefix	This prefix is used to locate the input file in the input directory. Any file matching the specified criteria is processed. To activate the file prefix, select **By prefix + extension** or **By trigger** from the **Recognition Method** list.
File Extension	This field is available only when you select **By prefix + extension** or **By trigger** from the **Recognition Method** list.
Trigger File Extension	This field is available only when you select **By trigger** from the **Recognition Method** list.

Transport

The Transport panel contains the following fields:

Field	Description
Session Reference	Every adapter configuration has one or more sessions. Sessions encapsulate connections to TIBCO Rendezvous and other messaging sources. The session object shown in this field is initially supplied by the adapter, depending on the selected quality of service selected. You can change the session by clicking the ▭ button.

Field	Description
Endpoint Reference	The default value is `#producer.PublicationServiceEndpoint`. You cannot change the value. When you delete related services, the endpoint reference is also deleted. For more information about endpoint reference objects, see *TIBCO Designer Palette Reference*.

RV

The RV panel is displayed when you select TIBCO Rendezvous session in the **Session Reference** field in the Transport panel. The RV panel contains the following fields:

Field	Description
Subject	The name of message subject.
Wire Format	Publishers and subscribers can only send and receive data if they agree on a specific wire format. When sending a message, select one of the following options: • **ActiveEnterprise Message** Control information for validation is sent in the message. If no control information is included, an exception is returned to the subscriber. ActiveEnterprise standard wire format provides class information and packing rules for the TIBCO ActiveMatrix Adapter SDK set of data types. In this format, ActiveEnterprise components can perform extra validation on messages sent or received. This option is unavailable for Simple File Transfer mode. It is available only for Record Transfer mode. • **XML Message** The XML Message wire format conforms to specifically constructed and fully compliant XML Schema (XSD) based on the existing definition of the ActiveEnterprise schema. This field is only enabled in Record Transfer mode. In Record Transfer mode, you can select **ActiveEnterprise Message** or **XML Message** for the Rendezvous transport type. In Simple File Transfer mode, this field is read-only. By default, it is **ActiveEnterprise Message** for the Rendezvous transport type, and **XML Message** for the JMS transport type.

JMS

The JMS panel is displayed when you select JMS session in the **Session Reference** field in the Transport panel. The JMS panel contains the following fields:

Field	Description
Destination	By default, a service uses a dynamic destination that is generated using the Domain and Deployment module properties, the adapter acronym, the adapter configuration name, and the service name. If you use this default dynamic destination, make sure that 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 runtime adapter.
Wire Format	Publishers and subscribers can only send and receive data if they agree on a specific wire format. 📝 This field is only enabled in Record Transfer mode. In Record Transfer mode, you can only select **XML Message** for the JMS transport type.
Delivery Mode	For Publication Service, a message is marked as persistent or non-persistent. • **Persistent** A message marked as persistent is available to a JMS client even if the JMS server goes down. • **Non-Persistent** A message marked as non-persistent is unavailable to a JMS client if the JMS server goes down. Messages with 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, no subscribers requests to resend messages when a server fails. In this case, you do not have to save messages, thus improving performance because disk I/O is not required.

Publication Service - Processing Tab

The **Processing** tab contains the Processing and Operation for Processing Script panels. Check the listed fields in each panel.

Processing

The Processing panel contains the following fields:

Field	Description
Working Directory	Publication Service uses this directory to process files that match the criteria. Based on the option selected in the **Post Processing** field, the file is either copied or moved into this directory. If you select **Leave as is** from the **Post Processing** list, the file is copied. If you select **Delete** or **Move to**, the file is deleted or moved. 📝 For adapter configurations, if the files processed by Publication Service are independent of each other, Publication Service can share the input, working, completion, and error directories. Otherwise, these directories must be unique.

Field	Description
Completion Directory	This field is available only when you select **Move to** from the **Post Processing** list. After the file in the working directory is processed, the file is moved to this directory.
Error Directory	When you use Simple File Transfer mode, this field is mandatory. When you use Record Transfer mode, this field is not mandatory. However, if specified, an XML file containing the name of the input file and the error details is created in this directory. For details about the usage and contents of this directory, see Error Handling.
Progress Directory	The directory where the progress file is written to. It applies to both Record and Simple File transfer modes. If no directory is specified in this field, the progress file is created in the directory where the adapter is started.
Post Processing	An action to be applied to the file that is currently in the working directory after Publication Service has finished processing the file. The following list describes the available postprocessing actions: • **Move to** Moves a file from the working directory to the completion directory. • **Delete** Deletes a file from the working directory. • **Leave as is** Deletes a file from the working directory (since files in the working directory are copies). The corresponding file in the input directory is left as it is.
Add TimeStamp to File Name	With this option, you can append the date and time of a file that is moved to the completion directory. The format of the date and time is YYYYMMDDHHMMSSmm.

Operation for Processing Script

The Operation for Processing Script panel contains the following fields:

Field	Description
Pre Processing Script File	The name of the script that is executed before the input file can be processed by the adapter. You can process the input file before it is processed by the adapter. Click **Browse** to locate and load the script file.

Field	Description
Pre Processing Arguments	The arguments passed to the preprocessing script file. Arguments are strings and are optional.
	The command line syntax of the arguments that is passed to the script is:
	`Script_filename Pre Processing Arguments`
	For example,
	`script.tcl inputFile0364.txt argument1 argument2 ...`
	where
	`script.tcl` is the script filename
	`inputFile0364.txt` is the name of the reprocessed file
	`argument1` is the first argument, and is followed by other arguments.
	The preprocessing script file reads the input file, renames the file, makes required modifications, and writes to the original filename.
	If five files are in the input directory, the adapter runs the script five times, once for each file. The adapter processes the files in ascending order based on their names. The adapter sorts the files according to their names alphanumerically in ascending order. It is case sensitive, and the upper case is followed by the lower case.
	For example, if the following files exist in the input directory:

	`1.csv`
	`11.csv`
	`111a.csv`
	`22.csv`
	`11a.csv`
	`11b.csv`
	`22b.csv`

	The adapter processes the files in the following order:

	`1.csv`
	`11.csv`
	`111a.csv`
	`11a.csv`
	`11b.csv`
	`22.csv`
	`22b.csv`

	During preprocessing, when the preprocessing script finds the file unsuitable for processing, the adapter does not process the file. The adapter logs feedback from the preprocessing script.
Post Processing Script File	The name of the script that is executed after the input file is processed by the adapter. Click **Browse** to locate and load the script.

Field	Description
Post Processing Arguments	The arguments you want to pass to the postprocessing script. Arguments are strings and are optional. The sequence of arguments passed to the postprocessing script is determined as follows: • If the transfer mode is Record Transfer, the argument sequence contains the name of the file, the arguments specified in the postprocessing arguments, and the status. The status succeeds if the publisher processes the file successfully. The status fails if the publisher has problems (for example, parsing) processing the file. • If the transfer mode is Simple File Transfer, the argument sequence contains the name of the file and is followed by the arguments specified in the postprocessing arguments.

Publication Service - Schema Tab

The **Schema** tab is available only for Record Transfer mode.

You can associate a read schema created for Publication Service by clicking **Add Wire Schema** ⬚, and selecting the read schema from the Select Schema Model window.

You can specify the option for filtering the fields in the file records. This is done by expanding the read schema tree and the subsequent file record and deselecting the fields from the file records. A check box under the Use? column is provided for each field for this purpose. Because TIBCO ActiveEnterprise applications use wire schema to exchange data, it is a good practice to inform the other adapter users about the change.

📋 When fields are filtered, it alters the wire schema.

Publication Service - SFT Tab

The **SFT** tab is available only for Simple File Transfer mode.

The **SFT** tab contains the following fields:

Field	Description
Transmission Buffer Size	The buffer size to use for each data message chunk. The default value is `32`.
Progress File Name	This parameter is optional. When the progress filename is blank, Publication Service does not restart from where it is suspended. When the progress filename is specified, Publication Service continues to process from where it is suspended. The default progress filename is `__InstanceNameServiceName.prg` When a path name is not specified as part of the progress filename, the progress file is created in the directory, which is specified in the **Progress Directory** field of the **Processing** tab. 📋 The progress filename is specified in the LATIN_1 or ASCII character set only.

Field	Description

File Transfer Mode	Two methods of transferring files are available:
	
	• **Binary**
	
	  This mode is used for transferring any type of files, and the file contents must not be altered during file transfer.
	
	• **Text**
	
	  This mode is used to transfer text files. When transferring files across platforms, the file created on the receiving platform might be altered to reflect the platform characteristics. For example, if a text file is transferred from Windows NT to Unix, the \r\n is replaced by \n and vice versa.

Publication Service - Encoding Tab

The **Encoding** tab is available for both Record Transfer mode and Simple File Transfer mode.

The **Encoding** tab contains the following fields:

Field	Description
File System Encoding	Provides aliases for the following commonly used encoding for file and directory names: **ASCII**, **ISO8859-1**, **UTF-8**, **Shift JIS (CP943)**, **Shift JIS (TIBCO)**, **EUC-JP**, **Big5**, and **Other**. When you select **Other**, you can enter an encoding string that is not present in the list. For a list of encoding strings, see *TIBCO ActiveMatrix Adapter for Files for Unix/Win Concepts*. When an invalid or unsupported encoding value is specified, an error occurs at run time. The default option is **UTF-8**, and only the default option is supported.
File System Encoding Other	This field is available only when you select **Other** in the **File System Encoding** list. Enter an encoding string that is not present in the list. For a list of encoding strings, see *TIBCO ActiveMatrix Adapter for Files for Unix/Win Concepts*. The supported option for the **File System Encoding** field is **UTF-8**, so you cannot enter value for this field.

Field	Description
File Content Encoding	Provides aliases for the following commonly used encoding for file contents: **ASCII**, **ISO8859-1**, **UTF16_BigEndian**, **UTF16_LittleEndian**, **UTF-8**, **Shift JIS (CP943)**, **Shift JIS (TIBCO)**, **EUC-JP**, **Big5**, and **Other**.
	If you want to transfer a file in ASCII encoding between open systems in Simple File Transfer mode, select **ASCII** from the **File Content Encoding** field.
	If you want to transfer a file in ASCII encoding between an open system and a mainframe system in Simple File Transfer mode, select **ISO8859-1** from the **File Content Encoding** field.
	When you select **Other**, you can enter an encoding string that is not present in the list. For a list of encoding strings, see *TIBCO ActiveMatrix Adapter for Files for Unix/Win Concepts*.
	The File Content Encoding option is unavailable when you select **Simple File Transfer** mode in the **Configuration** tab and select **Binary** from the **File Transfer Mode** list in the **SFT** tab.
	When an invalid or unsupported encoding string value is specified, an error is displayed at run time.
File Content Encoding Other	This field is available only when you select **Other** in the **File Content Encoding** list.
	By default, the repository encoding is ISO-8859-1. When the file system or file content encoding is set to any encoding other than ASCII or ISO-8859-1, you have to set the repository encoding to UTF-8. If the repository encoding field is not set correctly, messages might be corrupted.
End of Line	Select the method according to how the lines in the input file are separated.
	• **System**
	Uses a carriage return (new line) to mark the end of a line.
	• **User Defined**
	Uses custom end of line characters to mark the end of a line.
	Currently, no facility is provided to distinguish custom end of line characters that are not actual characters.
	• **System and User Defined**
	Uses a combination of carriage returns and custom characters to mark the end of a line.
User Defined EOL	This field is available only when the **End of Line** field is not **System**. Enter the characters to mark the end of a line.
	The **End of Line** and **User Defined EOL** fields are unavailable when the transfer mode is Simple File Transfer.

Publication Service - Advanced Tab

The **Advanced** tab is available only for Record Transfer mode. The tab has the Advanced, Business Document, and Processing and Publishing panels. Check the listed fields in each panel.

Advanced

The Advanced panel contains the following fields:

Field	Description
Input Buffer Max Size	Specifies the amount of data. Publication Service reads the data from the input file every time it reads the file. Usually, this field uses the default value.
Document Delay (milliseconds)	Publication Service occasionally overwhelms the underlying messaging infrastructure by publishing messages faster than they are received. Under these circumstances, Publication Service uses the flow control. Specify the document delay in milliseconds, and the delay is applied when publishing a message. The message is either **Object (MInstance)** or **Business Document (MBusinessDocument)**. Publication Service publishes a MInstance or MBusiness Document every time a delay timer triggers.
Checkpoint Restart	When you select this check box, you can enable the checkpoint restart capability for Publication Service. If the adapter crashes when processing a file, Publication Service continues to process the file from where it crashed upon restart. Clearing the check box means a higher throughput is possible, but you have to restart the service manually. For more information, see CheckPoint Restart.

Business Document

The Business Document panel contains the following fields:

Field	Description
Grouping Factor	Specifies how many objects are grouped when publishing a business document. You can group multiple file records into one business document message. Objects belonging to the same file are grouped together.
Data Format	Select the data format for message to be published. You can select **Object (MInstance)** or **Business document (MBusinessDocument)** from the **Data Format** list. When selecting **Business document**, MInstances are batched before sending. For a given size of MInstance, the number of MInstances per business document varies to achieve the desired throughput. The performance of Subscription Service improves when it receives a MBusinessDocument.

Field	Description
Business Document Name	This field is available only when the data format is **Business document (MBusinessDocument)**. It is the name of the Business document that is to be published, and is a required attribute for the BusinessDocument format of messages.

Processing and Publishing

The Processing and Publishing panel contains the following fields:

Field	Description
Synchronous Mode	Select this check box if Publication Service operates in synchronous mode. Clear this check box if Publication Service operates in asynchronous mode. Each Publication Service works on its own thread; therefore, operation in synchronous mode or asynchronous mode makes no difference.
Publication Semantics	When you select the check box under this field, Publication Service publishes an object only if all of its associations have no errors.

Publication Semantics

If you want to publish an object only if all the associations (children) have no issues, select the **Only publish an identified object whose associations, if defined, have no errors** check box under the **Publication Semantics** field.

For example, if a ReadSchema is defined as follows:

```
H
  D
```

where H and D are delimited file records (identified by a constant) with the following fields and types:

```
H
field_1 String (Header)
field_2 ui4
D
field_1 String (Detail)
field_2 ui4
```

The `input.txt` input filename contains:

```
Header, 10
Detail, -20
```

Since `Detail, -20` has a validation issue, the entire object is not published when this option is enabled. If this option is disabled, and the object containing only `Header, 10` is published.

Subscription Service

Subscription Service is used to listen for messages in the TIBCO environment, process received messages, and write the contents to text files.

Each Subscription Service consists of the **Configuration**, **Processing**, **Schema** (exclusive for Record Transfer mode), **SFT** (exclusive for Simple File Transfer mode), and **Encoding** tabs.

Subscription Service - Configuration Tab

The **Configuration** tab contains the Configuration, Output Setting, Transport, and RV/JMS panels. Check the listed fields in each panel.

Configuration

The Configuration panel contains the following fields:

Field	Description
Name	The name of Subscription Service. The name is unique among other subscribers assigned to the adapter configuration. The name only contains alphanumeric characters, including the underscore (_), and has 80 characters at most. The name cannot contain any space character, or use module properties.
Description	(Optional) A short description for Subscription Service.
Transfer Mode	This option determines the operation mode for Publication Service. The available options are **Record Transfer** and **Simple File Transfer**.

Output Setting

The Output Setting panel contains the following fields:

Field	Description
Wip Creation Mode	The creation mode of the file in the working directory. This option is unavailable when you select Simple File Transfer mode. Subscription Service uses the working directory to create the file. Based on the semantics of Wip Creation mode options, the file is moved to the output directory with the following options: • **Append messages to file, close on Timer** Received messages are appended to a file created in the working directory. The file is closed and moved to the output directory on receiving the timer event specified in the **Elapsed** field. And then a new file is created in the working directory. • **Append messages to file, close on rvMessage** Received messages are appended to the file created in the working directory. The file is closed and moved to the output directory on receiving a TIBCO Rendezvous message on the subject specified in the **Closing Subscriber** field. And then a new file is created in the working directory. • **Append messages to file, close on jmsMessage** Received messages are appended to the file created in the working directory. The file is closed and moved to the output directory on receiving a JMS message. And then a new file is created in the working directory. • **Append messages to file, close on Business Document Lot End** Closes the working file of Subscription Service when the **LotEnd** field is set to `true` and a business document is received. And then a new file is created in the output directory. You can use this option when transferring files. For more information, see the transferReader and transferWriter example in *TIBCO ActiveMatrix Adapter for Files for Unix/Win Examples*. • **One message per file** Each received message is written to a file and then moved to the output directory.
Output Creation Mode	Two options are available: • **Overwrite (always create new file)** Overwrites the file when moving a file from the working directory to the output directory. • **Append (if same file name exists)** Appends the text that is received from Publication Service to the file in the output directory when moving a file from the working directory to the output directory.
File Name	The name of the output file that is to be created. This field is available only when you select **From Configuration** in the **File Name Selection** list. 📝 The filename cannot contain path information.

Field	Description
File Name Selection	With Record Transfer mode, only the **From Configuration** option is available. You have to specify a name in the **File Name** field. Subscription Service uses the name that you specify in the **File Name** field as the output filename.
	With Simple File Transfer mode, you can select **From Configuration** or **From Message**. When you select **From Message**, Subscription Service uses the same filename that is available in the message as the output filename.
Close file after time elapsed (minutes)	The amount of time before the file is closed in the working directory. You can type a specific time in this field or use a module property. This field is available only when you select **Append messages to file, Close on timer** in the **Wip Creation Mode** list.
Message Time Out (minutes)	This field is available only if you select **Append messages to file, close on Timer** in the **Wip Creation Mode** list. It specifies the time to wait for incoming messages. After a message is received, the next message must be received within the timeout that you have specified, or Subscription Service closes the work-in-progress file and transfers it to the output directory. The default value 0 indicates no timeout.
Closing Subscriber	This field is available only if you select **Append messages to file, close on rvMessage** or **Append messages to file, close on jmsMessage** in the **Wip Creation Mode** list. The subscriber used to receive and signify the output file is closed.

Transport

The Transport panel contains the following fields:

Field	Description
Session Reference	Every adapter configuration has one or more sessions. 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 clicking the ⋯ button.
Endpoint Reference	The default value is `#consumer.SubscriptionServiceEndpoint`. You cannot change the value. When you delete related services, the endpoint reference is also deleted.
	For more information about endpoint reference objects, see *TIBCO Designer Palette Reference*.

RV

The RV panel is displayed when you select TIBCO Rendezvous session in the **Session Reference** field in the Transport panel. The RV panel contains the following fields:

Field	Description
Subject	The name of the message subject.
	By default, a service uses a message subject that is generated using the Domain and Deployment module properties, the adapter acronym, the adapter configuration name, and the service name. If you use the default subject, ensure that the values for Domain and Deployment are not empty. You can type a TIBCO Rendezvous subject name different from the default.
Wire Format	Publishers and subscribers can only send and receive data if they agree on a specific wire format. When sending a message, select one of the following options:
	• **ActiveEnterprise Message**
	Control information for validation is sent in the message. If no control information is included, an exception is returned to the subscriber. ActiveEnterprise standard wire format provides class information and packing rules for the TIBCO ActiveMatrix Adapter SDK set of data types. In this format, ActiveEnterprise components can perform extra validation on messages sent or received. This option is unavailable for Simple File Transfer mode of the adapter. It is available only for Record Transfer mode.
	• **XML Message**
	It is a text message which contains XML data that is validated against an XSD. This option is available only for Record Transfer mode.
	This field is available only with Record Transfer mode. With Record Transfer mode, you can select **ActiveEnterprise Message** or **XML Message** for the Rendezvous transport type.
	With Simple File Transfer mode, this field is read-only. By default, it is **ActiveEnterprise Message** for the Rendezvous transport type, and **XML Message** for the JMS transport type.

JMS

The JMS panel is displayed when you select JMS session in the **Session Reference** field in the Transport panel. The JMS panel contains the following fields:

Field	Description
Destination	By default, a service uses a dynamic destination that is generated using the Domain and Deployment module properties, the adapter acronym, the adapter configuration name, and the service name. If you use the default dynamic destination, ensure that 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 runtime adapter.

Field	Description
Wire Format	Publishers and subscribers can only send and receive data if they agree on a specific wire format. This field is available only with Record Transfer mode. With Record Transfer mode, you can only select **XML Message** for the JMS transport type.
Delivery Mode	For Subscription Service, a message is marked as durable or nondurable. • **Durable** A durable service is registered with the JMS server. Messages sent to a durable service are held by the JMS server until they are received by the service. If Subscription Service crashes, messages are received when the service resumes. • **Non-Durable** Messages sent to a nondurable service are not held by the JMS server. If Subscription Service crashes, messages which arrived on the JMS server are not received when the service resumes. The semantics for these fields are more complex than the explanation given here. For more information, see the TIBCO Enterprise Message Service documentation.

Subscription Service - Processing Tab

The **Processing** tab contains the Processing, Operation for Process Script and Options for Modifying Output Filename panels. Check the listed fields in each panel.

Processing

The Processing panel contains the following fields:

Field	Description
Working Directory	Specifies the directory where contents for the actual output file is composed from received messages. When the files generated by Subscription Services are independent of each other, the services share the working, output and error directories. Otherwise, the directories must be unique. On Unix, the processing directories such as the working directory, and the done or output directory are required to be on the same file system.
Output Directory	Specifies the directory where files in the working directory are moved to.
Error Directory	When you use Simple File Transfer mode, this field is mandatory. When you use Record Transfer mode, this field is not mandatory. However, if specified, an XML file containing the name of the input file and the error details are created in this directory. For details about the usage and contents of this directory, see Error Handling.

Field	Description
Progress Directory	Specifies the directory where the progress file is written to. It applies to both Record and Simple File transfer modes. If no directory is specified in this field, the progress file is created in the directory where the adapter is started.

Operation for Process Script

The Operation for Process Script panel contains the following fields:

Field	Description
Post Processing Script File	Specifies the name of the script that is executed after the file is moved to the output directory. Click **Browse** to locate and load the script.
Post Processing Arguments	Specifies the arguments you want to pass to the postprocessing script. The sequence of arguments contains the name of the file, followed by the arguments specified in the **Post Processing Arguments** field.

Options for Modifying Output Filename

The Options for Modifying Output Filename panel provides options to modify the name of the generated output file by appending a timestamp or sequence number. This panel contains the following fields:

Field	Description
Append	Three options are available: • **None** Does not append anything to the filename. • **Timestamp** Appends a timestamp to the filename. The format is YYYYMMDDHHMMSSmm. Selecting this option activates the **Location** field. • **Sequence Number** Appends a sequence number to the filename. Selecting this option activates the **Location** and **Sequence Number Width** fields.
Location	Two options are available: • **Prefix** Appends the timestamp or sequence number to the filename as a prefix. • **Suffix** Appends the timestamp or sequence number to the filename as a suffix. The separator between the prefix and suffix is assumed to be a period (.).

Field	Description
Sequence Number Width	Specifies the width of the sequence number. For example, when you select 4 as the width of the sequence number, the number that is appended to the file is 0000. The sequence number begins with 0 and is incremented for each file that is moved to the output directory. Upon reaching the maximum value for a given width, the sequence number rolls over.
	When the adapter is restarted, Subscription Service remembers the previous sequence number and continues from there.

Subscription Service - Schema Tab

The **Schema** tab is available only for Record Transfer mode.

You can associate a write schema that has been created to Subscription Service by clicking the [icon] icon, and selecting the write schema from the Select Schema Model window. Repeat this to associate more write schemas.

This tab displays the structure of the selected schema for the adapter service. You can also use this tab to design new schemas wherever applicable.

Subscription Service - SFT Tab

The **SFT** tab is available only for Simple File Transfer mode. The **SFT** tab contains the **File Transfer Mode** field.

File Transfer Mode

File Transfer mode has two options:

- **Binary**

  This mode is used to transfer files, and the file contents cannot be altered during the file transfer process.

- **Text**

  This mode is used to transfer text files. When transferring files across platforms, a file created on the receiving platform is altered to reflect the platform characteristics. For example, if a text file is transferred from Windows NT to Unix, the \r\n is replaced by \n and vice versa.

This mode reflects the settings made for Publication Service. The subscriber is associated with Publication Service.

Subscription Service - Encoding Tab

The **Encoding** tab is available for both Record Transfer mode and Simple File Transfer mode.

By default, the repository encoding is ISO-8859-1. When the file system or file content encoding is set to any encoding other than ASCII or ISO-8859-1, you have to set the repository encoding to UTF-8. If the repository encoding field is not set correctly, messages might be corrupted.

The **Encoding** tab contains the following fields:

Field	Description
File System Encoding	Provides aliases for the following commonly used encoding for file and directory names: **ASCII**, **ISO8859-1**, **UTF-8**, **Shift JIS (CP943)**, **Shift JIS (TIBCO)**, **EUC-JP**, **Big5**, and **Other**.
	When you select **Other**, you can enter an encoding string that is not present in the list. For a list of encoding strings, see *TIBCO ActiveMatrix Adapter for Files for Unix/Win Concepts*.
	When an invalid or unsupported encoding value is specified, an error occurs at run time.
	The default option is **UTF-8**, and only the default option is supported.
File System Encoding Other	This field is available only when you select **Other** in the **File System Encoding** list. Enter an encoding string that is not present in the list. For a list of encoding strings, see *TIBCO ActiveMatrix Adapter for Files for Unix/Win Concepts*.
	The supported option for the **File System Encoding** field is **UTF-8**, so you cannot enter value for this field.
File Content Encoding	Provides aliases for the following commonly used encoding for file contents: **ASCII**, **ISO8859-1**, **UTF16_BigEndian**, **UTF16_LittleEndian**, **UTF-8**, **Shift JIS (CP943)**, **Shift JIS (TIBCO)**, **EUC-JP**, **Big5**, and **Other**.
	When you select **Other**, you can enter an encoding string that is not present in the list. For a list of encoding strings, see *TIBCO ActiveMatrix Adapter for Files for Unix/Win Concepts*.
	The File Content Encoding option is unavailable when you select **Simple File Transfer** mode from the **Transfer Mode** list (in the **Configuration** tab) and select **Binary** from the **File Transfer Mode** list (in the **SFT** tab).
	When an invalid or unsupported encoding string value is specified, an error is displayed at run time.
File Content Encoding Other	This field is available only when you select **Other** in the **File Content Encoding** list.

Field	Description
End of Line	Select method according to how the lines in the input file are separated. • **System** Uses a carriage return (new line) to mark the end of a line. The separator is dependent on run time. It follows the DOS convention on Microsoft platforms, and the Unix convention on Unix platforms. • **DOS** A carriage return and line feed. • **UNIX** A line feed. • **Unicode** Unicode Line Separator. Use only in conjunction with UTF8, UTF16BE, or UTF16LE file encoding. • **User Defined** You can define characters to mark the end of line. Specify the characters in the **User Defined EOL** field. The End of Line option is unavailable when the transfer mode is Simple File Transfer.
User Defined EOL	This field is available when you select **User Defined** in the **End of Line** list. Enter characters to mark the end of a line. The **End of Line** and **User Defined EOL** fields are unavailable when the transfer mode is Simple File Transfer.
Byte Order Mark	Byte Order Mark applies to **UTF-8** only. The option is not supported in Java.

Defining Schemas

TIBCO ActiveMatrix Adapter for Files for Unix/Win (TIBCO Business Studio) provides two types of schema: read schema and write schema. Read schema includes ReadSchema(Delimited) and ReadSchema(Positional), and write schema includes WriteSchema(Classic) and WriteSchema(Manual). You can define read schemas and write schemas in Business Object Editor.

A typical workflow of defining schema includes creating a schema, defining the relations between file records, and configuring the schema.

Defining Read Schemas

Two types of read schemas are available: ReadSchema(Delimited) and ReadSchema(Positional), you can define them using Business Object Editor.

Defining or generating a read schema is done at adapter configuration level. Therefore, the defined read schema can be shared across Publication Service services of that adapter configuration.

In Record Transfer mode, each input data record must not exceed 256 KB in size.

To define a read schema, complete the following tasks:

Procedure

1. Creating a Read Schema

2. Defining Relationships

3. Configuring a read schema

   See Configuring a Read Schema (Delimited) or Configuring a Read Schema (Positional).

Creating a Read Schema

You can create a read schema in Business Object Editor.

When the configuration of a read schema is changed, you can click the **Refresh Business Object** link (in the **Adapter Services** tab) to refresh the read schema and its relation with Publication Service.

Procedure

1. In the Project panel, double-click an adapter configuration which you want to add a read schema to.

2. In the Adapter for Files Configuration panel, click the **Business Object Editor** link in the Getting Started panel.

3. In the File Schemas panel, click **Add Read Delimited Record** or **Add Read Positional Record**.

4. Configure the record added in Step 3.

   For detailed information about configuration, see Configuring a Read Schema (Delimited) or Configuring a Read Schema (Positional).

   Optionally, you can specify a new name for the schema and provide a description. Select the **Is Base Records** check box if you want to create a file record that contains a Record data type, that is, for Container records. For information about Container Records and Base Records, see *TIBCO ActiveMatrix Adapter for Files for Unix/Win Concepts*.

5. Associate the read schema with Publication Service of an adapter configuration.

   For detailed information about associating a read schema with Publication Service, see Publication Service - Schema Tab.

Defining Relationships

A read schema can contain a combination of delimited file records or positional file records. You can create file records and define relationships among them.

When you create the first file record for a read schema, and then create additional records for the same read schema, these records form a relationship automatically. The relationship depends on how you create file records. The first file record created in a read schema is referred to as the root record. A read schema has only one root record. All the other records created after the first file record are referred to as child records.

A root record can have many child records, and each child record can further have child records. The child file records defined immediately under the root record have a `Position From Header` attribute. With this attribute, the immediate children of the root record in the actual data file are displayed before or after the root record.

The `Position From Header` attribute is only available for child records defined under the root record.

For example, if a read schema is defined as follows:

```
Order
        Customer
        Item
```

The data file contains the following records, and the `Position From Header` attribute for the Customer file record and the Item file record is defined as **Begin**.

```
Order
Customer
Item
Item
Order
Customer
Item
```

If the data file contains the following records, and the `Position From Header` attribute for the Customer file record and the Item file record is defined as **End**.

```
Customer
Item
Item
Order
Customer
Item
Order
```

The `Position from Header` attribute set for the child records is either **Begin** or **End**, and it cannot be in combination. For example, you cannot set **Begin** for the Customer file record and **End** for the Item file record.

Configuring a Read Schema (Delimited)

You can configure a delimited read schema using Business Object Editor.

Procedure

1. Configure attributes in the Reader Delimited Business Object.

   For detailed information about these attributes, see Read Schema (Delimited) Reference.

2. In the Attributes panel, click **Add** to add some fields.

   The name of the field is generated automatically as field, field1, field2 and so on. You can edit the field names. With the **Add**, **Remove**, **Up** and **Down** buttons, you can add or remove attributes, move attributes up and down.

3. Configure the Type, Constant, and Sample Value columns for the added fields.

- Type

  The data type in the corresponding field in the file is converted before publication.

  If the dateTime type is used, you must specify the pattern and locale: right-click the field name and select **Set locale and pattern**, and then select a supported locale and pattern from the list.

  The record type is used to interpret a delimited record embedded within another delimited record. Currently, delimited records that are identified by a constant field value support this feature. When this is selected, it brings up a pop-up dialog which displays the base records that are defined.

  > When using the record data type, you have to define the base records first.

- Constant

  Available only when you have selected **Field Value** in the **Identify Type** field. Enter a constant value for the appropriate field.

- Sample Value

  The value is set to <unknown> if the Display Wizard panel is not extended. It shows the actual value when the Display Wizard panel is used and the line in the file is highlighted. You can remove or shuffle the fields.

4. Create fields and the names of the fields using the Display Wizard panel.

   a) Select the encoding of the file content.
      Before opening the file, you have to select the file content correctly.
   b) Click **Browse** to navigate to the file.
   c) After identifying the desired file, click **Open**.
      The file content is displayed in the Display Wizard window.
   d) Highlight the line where you want to create a file record, and click **Create**.

   This creates fields corresponding to the line using the specified delimiter as the reference. The generated fields are field, field1, field2, and so on by default.

   The Generate Names feature can be used if the records in the file are actually headers or structural representations of the actual data record. When this feature is used, the actual field names specified in the structural representation are generated and used instead of the default names.

Read Schema (Delimited) Reference

The delimited read schema has some attributes, you can check the following table for the attributes and corresponding description:

Attribute Name	Description
Name	The file record names have alphanumeric characters only and have 80 characters at most. Each name is unique within the adapter configuration.
Description	(Optional) A short description for the delimited file record.
Position From Header	This field is available only for records that are defined as immediate child record of the root record. Select **Begin** or **End** based on the structure of the actual records in the data file.
Strip Blanks	Select this check box to remove the blank spaces for each field identified for a record in a data file.

Attribute Name	Description
Repeating Delimiter	Specifies whether to ignore repeated delimiters. Select this check box to translate any repeated delimiters between fields as only one field separator. Clear this check box to translate repeated delimiters as empty fields.
Delimiter	Specifies the separator between fields within each line of a file for the line that is parsed. **Space**, **Tab** and **Other** are the available options. If you select **Other**, the **Delimiter Other** field becomes available.
Delimiter Other	This field is available only if you select **Other** from the **Delimiter** field. Specifies the delimiter to be used. You can specify single or multiple characters as a delimiter.
	To distinguish delimiter characters from actual field values, you can enclose data fields including the delimiter characters in double quotes and set the `adfiles.quotedField` property to `ON` in the `adfilesagent.tra` file (in the `TIB_ADFILES_HOME\bin` directory).
	For example, a `John, "3301 Hillview Ave, Palo Alto CA"` comma-delimited record is treated as two fields because of the double quotes.
	📝 Quoted field is supported with regular records only. Container records and End-of-line cannot be enclosed in double quotes.
Identifier Type	Specifies the method to identify a record. **Field Value** (constant field value) and **Number of Fields** are the available options.
	When the **Identifier Type** field is set to **Field Value**, the constant attribute for one of the fields is specified when defining the fields for the records. When the **Identifier Type** field is set to **Field Value**, Publication Service can also be configured to check the field count in addition to the constant field value in validating a delimited record.
	For example, a read schema defines a record to contain three String fields with the first field containing a constant string "Order". A record, such as "Order,ID1234,Aug20" is accepted by Publication Service. However, a record with a valid constant Order but contains four fields, such as "Order,ID1234,Aug20,CA" is rejected. The following error is logged: `1 lines couldn't be interpreted`.
	To use this feature, you have to set the `adfiles.matchFieldCountAndRecordLength` property to `ON` in the `adfilesagent.tra` file (in the `TIB_ADFILES_HOME\bin` directory).
	📝 Only one constant field value can be used to identify a record.
Is Base Record	Select this check box to indicate the current record is a base record.

Configuring a Read Schema (Positional)

You can configure a positional read schema using Business Object Editor.

Procedure

1. Configure attributes in the Reader Positional Business Object.

   For detailed information about these attributes, see Read Schema (Positional) Reference.

2. Click **Add** to add some fields.

   The name of the field is generated automatically as field, field1, field2 and so on. You can edit the field names. With the **Add**, **Remove**, **Up** and **Down** buttons, you can add or remove attributes, move attributes up and down.

3. Configure the Type, Constant, and Sample Value columns for the added fields.

   • Type

     The data type in the corresponding field in the file is converted to before publication.

     When you use the dateTime type, you must specify the pattern and locale: right-click the field name and select **Set locale and pattern**, and then select a supported locale and pattern from the list.

     > The parser that matches the pattern specified with the actual date and time is not a strict parser. Even if the pattern and the actual date and time do not match, you might not receive an error. To ensure that the pattern used is correct, you have to verify the date and time actually published with the actual date and time in the data file.

   • Start

     The starting position for the field in the record.

   • Length

     The length of the field.

   • Constant

     Available only when you have selected **Field Value** from the **Identify Type** field. Enter a constant value for the appropriate field.

   • Sample Value

     The value is set to `<unknown>` if the Display Wizard panel is not extended. It shows the actual value when the Display Wizard panel is used and the line in the file is highlighted. You can remove or shuffle the fields.

4. Create fields and the names of the fields using the Display wizard.

   The wizard text box is used to display the contents of the file for the record being created. You can specify the start and length of each field.

   a) Select the encoding of the file content.
      Before opening the file, you have to select the file content.
   b) Click **Browse** to navigate to the file.
   c) After identifying the desired file, click **Open**.
      The file content is displayed in the Display Wizard panel.
   d) Highlight the line where you want to create a file record.
      The file record lines are displayed in the Sample Line box.
   e) Click **Add** to add a field.
   f) Select the text in the sample line, the start and the length for the fields are calculated automatically.
   g) Repeat Step e and Step f to create other fields.

   After creating all fields for the record, if you have selected **Record Length** from the **Identifier Type** list, you can update the length configuration parameter using the **Update Length** button.

Read Schema (Positional) Reference

The positional read schema has some attributes, you can check the following table for the attributes and the corresponding description:

Attribute Name	Description
Name	The file record names have alphanumeric characters only and have 80 characters at most. Each name is unique within the adapter configuration.
Description	(Optional) A short description for the positional file record.
Position From Header	This field is available only for records that are defined as immediate child record of the root record. Select **Begin** or **End** based on the structure of the actual records in the data file.
Strip Blanks	Select this check box to remove the blank spaces for each field identified for the record in data file.
Repeating Delimiter	Specifies whether to ignore repeated delimiters. Select this check box to translate any repeated delimiters between fields as only one field separator. Clear this check box to translate the repeated delimiters as empty fields.
Identifier Type	Specifies the method to identify a record. **Field Value** (constant field value) and **Record Length** are available options. When the **Identifier Type** field is set to **Field Value**, the constant attribute for one of the fields is specified when defining the fields for the records. When the **Identifier Type** field is set to **Field Value**, Publication Service can also be configured to check the field count in addition to the constant field value in validating a delimited record. In addition to checking only the constant field value in validating a record, Publication Service can also be configured to check the record length. For example, a read schema defines a record to contain three String fields with the first field containing a constant string "Order". A record, such as "Order,ID1234,Aug20" is accepted by Publication Service. However, a record with a valid constant Order but contains four fields, such as "Order,ID1234,Aug20,CA" is rejected. The following error is logged: 1 `lines couldn't be interpreted`. To use this feature, you have to set the `adfiles.matchFieldCountAndRecordLength` property to `ON` in the `adfilesagent.tra` file (in the `TIB_ADFILES_HOME\bin` directory). 📝 Only one constant field value can be used to identify a record.
Length	This field is available only if you have selected **Record Length** in the **Identifier Type** field.

Defining Write Schemas

Two types of write schemas are available: WriteSchema(Classic) and WriteSchema(Manual), you can define them using Business Object Editor.

Defining or generating a write schema is done at adapter configuration level. Therefore, the generated or created write schemas can be shared across Subscription Service services of that adapter configuration.

Copying and pasting write schemas is not supported.

To define a write schema, complete the following tasks:

Procedure

1. Create a write schema
   See Creating a Write Schema (Classic) or Creating a Write Schema (Manual).
2. Defining Relationship
3. Configuring a Write Schema

Creating a Write Schema (Classic)

Before configuring a write schema (classic), you have to create a write schema (classic) first.

Before clicking **Add Write Record (Classic)**, you have to associate a read schema to a corresponding Publication Service. Otherwise, no nodes are available in the Select Schema Model window after clicking **Add Write Record (Classic)**.

Procedure

1. In the Project panel, double-click an adapter configuration that you want to add a write schema.
2. In the Adapter for Files Configuration panel, click the **Business Object Editor** link in the Getting Started panel.
3. In the File Schemas panel, click **Add Write Record (Classic)**.
4. In the Select Schema Model window, select appropriate node in the Available Nodes panel. Click **OK**.

   The selected record and its child records are added to the File Schemas panel of the Business Object Editor.
5. Click **Save All**.
6. Optionally provide a new name and description for the write schema.
   The **Write Schema Type** field is displayed with read-only value `Classic`.

Creating a Write Schema (Manual)

If you want to create a write schema manually, you can click **Add Writer Record (Manual)**.

A WriteSchema(Manual) has only one root record. A root record can have many child records, and each child record can have many child records.

When the configuration of a write schema is changed, you can click the **Refresh Business Object** link (in the **Adapter Services** tab) to refresh the write schema and its relationship with Subscription Service; however, the root record is not changed.

To create WriteSchema(Manual), complete the following steps:

Procedure

1. Create a root write record.
   a) In the Project panel, double-click an adapter configuration that you want to add a classic write schema.
   b) In the Adapter for Files Configuration panel, click the **Business Object Editor** link in the Getting Started panel.
   c) In the File Schemas panel, click **Add Write Record (Manual)**.
      The **Write Schema Type** field is displayed with read-only value `Manual`.
   d) Configure the root file record.

      For more information about configuration, see Configuring a Write Schema.

2. Create a child file record for the root file record.
   a) In the File Schemas panel, click **Add Child Write Record (Manual)**.
      The **Write Schema Type** field is displayed with read-only value `Manual`.
   b) Configure the child write record.

      For more information about configuration, see Configuring a Write Schema.
   c) Repeat Step a and Step b to create more child file records for the root file record.

Defining Relationship

The write schema reflects the relationship of the file records. The relationship in the write schema determines the sequence of lines that are generated in the output file.

To view the write schema, click *Project_name* > **BO Models**, and you can expand the write schema that you created. The file records and the corresponding relationship defined in the wire schema or the canonical schema, or Write schema (Manual) is displayed.

The first file record generated or created for the write schema is referred to as root record. All the other records created after the first file record are referred to as child records of the root record.

A root record can have many child records, and each child record can further have child records. The child file records defined immediately for the root record have a `Position From Header` attribute. With this attribute, the immediate children of the root record in the actual data file are displayed before or after the root record.

The `Position From Header` attribute is only available for child records defined under a root record.

For example, if a write schema is defined as follows:
```
Order
        Customer
        Item
```

If you select **Begin** from the **Position From Header** list for the Customer file record and Item file record, the contents of the generated data file is displayed as follows:
```
Order
Customer
Item
Item
Order
Customer
Item
```

If you select **End** from the **Position From Header** list for the Customer file record and Item file record, the contents of the generated data file is displayed as follows:
```
Customer
Item
Item
Order
Customer
```

```
Item
Order
```

The **`Position from Header`** attribute setting for the child records is either **Begin** or **End**, and it cannot be in combination. For example, you cannot set **Begin** for the Customer file record and **End** for the Item file record.

Configuring a Write Schema

You can configure write schemas using Business Object Editor.

Procedure

1. Configure attributes in the Write Business Object.

   For detailed information about these attributes, see Write Schema Configuration Reference.

2. Edit attributes.

   Attributes are editable. You can change the fields of the attributes or change the number of attributes.

   > If you want to make significant changes to WriteSchema(Classic), you can regenerate the write schema rather than edit it.

   - To edit an attribute, click the attribute.
   - To add an attribute, click **Add**.
   - To remove a field, click **Remove**.
   - To shuffle fields, click **Up** or **Down**.
   - To set default value for an attribute, right-click the attribute.

3. Modify the relationship of file records in a write schema.

   The write schema reflects the relationship of the file records. The relationship in the write schema determines the sequence of lines that are generated in the output file.

   For example, if the write schema is defined as follows:
   ```
   WriteSchema
       FileRecord_Parent
           FileRecord_Child_1
               FileRecord_Child_2
           ...
           ...
   ```

   The sequence of the lines in the output file is:
   ```
   FileRecord_Parent
       FileRecord_Child_1
       FileRecord_Child_1
       ...
           FileRecord_Child_2
       ...
       ...
   ```

   You can rearrange the order of the child records of a write schema. Right-click the child record you want to change, select **Move Up** or **Move Down** in the pop-up menu to change the order.

   Once you change the order of the child records of a write schema, the child records are written to a file in the exact same order as you have configured for the write schema at design time.

> If you have set the `adfiles.toggleChildRecordsOrdering` property in the TRA file, and then rearrange the order of the child records in TIBCO Business Studio, the property setting has no effect.
>
> If you have enabled the `adfiles.toggleChildRecordsOrdering` property for projects created in earlier versions of 6.2, you can also run the projects successfully in this release.
>
> When you have to write child records as part of a parent record, you can promote a child record. A promoted child becomes a field of type record in the parent record.
>
> Conversely, you can demote a promoted child record.

Write Schema Configuration Reference

A write schema has some attributes, you can check the following table for the attributes and corresponding description.

Attributes in the Writer Delimited Business Object

Attribute Name	Description
Name	Displays the name of the file record as given in the wire schema by default. You can modify it when necessary. File record names have alphanumeric characters only and 80 characters at most. Each name is unique within the adapter.
Position From Header	This field is available only for records that are defined as immediate child record of the root record. Select **Begin** or **End** based on the structure of the actual records in the data file.
Wire Schema Path	The wire schema that is used to create this write schema.

Parse	Specifies whether the generated output line uses a delimiter to separate fields or whether it uses absolute line positions. **Delimited** and **Positional** are the available choices.
	
	• **Delimited**
	
	  Select the choices specified in the Delimiter configuration item to separate fields within each generated output line.
	
	• **Positional**
	
	  Use the line position to separate fields. That is, each field begins at a fixed position (offset from the start of the line).

Delimiter	Specifies the separator between fields within each line of a file for the line that is processed. **Space**, **Tab** and **Other** are available choices. If you select **Other**, the **Delimiter Other** field becomes available.
Delimiter Other	This field is available only if you select **Other** in the **Delimiter** list. Specifies the delimiter to be used. You can specify single or multiple characters as a delimiter.
Write Schema Type	**Classic** and **Manual** are the two available types. A value is set when you create the write schema.

Attributes for Write Schema

Field	Description
Name	The attribute name of the object from the incoming message. The value of this field is populated automatically when generating the WireSchema for WriteSchema(Classic).
Type	The data type of the attribute. The value of this field is populated automatically when generating WireSchema for WriteSchema(Classic).
	If the dateTime type is used, you must specify the pattern and locale: right-click the field name and select **Set locale and pattern**, and then select a supported locale and pattern from the list.
	The record type is not displayed for WriteSchema(Classic) after generating the write schema from the wire schema.
Width	The number of characters used to represent the field. `Zero (0)` signifies that the field is not included in the output line. `Negative one (-1)` signifies that as many characters as needed (without padding) are used (this is the default value for lines that use delimiters. Lines using the Positional method cannot specify `-1`).
Alignment	Specifies whether the data is aligned to the left or the right.
Padding	The character added to this field pads the field to the specified width.
Sign	For numeric fields only, used to add a prefix to the positive sign before positive numbers in the output. If you want to display the sign in the output, set it to `True`; otherwise, set it to `False`.
Precision	Specifies the number of digits after the decimal place. For example, with a precision of `3`, the number `3.14159` is displayed as follows: • Scientific notation: `3.142e+1` • Non-scientific notation: `3.142` A value of `-1` indicates that the entire precision provided by the operating system is used.
Sc. Notation	For real numbers only, used to determine whether scientific notation is used for the output of this field. To use the scientific notation, set it to `True`.
Upper Case	Valid only when the Sc. Notation is set to `True`. When this field is set to `True`, the `E` in the scientific notation is in uppercase. When this field is set to `False`, the `e` in the scientific notation is in lowercase.

Working with a Process

TIBCO ActiveMatrix BusinessWorks processes capture and manage the flow of business information in an enterprise between different data sources and destinations. The Adapter palette installed with TIBCO ActiveMatrix Adapter Framework provides activities for wiring adapter services into processes. To run a process, you have to map and transform the input data of each activity.

With processes, you can implement business logic that can obtain and manage the flow of information in an enterprise between a source and different destinations. TIBCO Business Studio workbench provides a design environment to develop and test a process.

Working with a business process consists of many procedures. Supposing you are an experienced user and familiar with TIBCO ActiveMatrix BusinessWorks, the procedures give a brief overview of the workflow.

For detailed information about developing a process, see "Developing a Basic Process" in *TIBCO ActiveMatrix BusinessWorks Application Development*.

For configuration on each activity in the Adapter Palette panel, see *TIBCO ActiveMatrix Adapter Framework Reference*.

For details on how the adapter interacts with a TIBCO ActiveMatrix BusinessWorks business process, see *TIBCO ActiveMatrix Adapter for Files for Unix/Win (TIBCO Business Studio) Examples*.

A typical workflow of developing and testing a process consists of the following tasks:

1. Creating a project
2. Creating a process
3. Configuring a process
4. Adding activities from the Adapter Palette panel
5. Configuring activities
6. Adding and Configuring transitions between activities
7. Mapping and transforming input data for each activity

   For mapping operations on activities, see *TIBCO ActiveMatrix BusinessWorks Bindings and Palettes Reference*.

8. Validating a process
9. Testing a process

Migrating a Project

You can migrate an adapter project created in TIBCO ActiveMatrix BusinessWorks 5.x to TIBCO ActiveMatrix BusinessWorks 6.x by using the migration tool in TIBCO Business Studio.

You can migrate a TIBCO ActiveMatrix BusinessWorks 5.x project to a TIBCO ActiveMatrix BusinessWorks 6.x project, but not in reverse. TIBCO Business Studio does not support importing `.dat` files from TIBCO Designer. To migrate a project stored in a `.dat` file, you can import the file in TIBCO Designer and save it as a multi-file project.

The name of a project migrated to TIBCO Business Studio cannot contain any special characters such as: (space) ! $ % & + . / @ \ ~

Prerequisites

Before migrating a project to TIBCO ActiveMatrix BusinessWorks 6, complete the following steps:

1. Migrating the project to the latest version of the runtime adapter.
2. Validating the project in TIBCO Designer.

   For more information, see "Validating Projects" in *TIBCO Designer User's Guide*.

   Before preparing a project for migration, ensure that no error occurs when validating it in TIBCO Designer.

Procedure

1. In TIBCO Business Studio, click **Project** > **Migrate BW Projects** to open the migration tool.

   Alternatively, you can migrate the projects by clicking **File** > **Import** from the main menu. In the Import dialog, click **Migrate BW Projects**, and then click **Next**.

2. In the BW Migration Tool wizard, specify the relevant information in the following fields.
   a) In the **BusinessWorks 5 Projects Folder** field, click **Browse** and provide the location of the ActiveMatrix BusinessWorks 5.x project that you want to migrate.

      The tool scans the ActiveMatrix BusinessWorks 5.x projects in the specified folder location and displays a list of projects.
   b) In the **Migrated Project Folder** field, click **Browse** and provide the location where you want to migrate the project.

3. In the BusinessWorks 5 Projects panel, select the projects you want to migrate and click **Migrate**. The migration progress is displayed in the Migration Output panel.

4. Click **Close** after the migration is completed.

What to do next

Depending on the configuration of the project to migrate, you have to complete some tasks after migration. For the postmigration tasks, see Configuring a Migrated Project.

Configuring a Migrated Project

If a migrated project uses internationalized data or the Start activity, you have to configure the migrated project after migration.
If the migrated project uses internationalized data, make sure the text file encoding is UTF-8 before processing UTF-8 messages. If the migrated project has a process that uses the Start activity, you have to manually create a new ActiveMatrix BusinessWorks process to call the migrated process.

Procedure

1. Create a new process in TIBCO Business Studio.

2. In the new process, add a Timer, File Poller or any activity of the process starter type.

3. In the new process, add a Call Process activity.

4. Add a transition between the process starter and the Call Process activity.

5. Select the Call Process activity and configure the activity.
   a) In the Properties view, click the **General** tab.
   b) Click **Add** next to the **Service** field.
   c) In the Select a Service dialog, select the migrated process.
   d) Click **OK**.

   The following figure shows the configuration of a sample Call Process activity.
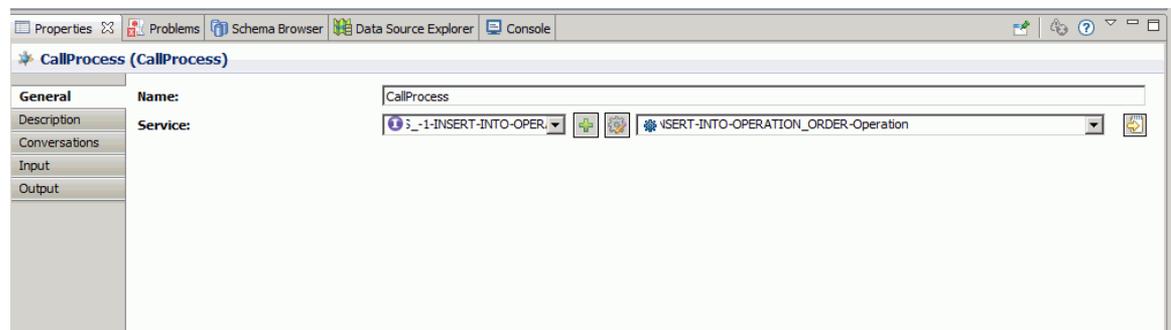
6. Select the reference added to the new process and configure the reference.
   a) In the Properties view, click the **General** tab.
   b) In the Wire to Process panel, click ▣ to select a corresponding process from the **Process** list, and select a service from the **Service** list.
   c) Click the blank area in the Process editor.
   d) Click the **Advanced** tab, and select **Single AppNode** from the **Activation** list.

7. Save the project.

   After finishing these steps, call the created process to invoke the migrated process.

Advanced Topics

When you use TIBCO ActiveMatrix Adapter for Files for Unix/Win (TIBCO Business Studio), you can use advanced configuration and deployment.

JMS Messages Compression

JMS message compression is useful when messages are to be stored on the TIBCO Enterprise Message Service server. The messages include persistent queue messages and topics with durable subscribers.

When compression is enabled, messages take less memory space in storage and are handled faster by the TIBCO Enterprise Message Service server. When JMS messages are compressed and stored, they are handled by the server in the compressed form.

The compression option only compresses the body of a message. Headers and properties are not compressed. When the message bodies are large and the messages are to be stored on a server, you can compress messages.

When messages are not to be stored, compression is not as useful. Compression takes time, and therefore the time to send or publish and receive compressed messages is generally longer than the time to send the same messages uncompressed. You do not have to compress small messages that are not to be stored by the server.

You can enable or disable this feature for message senders with the following adapter property:

`adapter property for JMS compressing`: the default value of the property is set to `OFF`, which indicates that messages are not compressed. If you want to enable the feature, set the property to `ON`.

Error Handling

Error Handling provides a convenient way of analyzing the parsing errors. It is used in both Record Transfer mode and Simple File Transfer mode.

Error Handling in Record Transfer Mode

Error Handling provides a convenient way of analyzing the parsing errors in Record Transfer mode.

When IO errors occur, Publication Service exits; Subscription Service does not confirm messages and exits by default. When you set the `adfiles.io.sub.exit4issue` property to `OFF`, Subscription Service confirms messages and continues to process other messages regardless of the IO errors.

Publication Service

When Publication Service detects a parsing error that violates the predefined schema, an XML file containing the name of the input file and the error details is created. To use this feature, you must specify a valid directory in the **Error Directory** field in the **Processing** tab.

For example: If a read schema is defined as follows:
```
H
  D
```

where `H` and `D` are delimited file records (identified by a constant) with the following fields and types:
```
H
field_1 String (Header)
field_2 ui4
D
field_1 String (Detail)
field_2 ui4
```

For example, if Publication Service parses an `input.txt` input file, which contains the following records:

```
Header, -10
Detail, 20
Header, 30
Detail, 40
Header, 50
Detail, 60
```

The line `Header -10` fails in validation. The XML file created is named `input.txt20030917151048000.xml`.

```
<?xml version="1.0" encoding="UTF-8" ?>
- <file name="input.txt20030917151048000">
- <errorObjects>
- <object>
<contents>Header, -10 </contents>
<location startLine="1" endLine="1" startOffset="0"
endOffset="11" />
<error>Header record contains invalid data field: Header, -10
[-10]</error>
</object>
</errorObjects>
```

The XML file contains the name of the input file and an object element for each error. Each object element contains details such as the entire identified object, location of the object in the file, and error details. The input file that has parsing errors is kept for analysis. The location and name depend on the postprocessing option specified in the **Processing** tab. If the input file is `abc.txt`, the following list describes the name and location based on various postprocessing options that are available:

• **Leave as is**

  – `abc.txt` is retained in the input directory

  – `abc.txt.timestamp` is created in the error directory

  – `abc.txt.timestamp.xml` is created in the error directory

• **Delete**

  – `abc.txt` is deleted from the input directory

  – `abc.txt.timestamp` is created in the error directory

  – `abc.txt.timestamp.xml` is created in the error directory

• **Move to (without timestamp)**

  – `abc.txt` is moved to the done directory

  – `abc.txt.timestamp.xml` is created in the error directory

  – no copy of the input file is created in the error directory

For easy identification, the timestamp on the XML file and the corresponding input file is the same.

Subscription Service

If an error occurs when processing a file transfer, one of the following scenarios might occur to Subscription Service:

• If an error occurs when Subscription Service is writing a file in the wip directory, Subscription Service does not confirm the message, it prints an error log and exits by default. When you set the `adfiles.io.sub.exit4issue` property to `OFF`, Subscription Service prints an error log, confirms the message, and then continues to receive other messages.

- If an error occurs when Subscription Service is moving the transfer file from the working directory to the output directory, and the error directory is specified, the adapter moves the file to the error directory and Subscription Service exits by default. When you set the `adfiles.io.sub.exit4issue` property to `OFF`, Subscription Service does not exit.

- When IO errors occur in Subscription Service, Subscription Service exits by default. If Publication Service continues to send messages using the Durable EMS transport or the Certified RV transport, Subscription Service resubscribes the remaining messages from the transport when restarted.

To save messages on the Certified RV transport, you have to select **Repeating** in the **Life Cycle** field in Publication Service.

When an IO error occurs, whether the data format of a message is **Business document (MBusinessDocument)** or **Object (MInstance)**, Subscription Service handles the IO error in the same way.

Diagnostics File

If the invalid records violate the schema and contain invalid characters, all the invalid records and their line numbers are created in a file in the specified error directory. The diagnostics file provides useful information for users to correct the invalid records and then republish them.

The following file is an example of an input file containing invalid records (marked in bold):

```
OrderX, ID41678, <10Apr2000
Item, GigaWidget, 60, $75
Item, MegaBucket, 48, $60
Customer, Hopkins Associates, ID36800
Order, ID41680, 20May2000
ItemX, Rt.Clopper, >40, $50
Item, Lt.Clopper, 50, $100
Customer, JerseyWebInovaters, ID46786
```

The schema diagnostics file has the format of an XML file and has a `.xml` extension, so it can be opened and viewed in a text editor as well as in an Internet browser. However, when the invalid records contain special XML characters, a diagnostics file might not be deemed as a well-formed XML file and cannot be opened in an Internet browser.

The XML specification states that characters '&', '<', and '>' are special characters, and these special characters must be replaced by strings "&", "<", and ">" respectively when they are displayed in a certain part of an XML document.

When invalid records containing special XML characters are written to the diagnostics file without proper conversion, the diagnostics file is not a well-formed XML file. However, the adapter provides three different formats for users to choose from for the diagnostics file:

- Plain text file
- XML file without special character conversion
- XML file with special character conversion

If you use TIBCO Administrator to deploy and run the adapter, you can create an `adfiles.schemaDiagnosticsFileFormat` module property and use it to specify the desired format for the diagnostics file. The three valid values for the module property are `xml` (default), `xmlconv`, and `text` for the three different available formats for the diagnostics file "XML without special character conversion", "XML with special character conversion", and "plain text".

For users who run the adapter by configuring or using the `.tra` file directly, they can add the `adfiles.schemaDiagnosticsFileFormat` property to the `.tra` file and specify the desired format for the diagnostics file. The three valid values for the module property are `xml` (default), `xmlconv`, and `text` for the three different available formats for the diagnostics file "XML without special character conversion", "XML with special character conversion", and "plain text".

Error Handling in Simple File Transfer Mode

Error Handling provides a convenient way of analyzing the parsing errors in Simple File Transfer mode.

Publication Service

For every Subscription Service that is marked unavailable, or if an available Subscription Service returns a checksum error for a given file transfer, Publication Service maintains a specific log file (`_.txt`) for Subscription Service in the error directory specified during configuration. This file contains information that points to the file that Subscription Service had an issue with.

Based on the postprocessing option, files are moved to an appropriate place for future use.

- If the postprocessing option is set to **Leave as is** or **Delete**, and Publication Service has detected an issue with Subscription Service, the file is moved to the error directory with the date and timestamp appended to it.

- If the postprocessing option is set to **Move to**, and Publication Service has detected an issue with Subscription Service, the file is moved to the completion directory. Depending on the option to add date and time, the date and timestamp are appended to the file.

Subscription Service

If an error occurs when processing a file transfer, Subscription Service prints an error log and exits by default. When you set the `adfiles.io.sub.exit4issue` property to `OFF`, Subscription Service prints an error log and does not exit.

When IO errors occur in Subscription Service, Subscription Service exits by default. If Publication Service continues to send messages using the Durable EMS transport or the Certified RV transport, Subscription Service resubscribes the remaining messages from the transport when restarted.

To save messages on the Certified RV transport, you have to select **Repeating** in the **Life Cycle** field in Publication Service.

File Recognition Methods for Publication Service

Publication Service has various configuration setting combinations, you can check the listed configuration setting combinations and the results. Each of the first four columns indicates a field value. An empty column indicates no value is specified in the corresponding field.

Recognition Method	File Prefix	File Extension	Trigger File Extension	Result
By prefix + extension				All files in the input directory are processed. This option is similar to using the `*.*` command.
	delimited			Only files in the input directory that contain delimited in the file prefix are processed. This option is similar to using a **`delimited.*`** command. For example, `delimited.txt, delimited123.txt, delimited456.csv`.

Recognition Method	File Prefix	File Extension	Trigger File Extension	Result
	delimited	txt		Only files in the input directory that contain `delimited` in the file prefix and have a `.txt` extension are processed.
		txt		Only files in the input directory with `.txt` file extensions are processed. This option is similar to using a `*.txt` command. For example, `abc.txt`, `123.txt`.
By Trigger			trg	This is equivalent to specifying that when a *prefix*.`trg` trigger file is created in the input directory, the adapter searches for a file named *prefix*, since the file extension is empty. For example, if a file named `delimited.trg` is created in the input directory, the adapter only parses files for a file named `delimited`. If a file named `payments.trg` is created, the adapter searches for a file named `payments`.
	delimited		trg	When a `delimited.trg` trigger file is created, the adapter searches for a file named `delimited`. Similarly, if a `delimited2.trg` file is created, the adapter searches for a file named `delimited2`. In this example, a file named `payments.trg` is ignored by the adapter.
	delimited	txt	trg	When a `delimited.trg` trigger file is created, the adapter searches for a file named `delimited.txt`. If a `delimited2.txt` file is present, it is not processed. The file is processed when `delimited2.trg` is created. In general, when the prefix name of the trigger file matches the prefix name of the file extension, the adapter processes the file.
		txt	trg	When a *prefix*.`trg` trigger file is created, the adapter searches for a file named *prefix*.`txt`. Similarly, when a `payments.trg` trigger file is created, the adapter searches for a file named `payments.txt`.

Data Validation

The runtime adapter performs data validation. You can check the following table for the valid format and range of each data type. When a field contains data that does not conform to the format or data whose value is beyond the valid range, the entire record is discarded and an error message is logged.

Data Type	Format	Valid Range
i1	[whitespace][{+|-}][digits]	<-128,127>
i2	[whitespace][{+|-}][digits]	<-32768,32767>
i4	[whitespace][{+|-}][digits]	<-2147483648,2147483647>
i8	[whitespace][{+|-}][digits]	<-9223372036854775808,9223372036854775807>
ui1	[whitespace][+][digits]	<0,127>
ui2	[whitespace][+][digits]	<0,32767>
ui4	[whitespace][+][digits]	<0,2147483647>
ui8	[whitespace][+][digits]	<0,9223372036854775807>
r4	[whitespace][sign][digits][. digits] [{e|E}[sign]digits]	<1.175494351e-38,3.402823466e+38>
r8	[whitespace][sign][digits][. digits] [{e|E}[sign]digits]	<2.2250738585072014e-308,1.7976931348623158e+308>
DataTime	A list of predefined patterns and an option to specify user-defined patterns are available. If the user-defined pattern option is enabled, you can use User-Defined Pattern to specify the pattern. For syntax and semantics for specifying the pattern, see Syntax for Specifying User-defined Date and Time Patterns.	

Defining Base Record

A base record is a record with fields that can only be scalar type. You can define base records.

Procedure

1. Select the **Is Base Record** check box in the Read Delimited Business Object panel. Once you set a record as a base record, you cannot restore it.

2. Configure the DelimitedFileRecord. If you want more than one base record, repeat Step 1.

Container Record

A container record is a record with fields that can be scalar type or record type.

To better explain the container record, an example is used.
```
Department, Engineering, Employee, ID0005, Mary, Employee, ID0006,
David, Employee, ID0008, John
```

In this example, `Employee` is a base record that contains fields of String type (`Employee`, `ID00005`, `Mary`). The `Department` record contains fields of string type and records. Fields of string type are `Department` and `Engineering`. Fields of record type are the `Employee` record.

The base and the container records are delimited with commas, and are identified using constant field values `Employee` and `Department`.

A container record can contain more than one type of base record.

Syntax for Specifying User-defined Date and Time Patterns

The date and time symbols are used as patterns to interpret and generate date and time in various formats.

Symbol	Meaning	Presentation	Example
G	era designator	(Text)	AD
y	year	(Number)	1996
Y	year/week of year	(Number)	1996
M	month in year	(Text & Number)	July & 07
d	day in month	(Number)	10
h	hour in am/pm (1~12)	(Number)	12
H	hour in day (0~23)	(Number)	0
m	minute in hour	(Number)	30
s	second in minute	(Number)	55
S	millisecond	(Number)	978
E	day of week	(Text)	Tuesday
e	day of week/local (1~7)	(Number)	2
D	day of year	(Number)	189
F	day of week in month	(Number)	2 (2nd Wed in July)
w	week in year	(Number)	27
W	week in month	(Number)	2

Symbol	Meaning	Presentation	Example
a	am/pm marker	(Text)	PM
k	hour in day (1~24)	(Number)	24
K	hour in am/pm (0~11)	(Number)	0
z	time zone	(Text)	Pacific Standard Time
'	escape for text		
''	single quote	'	

Locales Supported for Date and Time

The country and language codes supported by the adapter are listed with corresponding abbreviations.

Abbreviation	Country/Language
ar_AE	Arabic United Arab Emirates
ar_BH	Arabic Bahrain
ar_DZ	Arabic Algeria
ar_EG	Arabic Egypt
ar_IQ	Arabic Iraq
ar_JO	Arabic Jordan
ar_KW	Arabic Kuwait
ar_LB	Arabic Lebanon
ar_LY	Arabic Libya
ar_MA	Arabic Morocco
ar_OM	Arabic Oman
ar_QA	Arabic Qatar
ar_SA	Arabic Saudi Arabia
ar_SD	Arabic Sudan
ar_SY	Arabic Syria
ar_TN	Arabic Tunisia
ar_YE	Arabic Yemen

Abbreviation	Country/Language
be_BY	Byelorussian Byelorussia
bg_BG	Bulgarian Bulgaria
ca_ES	Catalan Spain
ca_ES_EURO	Catalan Spain
cs_CZ	Czech Republic
da_DK	Danish Denmark
de_AT	German Austria
de_AT_EURO	German Austria
de_CH	German Switzerland
de_DE	German Germany
de_DE_EURO	German Germany
de_LU	German Luxembourg
de_LU_EURO	German Luxembourg
el_GR	Greek Greece
en_AU	English Australia
en_BE	English Belgium
en_CA	English Canada
en_GB	English United Kingdom
en_IE	English Ireland
en_IE_EURO	English Ireland
en_NZ	English New Zealand
en_US	English United States
en_ZA	English South Africa
es_AR	Spanish Argentina
es_BO	Spanish Bolivia
es_CL	Spanish Chile

Abbreviation	Country/Language
es_CO	Spanish Colombia
es_CR	Spanish Costa Rica
es_DO	Spanish Dominican Republic
es_EC	Spanish Ecuador
es_ES	Spanish Spain
es_ES_EURO	Spanish Spain
es_GT	Spanish Guatemala
es_HN	Spanish Honduras
es_MX	Spanish Mexico
es_NI	Spanish Nicaragua
es_PA	Spanish Panama
es_PE	Spanish Peru
es_PR	Spanish Puerto Rico
es_PY	Spanish Paraguay
es_SV	Spanish El Salvador
es_UY	Spanish Uruguay
es_VE	Spanish Venezuela
et_EE	Estonian Estonia
fi_FI	Finnish Finland
fi_FI_EURO	Finnish Finland
fr_BE	French Belgium
fr_BE_EURO	French Belgium
fr_CA	French Canada
fr_CH	French Switzerland
fr_FR	French France
fr_FR_EURO	French France

Abbreviation	Country/Language
fr_LU	French Luxembourg
fr_LU_EURO	French Luxembourg
hr_HR	Croatian Croatia
hu_HU	Hungarian Hungary
is_IS	Icelandic Iceland
it_CH	Italian Switzerland
it_IT	Italian Italy
it_IT_EURO	Italian Italy
iw_IL	Hebrew Israel
ja_JP	Japanese Japan
ko_KR	Korean Korea
lt_LT	Lithuanian Lithuania
lv_LV	Latvian Latvia
mk_MK	Macedonian Macedonia
nl_BE	Dutch Belgium
nl_BE_EURO	Dutch Belgium
nl_NL	Dutch Netherlands
nl_NL_EURO	Dutch Netherlands
no_NO	Norwegian (Nynorsk) Norway
no_NO_NY	Norwegian (Bokmål) Norway
pl_PL	Polish Poland
pt_BR	Portuguese Brazil
pt_PT	Portuguese Portugal
pt_PT_EURO	Portuguese Portugal
ro_RO	Romanian Romania
ru_RU	Russian Russia

Abbreviation	Country/Language
sh_YU	Serbo-Croatian Yugoslavia
sk_SK	Slovakian Slovakia
sl_SI	Slovenian Slovenia
sq_AL	Albanian Albania
sr_YU	Serbian (Cyrillic) Yugoslavia
sv_SE	Swedish Sweden
th_TH	Thai Thailand
tr_TR	Turkish Turkey
uk_UA	Ukranian Ukraine
vi_VN	Vietnamese Vietnam
zh_CN	Chinese (Simplified) China
zh_HK	Chinese Hong Kong
zh_TW	Chinese (Traditional) Taiwan

Trigger Messages

A Rendezvous or JMS trigger message can contain the `USERID`, `inputDirectory`, `fileName`, `filePrefix`, and `fileExtension` properties. When you use a message that has those properties to trigger Publication Service, the JMS trigger message properties override corresponding properties that you set at design time. The changes made at run time do not change the configuration in the repository.

If you select **By File Name** in the **Recognition Method** list, the `fileName` property only contains the file name, and does not contain ICU regular expressions.

When an ICU regular expression is used in trigger messages, if the ICU regular expression contains a comma, the comma is skipped.

For example, a trigger message `fileName=go{1,5}gle\.txt` contains a comma. The comma is not a property separator, so an "escaping" comma in front of the comma is added: `fileName=go{1,,5}gle \.txt`.

You can send multiple trigger messages at one time, and each message has properties. After Publication Service receives trigger messages, the messages queue. Publication Service processes them one by one.

When you send multiple trigger messages, ensure that the number of the messages does not exceed reasonable amount that your system resource supports.

TIBCO ActiveMatrix Adapter for Files for Unix/Win makes the best attempt to keep the trigger messages but does not guarantee the success.

Syntax

Follow these rules when using Rendezvous or JMS trigger messages:

- Only the String property is supported.

- The properties must be enclosed in double quotation marks.

- The properties are separated by commas. No spaces are inserted between properties.

  For example, `"inputDirectory=reader/input,fileName=foo.txt"`.

- The property value can be String and Integer. It cannot contain commas. Other date types are not supported.

Scenarios

Rendezvous and JMS trigger messages are used in the following scenarios:

- If the trigger message is empty, Publication Service uses the properties defined at design time.

- The input directory of Publication Service can be changed at run time by sending the following trigger message.

  - Rendezvous message:

    ```
    {
    RVMSG_STRING 6 inputDirectory "/home"
    }
    ```

  - JMS message:

    ```
    "inputDirectory=reader/input"
    ```

    The input directory is changed. The change remains in effect until another trigger message with the above format is received.

- If you select **By file name** in the **Recognition Method** list for Publication Service, it can be changed at run time by sending the following message:

  - Rendezvous message:

    ```
    {
    RVMSG_STRING 8 fileName "abc.txt"
    }
    {
    RVMSG_STRING 8 DATA "abc.txt"
    }
    ```

    In this example, `8` is the length of the `abc.txt` filename plus one.

  - JMS message:

    ```
    "fileName=abc.txt"
    ```

    > If a JMS trigger message contains no file name property, Publication Service uses the file name specified in the last JMS trigger message, or the file name configured in TIBCO Business Studio if no file name is specified in previous trigger messages. `"The trigger message does not mention the input filename therefore use the input filename currently in-effect File_name."` is logged.

- If you select **By prefix + extension** in the **Recognition Method** list for Publication Service, it can be changed at run time by sending the following message:

  - Rendezvous message:

    ```
    {
    RVMSG_STRING 4 filePrefix "abc"
    RVMSG_STRING 4 fileExtension "txt"
    }
    Additionally the following special fields are also recognized
    {
    RVMSG_STRING 1 USERID ""
    ```

```
}
{
RVMSG_STRING 1 TRACKINGID ""
}
```

– JMS message:

```
"filePrefix=abc, fileExtension=txt"
```

- When Publication Service gets an invalid trigger message, for example, "fileName=foo.txt,=John" or "reader/input,fileName=foo.txt", Publication Service discards the message.

- Publication Service can receive more than one trigger message at one time but processes them one by one. Trigger messages waiting to be processed are saved. If Publication Service stops and then restarts, it continues to process the saved trigger messages.

Selective Routing Over JMS

When using JMS trigger messages to trigger Publication Service, you can send different files to the selected subscribers dynamically using Selective Routing.

Syntax

- The trigger message must be a JMS text message. The text must be in the "Prop=Val" format.

- Only string and numeric message selector properties are supported.

- Multiple message selector properties can be entered, and they must be separated by a comma. The value of properties must be enclosed in double-quotes. For example: "Prop1=Val1,Prop2=Val2".

> The USERID, inputDirectory, fileName, filePrefix, and fileExtension properties are reserved for TIBCO ActiveMatrix Adapter for Files for Unix/Win (TIBCO Business Studio). The message selector properties that you define must not use the same name. All the properties that you define are included in the properties section of every outgoing JMS data message.

Configuration

To use selective routing, you have to specify the JMS message selector for subscribers that you want to send messages to. You can specify message selectors in TIBCO Business Studio.

Setting JMS Message Selectors

You can set JMS Message Selectors in TIBCO Business Studio.

Procedure

1. Start the EMS server.

2. Open the project containing Subscription Services in TIBCO Business Studio.

3. Specify JMS message selectors.
   a) In the **Transports** tab of Subscription Service, expand the DefaultJMSTopicSession tree, and select a SubscriptionServiceEndPoint.
   b) In the EndPoint Configuration panel, specify the selector in the **Message Selector** field. For example, type (Branch='Boston' OR Branch='East Coast' OR Branch='ALL') AND ((SalesUpper>=62 AND SalesLower<=62) OR Sales volume='ALL').
   c) Repeat Step a and Step b to specify a JMS selector for another Subscription Service. For example, type (Branch='New York' OR Branch='East Coast' OR Branch='ALL') AND ((SalesUpper>=90 AND SalesLower<=90) OR SalesVolume='ALL') in the **Message Selector** field for Subscription Service named mysub2.

4. Select **Simple File Transfer** from the **Transfer Mode** list.

5. Start Subscription Service and connect to the EMS server.

6. Start Publication Service and connect to the EMS server. Publication Service uses Simple File Transfer mode. The polling method is JMS (topic) message.

7. Send a JMS trigger message to Publication Service.
   For example, send the following JMS message: `fileName=File_name1,Branch='East Coast',SalesUpper=70,SalesLower=50.`

Result

After receiving a trigger message, Publication Service starts transferring the files to Subscription Services specified in the JMS message selectors in Step 3. The `fileName`, `Branch`, `SalesUpper`, and `SalesLower` properties are included in the outgoing data messages. In this example, mysub1 matches the conditions in the JMS message. Publication Service routes the messages to mysub1.

See the TIBCO Enterprise Message documentation about configuring EMS servers, creating EMS routes, constructing JMS messages, and specifying JMS message selectors.

Dynamically Changing Output File Names at Run Time

In Record Transfer mode, Subscription Service uses the filenames configured at design time when creating the output files to store incoming messages by default.

When you want Subscription Service to use different filenames dynamically from what is configured at design time when creating the output file, you can use the Business Document data format in the data messages and embed the new filenames in the data messages. One scenario is when the publisher is Publication Service and you want Subscription Service to create the output file using the exact same filenames when files are being published.

Procedure

1. Select **Business Document** from the **Data Format** list in the **Advanced** tab of Publication Service.

2. Type the name of the Business Document in the **Business Document Name** field.

3. Select **Append messages to file, Close on Business Document Lot End** from the **Wip Creation Mode** list in the **Configuration** tab of Subscription Service.

4. Add `adfiles.useBDEmbeddedFileName ON` to the `adfilesagent.tra` file (in the `TIB_ADFILES_HOME\bin` directory), or create a module property named `adfiles.useBDEmbeddedFileName`. The value of the property is `ON`. This is for Subscription Service.

   > If you use TIBCO Administrator to deploy and run the adapter, create an `adfiles.useBDEmbeddedFileName` module property and set the value to `ON`.

Adapter Configuration Encoding

Adapter configuration encoding includes file and directory encodings, and content encodings. To interoperate with other TIBCO ActiveEnterprise applications, you have to set adapter configuration encoding for the repository correctly.

File and Directory Encodings

The adapter provides support to set the encoding of the file and the directories that it processes at service level. This is provided as a configuration parameter for the service.

Based on the platform encoding on which the runtime adapter runs, you have to set parameters appropriately at design time.

Content Encodings

The adapter provides support to process file contents with various encodings. This is provided as a configuration parameter for the service.

The standard that TIBCO ActiveEnterprise uses is LATIN_1 or ISO-8859-1 for exchanging LATIN_1 or ASCII data, and UTF-8 for other encoding such as SHIFT_JIS.

Depending on the file encoding setting for the adapter configuration, the repository instance encoding has to be correctly set. The repository instance encoding determines whether the data is exchanged in LATIN_1 or UTF-8. The default repository encoding for local or the remote repository is LATIN_1 or ISO-8859-1. This conforms to the file encodings LATIN_1 and ASCII.

When processing other encoding such as SHIFT_JIS, the repository instance encoding must be set to UTF-8.

Changing Adapter Configuration Encoding

When changing adapter configuration encoding, you have to consider whether it is remote repository or local repository.

- Remote Repository

  You can set the `repo.encoding` property to `UTF-8` or `Latin-1` in the remote repository initialization file (`tibcoadmin<domain>.tra`).

- Local Repository

  If an adapter configuration is saved in a local project, the intercommunication encoding is determined by the encoding property of the local project file. To communicate with other adapters using the same encoding, all adapters and applications must set their local project file encoding property identically. The encoding value is set on the root project folder in the **Project Settings** tab. Select the appropriate repository instance encoding (ISO-8859-1 or UTF-8). Click **Apply** to save the settings.

Creating Business Event Messages

The adapter provides business-event level notifications for TIBCO Business Events.

Procedure

1. Create a network sink in TIBCO Business Studio:

   1. Drag the network sink object from the Adapter Resources palette.

   2. Assign the EEMEvent role to the network sink.

   For detailed information about creating a network sink, see *TIBCO Designer User Guide*.

2. Configure the `adfiles.EEMEnable` property:

   - If you are using TIBCO Administrator: in the Module Properties panel, add an `adfiles.EEMEnabled` module property and set the value to `ON`.

   - If you are not using TIBCO Administrator, add the `adfiles.EEMEnabled` property to the `adfilesagent.tra` file and set the value to `ON`.

Simple File Transfer Status Message

At the end of each file transfer, Publication Service and Subscription Service send or log a status message based on the selected log roles.

The status message contains the following information:

Name	Description
fileName	The name of the file that is being transferred.
hostName	The name of the host on which Publication Service is running.
instanceName	The name of the adapter configuration.
serviceName	The name of Publication Service.
trackingId	The unique tracking ID generated for that polling adapter configuration every time the polling trigger activates the publisher. You cannot set the tracking ID if you have selected TIBCO Rendezvous message as the polling method, see Trigger Messages.
fileTransferDuration	The time taken to transfer the file.
currentDateTime	The date and time at the end of the file transfer.
status	The status of the file transfer. The value is set to n/a, or not applicable.

You can configure a status message to be sent as a TIBCO Rendezvous status message or logged to the appropriate sinks.

To send a status message as an exclusive TIBCO Rendezvous message on a user-defined subject, create a network sink, create a role named ecmPubStatus for Publication Service or create a role named ecmSubStatus for Subscription Service, and then associate the role with the network sink.

To create user-defined roles such as ecmPubStatus or ecmSubStatus and use network sink, turn on the advanced logging options for the adapter configuration in TIBCO Business Studio. Otherwise, turn on debugRole for the sinks that are currently being used to log a status message.

Load Balancing Mode

For versions earlier than TIBCO ActiveMatrix Adapter for Files for Unix/Win 6.2, only Publication Service in one adapter configuration can be started. From release 6.2, you can run Publication Service in multiple adapter configurations to process separate files in parallel.

To enable Load Balancing mode, you have to set the adfiles.LBEnabled property to ON in the TRA file.

Before Publication Service in an adapter configuration processes an input file, the adapter configuration locks the file by creating a .lck file. Therefore other adapter configurations ignore this input file and find other input files to publish.

- In the **Post Processing** field, you cannot select **Leave as is** for Load Balancing mode. When **Leave as is** is selected, the following error message is displayed after starting Publication Service:

  ```
  Notice: Load Balance not support the post processing mode leave-as-is, please
  correct it
  ```

- Publication Service in each adapter configuration has a unique message subject.

- Load Balancing mode cannot be used together with Delta Publishing mode.

- Load Balancing mode supports the Checkpoint Restart feature.

Delta Publishing Mode

Delta Publishing mode only supports the Timer polling method. When Delta Publishing mode is enabled, Publication Service checks the input file on a preconfigured timer interval, and copies any new data to a work file in the WIP directory, and then processes and publishes it.

To better explain the mechanism of Delta Publishing mode, the delimitedReader example is used. The example is located in the `SimpleRecord.zip` file in the `TIB_ADFILES_HOME\examples\samples \BusinessWorks5` directory.

In the delimitedReader example, the `delimited.txt` input file is shown as follows:

```
Order, ID41678, 20May2000
Item, GigaWidget, 60, $75
Item, MegaBucket, 48, $125
Customer, Hopkins Associates, ID26490
Order, ID41680, 20May2000
Item, Rt.Clopper, 40, $100
Item, Lt.Clopper, 50, $100
Customer, Jersey WebInovaters, ID46786
```

When using Delta Publishing mode, you have to consider the following two conditions:

- Each line

  Each line might be written in the input file by another application at intervals. The intervals are set at three times of the polling interval that you have set in the **Configuration** tab. The adapter can handle all kinds of writing rates for each line when you set a proper polling interval.

  When a line is published as one message in Delta Publishing mode, the intervals are reset. When setting the polling interval, you have to consider the writing rate of the application.

- Each complete message

  A complete message is written in the input file at intervals. The intervals are set at three times of the polling interval you have set in the **Configuration** tab. In this example, according to Read Schema, `Order` is the root record, so the first 4 lines form a complete message, and the last 4 lines form another complete message.

  In Delta Publishing mode, the adapter cannot recognize where the next complete message starts, and the data is written in the input file by another application continuously. If no data is written in the input file at the specified intervals (3 times of the polling interval), the last 4 lines are published as a complete message.

  After the last 4 lines are published as a complete message in Delta Publishing mode, other data appended to the last line is not allowed. For example, after publishing the last 4 lines as a complete message, it is not allowed to append data to line 8 as follows: `Customer, Jersey WebInovaters, ID46786 fdsfdsfds`

Big Files Handling

This function is used to handle big files. When `adfiles.pub.multithread4Bigfile` is enabled, you can handle a big file with multi-thread, and then you can set the `adfiles.pub.multithreadCount` property to `2`, `3`, `4`, or `5`.

This function is not supported when:

- the Position From Header of a positional child record is **End**
- Delta Publishing mode is enabled
- the Data Format is Business document
- the Life Cycle is repeating
- the big file contains base records

When you assign a value to the `adfiles.pub.multithreadCount` property, you have to consider the following factors:

- Performance of the machine
- File size
- Size and schema of each message

CheckPoint Restart

CheckPoint Restart stores records in a progress file when the adapter crashes. If the adapter crashes when processing a file, Publication Service continues to process the file from where it crashes upon restart. The restart information is stored in a progress file in the directory where the runtime adapter is executed.

To enable this functionality for Publication Service:

- In Record Transfer mode, select the **CheckPoint Restart** check box in the **Advanced** tab.
- In Simple File Transfer mode, specify the progress file name in the **SFT** tab. The name of the progress file is `__InstanceNamePublicationServiceName.prg`.

If Load Balancing mode is enabled, the subject or destination name is added to the progress file name. For example: `__InstanceNamePublicationServiceNameSubjectName.prg`.

Since Publication Service does not update the progress file or perform postprocessing on the file automatically, when the adapter exits abnormally, the following situations occur occasionally:

- Duplicate Messages

  Because Publication Service does not publish messages or update the progress files automatically, duplicate files might be created when Publication Service resumes. The CheckPoint Restart function is used only if duplicated messages are tolerant in the project.

  When using Object (MInstance) or the grouping size of the Business document is one, only one duplicate file is generated. When using Business document, and the grouping size is greater than one, multiple duplicate files are generated.

  When you use the MInstance message format, the subscriber occasionally writes a duplicate message after a checkpoint restart.

  The MBusiness Doc format uses sequence numbers to prevent duplicate messages. This preventative feature is only effective if the Document Delay parameter is set to `0` (zero).

- Missing File in the Working Directory

Missing file in the working directory occurs when the postprocessing on the working file is completed but the progress file is not updated. Upon restart, the adapter prompts that the file in the working directory is missing.

> 📝
> - Enabling the CheckPoint Restart feature results in lower performance. Disabling this feature means a higher throughput is possible, but you have to restart the service manually.
> - When CheckPoint Restart is enabled, Publication Service sends duplicate messages, and Subscription Service might output duplicate messages even though Publication Service does not resend them.
> - Ensure that you set the progress directory if you want to enable the CheckPoint Restart feature.

Configuring the Log4J Log

The adapter supports two log types: the SDK log and the Log4J log. The SDK log is used by default. If you want to use the Log4J log, you have to set the `adfiles.trace.extended` property to `ON` in the `adfilesagent.tra` file.

The adapter provides a default `adfiles.log4j.properties` file, which is a sample for configuring the Log4J properties. The properties file is located in the `TIB_ADFILES_HOME/bin` directory.

Procedure

1. In the `adfilesagent.tra` file, set the `adfiles.trace.extended` property to `ON`.
   This is to enable the Log4J log.

2. Specify a path for the `adfiles.trace.log4j.properties` property, the default path is `%TIB_ADFILES_HOME%/bin/adfiles.log4j.properties`.
   This is to specify the path for the Log4J configuration file.

3. Configure the `adfilesagent.tra` file, and invoke the refreshExtendedLogging() method in Hawk.

   This is to refresh the Log4J log without restarting the adapter whenever changes are made to the Log4J properties file.

Fault Tolerance

Multiple adapter configurations can substitute for each other with Fault Tolerance. When the primary adapter configuration ends unexpectedly, the token held by the primary adapter configuration can be taken over by an adapter configuration in the standby state. In the process of replacement, the standby adapter configuration is promoted to the primary adapter configuration.

Fault Tolerance is based on the JMS queue. Before enabling Fault Tolerance, you have to define a JMS queue, set the **prefetch** parameter of the JMS queue to `none`, and then put several JMS messages in the JMS queue as tokens. The number of tokens corresponds to the number of primary adapter configurations.

> 📝
> - When a standby adapter configuration becomes a primary adapter configuration, it does not take the instance ID of the original primary adapter configuration that ended unexpectedly and still has its own instance ID.
> - When running JMS topic as durable, durable names exist on the EMS server for each receiver, regardless of whether adapter configuration is primary or standby.
> - To detect broken connections more quickly, you can add the `client_heartbeat_server=3` property to the `tibemsd.conf` files of all the primary servers and standby servers.

The following diagram shows how Fault Tolerance works. At first, adapter configuration 1 and 2 fetch one of the two tokens in the JMS queue respectively. They hold the tokens and process messages as

primary adapter configurations. Adapter configuration 3 does not fetch tokens and runs in standby state. If adapter configuration 2 ends unexpectedly, it releases the fetched token. Adapter configuration 3 fetches the token released by adapter configuration 2 and continues to process messages as primary adapter configuration.
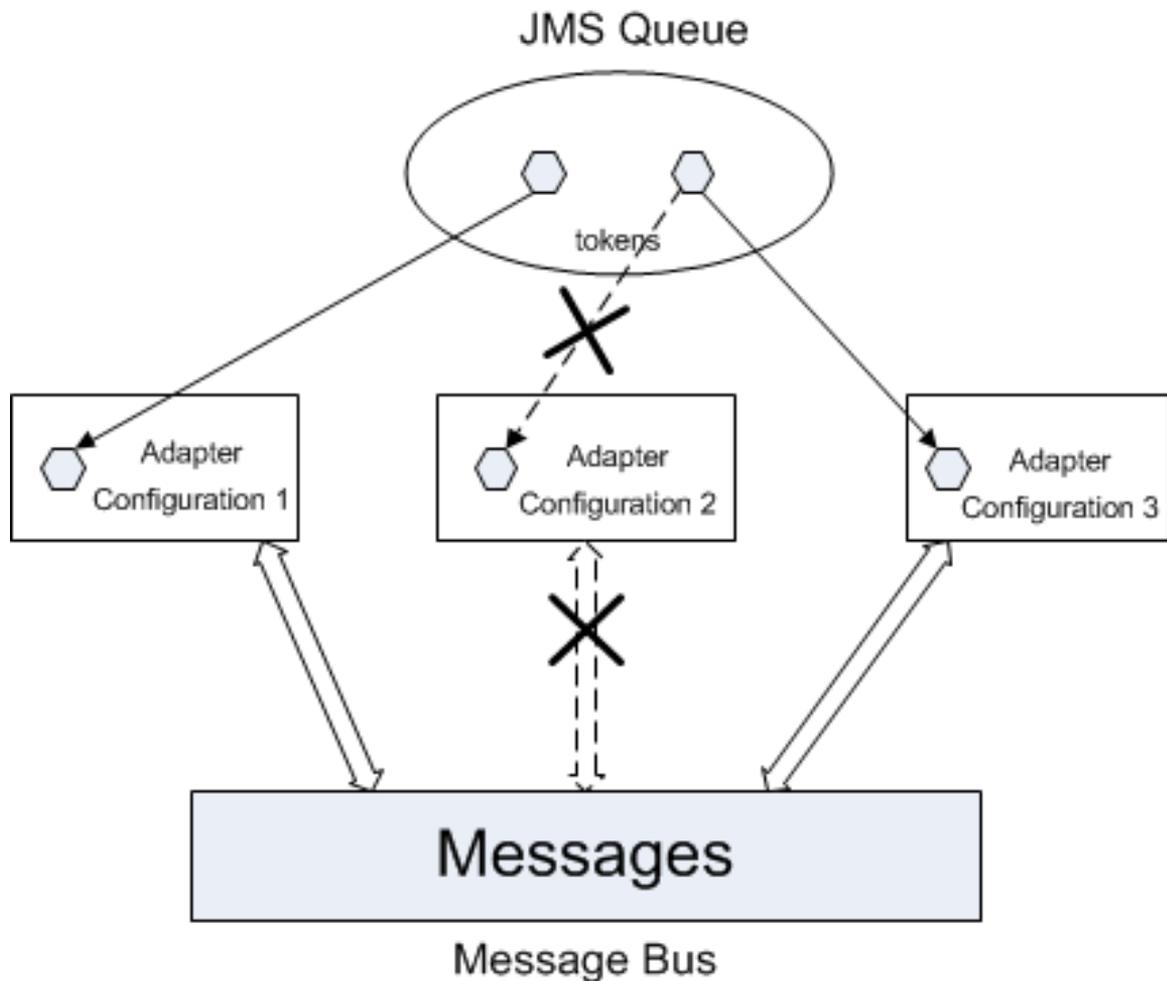
When Fault Tolerance is enabled, the following two issues occur:

1. If the number of tokens is more than 1, an exception is thrown in any of the following conditions:

   - The transport type is RVCM.

   - The transport type is JMS. The primary and standby adapter configurations have the same client ID.

   - The delivery mode is Durable in Subscription Service and Request-Response Service.

2. If a primary EMS server switches to the standby state, all primary adapter configurations that fetch tokens from the primary EMS server restart.

Enabling Fault Tolerance

You can configure Fault Tolerance properties in the TRA file and in TIBCO Administrator.

To enable the Fault Tolerance features, set the `tibco.sdk.faultTolerance.ems.enabled` property to `ON` in the `adfilesagent.tra` file, and set SDK fault tolerance properties accordingly.

To enable SDK fault tolerance in the adapter, conform to the following configurations:

- In Publication Service:

  - In Record Transfer mode, select the **CheckPoint Restart** check box in the **Advanced** tab.

  - In Simple Transfer mode, specify the progress file name in the **SFT** tab.

- In Subscription Service, do not select **One message per file** from the **Wip Creation Mode** list when the transfer mode is Record Transfer.

- Make sure only one primary adapter configuration is running.

- When the transport type is JMS, ensure that the delivery mode is **Persistent** for Publication Service, and **Durable** for Subscription Service.

- When the transport type is RVCM, you cannot use the **Once-only** life cycle for Publication Service.

- When the SDK fault tolerance and CheckPoint Restart features are enabled, Load Balancing mode or Delta Publishing mode is not supported.

Module Properties

Module properties provide an easy way to set defaults for use throughout your project. Module properties are visible within the module, you cannot change them from TIBCO Enterprise Administrator, and you cannot assign them to an activity directly. You have to reference a module property from a process property, and then reference the process property from the activity.

Module properties used in the adapter are called global variables when using TIBCO Designer with TIBCO ActiveMatrix BusinessWorks 5.

In TIBCO Business Studio, you can use module properties in several ways:

- Define a property using TIBCO Business Studio, and then override the value for individual applications at deployment time using TIBCO Enterprise Administrator. You can also override values for predefined properties unless you cannot set them on the user interface later.

- Predefine a property using TIBCO Business Studio, and then override the value for individual services (for example, Publication Service or TIBCO ActiveMatrix BusinessWorks process) at deployment time using TIBCO Enterprise Administrator. The values you specify are used at run time. You can also override values for predefined properties unless you cannot set them on the user interface later.

  For example, when you assign a value `7474` to the `RvDaemon` module property, you can use the property in different sessions in the adapter. If you want to change the TIBCO Rendezvous daemon for the adapter, you can globally set it to a different value or override it from the command line.

  You can also add and define module properties in the TRA file. The module property follows the convention: `adfiles.`*instance-name_service-name*`_UserId`.

Module properties can be specified in a TRA properties file or on the command line at design time. A property value set on the command line overrides the same property value set in the properties file. Similarly, a property value set in the properties file overrides the same property set in TIBCO Business Studio.

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

The property substitution mechanism can override module properties predefined in the project in a restricted manner. For details, see Property Substitution.

For how to add, specify, and group module properties in the Module Properties editor, see Using Module Properties.

You can also use Binding Editor 🔲 to specify module properties. For details, see Configuring Module Properties in Binding Editor.

A number of module properties are predefined, see Predefined Module Properties. You can add definitions to the predefined properties.

Property Substitution

The property substitution mechanism can override module properties predefined in a project in a restricted manner. You can view and set predefined properties in TIBCO Business Studio. Properties are specified as *%%VARNAME%%* and cannot contain any white space.

With property substitution, you can accomplish the following operations:

- Substitute string, int, and other properties specified in a project at startup time.
- Locally define the value for a property for a specific project. The local value takes precedence over any global value.

- Specify the value for a property in a properties file. This overrides the project repository and values set in code, but not properties set on the command line.

- Enforce the predefined properties listed in Predefined Module Properties.

  Properties can be used anywhere in the configuration and can be replaced by the locally-defined adapter configuration.

Using Module Properties

You can add, specify, and group module properties in the Module Properties editor.

Procedure

1. In the Project panel, expand the **Module Descriptors** folder, and then double-click the **Module Properties** folder.
   The Module Properties editor is displayed in the right panel.

2. Define module properties in the Module Properties editor.

   - To add a property, click **New Property**. A new global property item is added to the bottom of the list. Specify the property name and the value in the default column. Press Enter when you're done.

   - To assign or change a property value, click the property. The information of the property is displayed in the panel below, you can change the property name. To change its value, double-click the default of the property, assign a new value, and press Enter when you're done.

   - To add a new property group, click **New Group**. Specifies the name of the group in the panel below. With the group selected, you can click **New Property** to add properties to the group.

3. Use the module property in the fields of a resource.

   - Enter the property name by using `%%` on both sides.

     For example, you must enter `%%Name%%` in the **Name** field to use the `Name` property.

   - Click **Binding Editor** to specify module properties. For details, see Configuring Module Properties in Binding Editor.

Configuring Module Properties in Binding Editor

Binding Editor is a tool to edit, pick, and clear module properties for adapter configurations.

Binding Editor is available only for the **Configuration** tab and the **Processing** tab.

You can select multiple module properties and edit each property using Binding Editor. The global variable type has to match the selected module property type.

If the input field has bound global variables, you have to clear the binding in the Binding Editor dialog, and then specify the value manually in the field.

Procedure

1. Define a module property in the Module Properties editor.

2. Click **Binding Editor** .
   Binding Editor provides three functions to configure module properties: pick, edit and clear. You can complete the following steps based on your requirements.

3. In the Binding Editor dialog, click **Pick** to select a substitution variable from the pop-up list.
   a) Select one of the properties in the Binding Editor dialog and click **Pick**.
   b) In the Select String Substitution Variable dialog, enter a search expression in the **Select an item to open** field.

      This search expression is used to filter out matching variables.
   c) Select the matching variable from the Matching Items panel.
   d) Click **OK**.
   e) In the Global Variable dialog, use the default value and click **OK**.
   f) Click **OK** to save the picked property.

4. In the Binding Editor dialog, click **Edit** to edit the selected module property.
   a) Select one of the properties in the Binding Editor dialog and click **Edit**.
   b) In the Global Variable dialog, enter a new global variable in the format of `%%Property_Name%%` and click **OK**.
   c) Click **OK** to save the property.

5. In the Binding Editor dialog, click **Clear** to clear the binding of a module property.
   a) Select one of the properties in the Binding Editor dialog.
   b) Click **Clear** to clear the variable value in the binding column.
   c) Click **OK** to save the property.

Predefined Module Properties

Some module properties are automatically used within the system when you configure an adapter configuration. Check the following table for the predefined module properties.

Property	Description
Deployment	Defaults to the TIBCO Business Studio project name. This value can be any string value. This module property is used by the system to partially define the subject name defined for a service.
DirLedger	Specifes 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 where the project is saved.
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.
JmsProviderUrl	Specifies where the JMS server is located. When only one JMS server is used, you can set this value in early stages of a project.
JmsSslProviderUrl	Specifies where the JMS SSL daemon is located.
RemoteRvDaemon	TIBCO Rendezvous routing daemon (`rvrd`) to be used. For details about setting up a domain using `rvrd`, see *TIBCO Administrator Server Configuration Guide*.

Property	Description
RvDaemon	TIBCO Rendezvous daemon. Sessions use this daemon to establish communication. The default value is `7500`.
RvNetwork	TIBCO Rendezvous network. This property is 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 as is.
RvService	TIBCO Rendezvous service. The TIBCO Rendezvous daemon divides the network into logical partitions. Each transport communicates on a single service. A transport can communicate only on the same service as other transports. Unless you are using a non-default TIBCO Rendezvous configuration, you can leave the default (`7500`).
RvaHost	The computer on which the TIBCO Rendezvous agent runs. This property is only relevant if you are using the TIBCO Rendezvous Agent (`rva`) instead of the TIBCO Rendezvous daemon, and if you configured a non-default setup. For details about specifying the `rva` parameters, see *TIBCO Rendezvous Administration*.
RvaPort	The TCP port where the TIBCO Rendezvous agent (`rva`) listens for client connection requests. For details about specifying the `rva` parameters, see *TIBCO Rendezvous Administration*. The default value is `7600`.
TIBHawkDaemon	TIBCO Rendezvous daemon used in the TIBCO Hawk session. For details about this parameter, see *TIBCO Hawk Installation and Configuration*.
TIBHawkNetwork	TIBCO Rendezvous network used by the TIBCO Hawk session. For details about this parameter, see *TIBCO Hawk Installation and Configuration*.
TIBHawkService	TIBCO Rendezvous service used by the TIBCO Hawk session. For details about this parameter, see *TIBCO Hawk Installation and Configuration*.

Adapter Microagents and Methods

You can use TIBCO Hawk microagents to supplement the monitoring information provided by the standard logging levels capability.

Examples of supplemental information that you can obtain with microagents include the repository URL, command line arguments used to start the adapter configuration, and so on.

Each adapter configuration has the following three microagents, with different capabilities and names. The same microagent follows different naming conventions, depending on how an adapter configuration is started: from TIBCO Business Studio, or from TIBCO Administrator.

- Standard Microagent

  Predefined in TIBCO Adapter SDK. You can use the microagent to perform queries on all running adapter configurations, regardless of their class or application.

  - Naming in TIBCO Business Studio

    COM.TIBCO.ADAPTER.adfiles.%%Deployment%%.%%InstanceId%%

  - Naming in TIBCO Administrator

    COM.TIBCO.ADAPTER.adfiles.domainName.appspaceName.nodeName.%%applicationName%%.%%InstanceId%%

- Class Microagent

  Predefined in TIBCO Adapter SDK. You can use the microagent to perform queries on one class of the adapter configuration.

  - Naming in TIBCO Business Studio

    COM.TIBCO.adfiles.%%Deployment%%.%%InstanceId%%

  - Naming in TIBCO Administrator

    COM.TIBCO.ADAPTER.adfiles.%%Deployment%%.%%InstanceId%%

- Custom Microagent

  Predefined in TIBCO ActiveMatrix Adapter for Files for Unix/Win. You can use the microagent to perform the adapter specific queries.

Available TIBCO Hawk Methods

The following table lists the microagent methods available for the adapter.

For more information about the methods available in the standard and class microagents, see " TIBCO Adapter SDK Hawk Microagents and Methods" in *TIBCO Adapter SDK Programmer's Guide*.

The getActivityStatisticsBySchema(), getActivityStatisticsByOperation(), getActivityStatisticsByService(), and resetActivityStatistics() methods are now deprecated. You must use the class microagent to get equivalent methods. When the value of the `adfiles.addCustomHawkMethodsToClassMAgent` property is set to `ON` in the TRA file, these methods are added to the class microagent.

Method	Description
activateTraceRole()	Activates a mapping of a role to a sink at run time.
deactivateTraceRole()	Deactivates a mapping of a role to sinks at run time.

Method	Description
getActivityStatisticsByOperation()	Returns statistics for one operation.
getActivityStatisticsBySchema()	Returns statistics for any activities on a particular object or schema.
getActivityStatisticsByService()	Returns statistics related to the data handled by a particular adapter service since it started.
getAdapterServicesInformation()	Returns information about the services implemented by this adapter.
getComponents()	Returns information about the publisher, subscriber and IODescriptor.
getConfig()	Returns basic configuration information. More specific information is accessed by using more specific methods.
getConfigProperties()	Returns all attributes and elements for the given repository object.
getDocumentDelay()	Returns the document delay setting for a given Publication Service.
getHostInformation()	Returns standard and extended application information.
getPollingInterval()	Returns the current polling interval setting.
getQueueStatistics()	Returns the current count of elements in any internal queue used by the adapter.
getRvConfig()	Returns information about all defined TIBCO Rendezvous sessions.
getStatus()	Returns general status information, such as the number of TIBCO Rendezvous messages received and published, the number of errors that have occurred since the last call, the PID of the application, and so on.
getTraceSinks()	Returns information about sinks to which traces currently go.
getVersion()	Return the configuration ID, application name, version, and date for this adapter instance.
_onUnsolictedMsg()	Displays alert messages sent to the current adapter.
preRegisterListener()	Preregisters an anticipated listener.
resetActivityStatistics()	Resets all the counts for the activity statistics.
reviewLedger()	Returns information retrieved from the ledger file of a certified messaging session for a publisher adapter.

Method	Description
setDocumentDelay()	Sets the document delay for Publication Service.
setPollingInterval()	Sets the polling interval for Publication Service.
setTraceSinks()	Adds a role or changes the file limit of a previously specified sink.
stopApplicationInstance()	Stops the running adapter instance.
unRegisterListener()	Unregisters a currently preregistered listener.
refreshExtendedLogging()	Refreshes the Log4J log without restarting the adapter whenever changes are made to the Log4J properties file.

activateTraceRole()

Activates a mapping of a role to a sink at run time. This replaces the deprecated setTraceSink() TIBCO Hawk method.

The following table lists the parameters:

Parameter	Type	Description
`Role Name`	String	The name of the role to activate.
`Sink Name`	String	The name of the sink for which to activate the role.

deactivateTraceRole()

Deactivates a mapping of roles to sinks at run time.

The following table lists the parameters:

Parameter	Type	Description
`Role Name`	String	The name of the role to activate.
`Sink Name`	String	The name of the sink for which to activate the role.

getActivityStatisticsByOperation()

Returns the total number of objects processed for all the schemas by each service that is associated with a specified operation, and returns the number of success and error objects.

The following table lists the parameter:

Parameter	Type	Description
`Operation`	String	The type of operation: read or write.

Returned Results

The following table lists the returned results:

Name	Type	Description
Service Name	String	The name of the service that is associated with the specified operation.
Total	String	The total number of objects processed for this schema for an adapter service.
Success	String	The number of objects that are successfully identified for this schema, and are to be published or written to a file.
Error	String	The number of objects that are identified for this schema but are not published. This reason is that the header of the schema fails in the validation for Publication Service, or is written to a file because the schema is not associated with the subscriber for Subscription Service.

getActivityStatisticsBySchema()

Returns the total number of objects processed for the given schema by each service that uses the schema, and returns the number of success and error objects.

The following table lists the parameter:

Parameter	Type	Description
Schema Name	String	The name of the schema.

Returned Results

The following table lists the returned results:

Name	Type	Description
Service Name	String	The name of the service that is associated with the specified operation.
Total	String	The total number of objects processed for this schema for an adapter service.
Success	String	The number of objects that are successfully identified for this schema, and are to be published or written to a file.
Error	String	The number of objects that are identified for this schema but are not published. This reason is that the header of the schema fails in the validation for Publication Service, or is written to a file because the schema is not associated with the subscriber for Subscription Service.

getActivityStatisticsByService()

Returns the total number of objects processed for each of the schemas associated with the specified service, and returns the number of success and error objects.

The following table lists the parameter:

Parameter	Type	Description
`Service Name`	String	The name of the service.

Returned Results

The following table lists the returned results:

Name	Type	Description
`Operation`	String	The type of operation that the service performs.
`Service Name`	String	The name of the service that is associated with the specified operation.
`Total`	String	The total number of objects processed for this schema for an adapter service.
`Success`	String	The number of objects that are successfully identified for this schema, and are to be published or written to a file.
`Error`	String	The number of objects that are identified for this schema but are not published. This reason is that the header of the schema fails in the validation for Publication Service, or is written to a file because the schema is not associated with the subscriber for Subscription Service.

getAdapterServicesInformation()

Returns information about the services implemented by this adapter. The information is a summary of available adapter services.

The following table lists the parameter:

Parameter	Type	Description
`Service Name`	String	The name of the service from which to get information. The default value is ALL.

Returned Results

The following table lists the returned results:

Name	Type	Description
`Line`	Integer	The sequential row number.
`Service Name`	String	The name of the service defined at design time.
`Endpoint Name`	String	The name of the endpoint used for this service.
`Type`	String	The type of the endpoint, for example, publisher or subscriber.

Name	Type	Description
`Quality of Service`	String	The quality of service for the endpoint. For example, RVCM or JMS Persistent.
`Subject`	String	The subject defined for this endpoint.
`Class`	String	The class associated with the endpoint.
`Number of Messages`		The number of messages processed for this endpoint.

getComponents()

Returns information about the currently active TIBCO Hawk components such as publishers, subscribers, or timers.

The following table lists the parameters:

Parameter	Type	Description
`Component Name`	String	The name of the component. If no value is entered, all components are displayed.
`Component Type`	String	Any of Publisher, Subscriber, Timer, or IODescriptor. The default value is `All`.

Returned Results

The following table lists the returned results:

Name	Type	Description
`Instance ID`	String	The name of the adapter configuration defined at design time.
`Adapter Name`	String	The name of the adapter.
`Component Name`	String	The 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 the TIBCO Adapter SDK documentation.
`Session Name`	String	The name of the session.
`Description`	String	The 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.

The following table lists the returned results:

Name	Type	Description
`Instance ID`	String	The adapter configuration ID.
`Adapter Name`	String	The name of the adapter.
`Repository Connection`	String	The URL of the repository used for the adapter configuration.
`Configuration URL`	String	The location of the adapter project. It is either a filename or configuration URL.
`Command`	String	The command line arguments used to start the adapter.

getConfigProperties()

Returns all attributes and elements for the given repository object.

The following table lists the parameter:

Parameter	Type	Description
`Property`	String	The name of the property for which elements (tags) and attributes are used. For example, `agentone/startup`. If no value is entered, all properties are returned.

Returned Results

The following table lists the returned results:

Name	Type	Description
`Element Name`	String	The repository directory for the property.
`Attribute Name`	String	The name of the repository object attribute.
`Attribute Value`	String	The value of the repository object attribute.
`Line`	Integer	The line number in which this property is defined in the project file.

getDocumentDelay()

Returns the document delay setting for a given Publication Service. If Publication Service is not given, a list of Publication Service services is returned with their corresponding polling interval.

getHostInformation()

Returns standard and extended application information set.

The following table lists the returned results:

Name	Type	Description
Name	String	The name of the property.
Value	String	The value of the property.

getPollingInterval()

Returns the current polling interval setting.

The following table lists the returned results:

Name	Type	Description
PollingInterval	String	Polling interval in milliseconds.

getQueueStatistics()

Returns the current count of elements in any internal queue used by the adapter. This includes the TIBCO Rendezvous event queues automatically created by Rendezvous for each adapter.

The following table lists the returned results:

Name	Type	Description
QueueID	String	A unique identification of a particular queue.
QueueType	String	A type or key that matches this queue to a thread or connection.
QueueCount	Integer	The current number of elements in the queue.
MaxQueueSize	Integer	The maximum number of elements in the queue.

getRvConfig()

Returns information about the TIBCO Rendezvous session defined by this adapter. Information about all currently defined sessions is returned if no value is provided for the **sessionName** parameter.

The following table lists the parameter:

Parameter	Type	Description
Session Name	String	The name of the TIBCO Rendezvous session for which configuration is required. If not given, information about all sessions is returned. The default value is `All`.

Returned Results

The following table lists the returned results:

Name	Type	Description
Instance ID	String	The adapter configuration ID.

Name	Type	Description
`Adapter Name`	String	The name of the adapter.
`Session Name`	String	The name of the session.
`Service`	String	The service parameter for this session.
`Daemon`	String	The daemon parameter for this session.
`Network`	String	The network parameter for this session.
`Synchronous?`	Boolean	Returns 1 if this is a synchronous session, returns 0 if this is an asynchronous session.
`Session Type`	String	The type of the session. The available types are: **M_RV**, **M_RVCM**, or **M_RVCMQ**.
`Certified Name`	String	The name of the certified session.
`Ledger File`	String	Ledger file for the certified messaging session. Returns the empty string for sessions that are not certified messaging sessions.
`CM Timeout`	String	Timeout for the 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 additional methods, see getConfig() and getRvConfig().

The following table lists the returned results:

Name	Type	Description
`Instance ID`	String	The adapter configuration ID.
`Adapter Name`	String	The name of the adapter.
`Uptime`	Integer	Time in seconds since startup.
`Messages Received`	Integer	The number of TIBCO Rendezvous messages received.
`Messages Sent`	Integer	The number of TIBCO Rendezvous messages published.
`New Errors`	Integer	The number of errors since the last call to this method.
`Total Errors`	Integer	The total number of errors since startup.
`Process ID`	Integer	The process ID of the application.

Name	Type	Description
`Host`	String	The name of host machine on which the adapter is running.

getTraceSinks()

Returns information about sinks to which traces go currently.

The following table lists the parameters:

Parameter	Type	Description
`Sink Name`	String	The name of the sink for which you need information. If no name is specified, information about all sinks is returned. The default value is `All`.
`Role Name`	String	The name of the role for which you need information for the specified sink or sinks. The default value is `All`.

Returned Results

The following table lists the returned results:

Name	Type	Description
`Instance ID`	String	The adapter configuration ID.
`Adapter Name`	String	The name of the application for this sink.
`Sink Name`	String	The name of the sink.
`Sink Type`	String	The type of this sink. The available types are: fileSink, rvSink, hawkSink, or stderrSink.
`Roles`	String	Roles supported by the sink. For example, warning, error, or debug.

getVersion()

Retrieves version information for the current application. Two lines are returned: one for the TIBCO Adapter SDK, the other for the adapter.

The following table lists the returned results:

Name	Type	Description
`Instance ID`	String	The adapter configuration ID.
`Adapter Name`	String	The name of the adapter.
`Version`	String	The version number. For example, `1.3`.

_onUnsolictedMsg()

Displays all alert messages sent from the adapter or an error if not successful.

preRegisterListener()

Returns OK if Subscription Service is preregistered successfully; otherwise, returns false.

Preregisters an anticipated Subscription Service. Some applications anticipate requests for certified delivery even before the listening applications start running. In such situations, Publication Service preregisters Subscription Service, so TIBCO Rendezvous software begins storing outbound messages in Publication Service ledger. If the listening correspondent requires old messages, it receives the backlogged messages when it requests certified deliver.

The following table lists the parameters:

Parameter	Type	Description
Session Name	String	The name of the session that anticipates the listener.
Publisher Name	String	The name of the component for which the listener are preregistered.
Listener Session Name	String	The name of Subscription Service to preregister.

resetActivityStatistics()

Resets all the counts for the activity statistics.

reviewLedger()

Returns information retrieved from the ledger file of a TIBCO Rendezvous certified messaging session.

Before invoking this method, make sure that the certified messaging publisher adapter has established a certified delivery agreement with its subscriber agents.

The following table lists the parameters:

Parameter	Type	Description
Session Name	String	The name of the TIBCO Rendezvous session for which ledger information is needed. The default value is All.
Subject	String	The name of the subject for which ledger information is needed.

Returned Results

The following table lists the returned results:

Name	Type	Description
Session Name	String	The name of the TIBCO Rendezvous CM session to which this information applies.
Subject	String	The subject name for this session.

Name	Type	Description
Last Sent Message	Integer	The sequence number of the most recently sent message with this subject name.
Total Messages	String	The total number of pending messages with this subject name.
Total Size	Integer	The total storage (in bytes) occupied by all pending messages with this subject name. If the ledger contains ten messages with this subject name, 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.
Line	Integer	Row number in ledger file.
Unacknowledged Messages	Integer	The 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.

setDocumentDelay()

Sets the document delay for Publication Service.

The following table lists the parameters:

Parameter	Type	Description
DocumentDelay	Integer	Document delay time in milliseconds.
ServiceName	String	The name of Publication Service.

setPollingInterval()

Sets the polling interval for Publication Service.

The following table lists the parameters:

Parameter	Type	Description
PollingInterval	Integer	Polling interval in milliseconds.
ServiceName	String	The name of service where the polling interval is set.

setTraceSinks()

Adds a role or changes the file limit of a previously specified sink. Returns OK if successful; returns an error if not successful.

The following table lists the parameters:

Parameter	Type	Description
`Sink Name`	String	The name of the sink for which you want to add a role or change the file limit.
`Role Name`	String	The name of the role you want to add to this sink (warning, error, debug, or user defined). The default value is `All`.
`File Size`	Integer	The maximum file size for the sink. This parameter is ignored if the sink specified by sinkName is not a file sink.

stopApplicationInstance()

Stops the specified adapter by calling the internal stop() method. This method returns `OK` if successful, and returns an error if not successful.

unRegisterListener()

Unregisters a currently preregistered Subscription Service. This method returns `true` if Subscription Service is unregistered successfully. Otherwise, it returns `false`.

The following table lists the parameters:

Parameter	Type	Description
`Session Name`	String	The name of the session that anticipates Subscription Service.
`Publisher Name`	String	The name of Publication Service, to which Subscription Service is preregistered.
`Listener Session Name`	String	The name of Subscription Service to be unregistered.

refreshExtendedLogging()

Refreshes the Log4J log without restarting the adapter whenever changes are made to the Log4J properties file.

Two log types are available: SDK log and Log4J log. By default, SDK log is used. If you want to use the Log4J log, you have to set properties to enable it.

To enable the Log4J log, complete the following steps:

Procedure

1. Set the `adfiles.trace.extended` property to `ON`.

   This is to enable Log4J log.

2. Specify a path for the `adfiles.trace.log4j.properties` property, the default path is `%TIB_ADFILES_HOME%/bin/adfiles.log4j.properties`.

   This is to specify the path for the Log4J configuration file.

3. (Optional) Set the `adfiles.trace.thread` property to `ON`.

Adapter Properties

The runtime adapter properties are in the `adfilesagent.tra` file. You can add, modify, delete adapter properties in the TRA file.

TIBCO ActiveMatrix Adapter for Files for Unix/Win properties are separated in two categories: the required properties and properties specific to the adapter. You can check this section for the adapter properties file, the required properties, and properties specific to the adapter.

- Adapter Properties File
- Required Properties
- Adapter Specific Properties

Adapter Properties File

The runtime adapter parses one or more properties files at startup. The default runtime adapter properties file is named as `adfilesagent.tra`.

The default properties file is located in the *TIB_ADFILES_HOME*\bin directory.

The adapter also provides template TRA files for each properties file when the actual properties file is corrupted or deleted by mistake.

Each line in a properties file is a single property. Each property consists of a key and a value. The key starts with the first non-whitespace character and ends with the first occurrence of these characters:

`(space) : =`

The value starts at the first character after any of the three characters listed above.

For example:
```
tibco.configurl=/tibco/private/adapter/test/config/config1
tibco.repourl=tibcr://TEST_PROJECT
tibco.username=admin
tibco.password=samplePassword
tibco.clientVar.service=7600
tibco.clientVar.daemon=tcp:7600
```

Properties defined in the properties file override the same properties defined in the project. The properties file for a deployed service is located in the *TIBCO_TRA_DOMAIN_HOME/domainName/*`application/`*applicationDeploymentName* directory.

Properties File Format

When using properties, you have to consider the following restrictions:

- Do not use the exclamation point (!) as a comment line indicator. Instead, use the number sign (#).
- The line continuation character is ignored, and you cannot define a property with multi-line values.
- A key cannot contain any of the termination characters. Although you can use termination characters by escaping the value with a preceding backslash (\) in Java, the adapter does not support this syntax.
- All paths inside a properties file, including Microsoft Windows directory names, must use forward slashes.

Required Properties

Some properties are required by a runtime adapter, and most required properties are predefined by TIBCO Adapter SDK. You can check the following table for required properties.

Property	Description
tibco.repourl *repository_url*	The absolute pathname to the local repository where the adapter configuration is defined. For a remote project, the `repourl` value uses the form `tibco.repourl tibcr@`*repository_name*. For example: `tibco.repourl tibcr@AdapterRepoDefault`.
	For Unix systems, the path separator must include a single forward slash (/). For example: `/local/tibco/repo/ repo.dat`
tibco.configurl *relative_path* or tibco.configurl *absolute_path*	The location of the adapter service inside the project file. If a relative path is specified, the adapter service is assumed to be in the default area in the project file (`/tibco/ private/adapter/`). For example, you can connect the `tibco.configurl adapterpub` value to an `adapterpub` adapter service in the `/tibco/private/adapter/` directory.
	If an absolute path is specified, the adapter configuration is looked up in the repository as defined by the argument. For example: `tibco.configurl/tibco/private/ adapter/adapterpub`
tibco.instanceid *instance_name*	The name of the adapter configuration.
	The length of the name cannot be larger than 80 characters.
application.args *arguments*	The properties (TRA) file to be passed to the running adapter. For example: `application.args - system:propFile `*TIBCO_TRA_DOMAIN_HOME/domainName/*`application/`*applicationDeploymentName/*`*applicationDeploymentName-ServiceName*`.tra`
application.start.dir *adapter_path_name*	The path name of the adapter to start. For example: `application.start.dir `*TIB_ADFILES_HOME*`\bin`.

Adapter Specific Properties

You can predefine TRA properties in TIBCO ActiveMatrix Adapter for Files for Unix/Win.

The following table lists the predefined properties in the alphabetical order. Properties that start with `ntservice` are available only on Microsoft Windows platforms.

Properties for Publication Service and Subscription Service

The following table lists properties applicable to both Publication Service and Subscription Service:

Property	Description
`adfiles.traceOldMessages`	Either `ON` or `OFF`. The default value is `OFF`.
	When it is set to `ON`, it enables 4.x style log messages and disables 5.x style log messages.
`adfiles.EEMEnabled`	Either `ON` or `OFF`. The default value is `OFF`.
	It enables or disables Business Event Messages. See Creating Business Event Messages on using this property.
`adfiles.perfMon`	Either `ON` or `OFF`. The default value is `OFF`.
`adfiles.addCustomHawkMethodsToClassMAgent`	Either `ON` or `OFF`. The default value is `OFF`.
	When it is set to `ON`, you can add custom methods to the adapter standard microagent.
`adfiles.SFTUseJMSMapMsg`	Either `ON` or `OFF`. The default value is `OFF`.
	When it is set to `ON`, SFT mode is enabled to exchange JMS map messages.
`adfiles.trace.devdebug`	Either `ON` or `OFF`. The default value is `OFF`.
	When it is set to `ON`, debug log is enabled for issue investigation for the adapter at run time.
`adfiles.trace.extended`	Either `ON` or `OFF`. The default value is `OFF`.
	When it is set to `ON`, the Log4J log is enabled.
`adfiles.trace.thread`	Either `ON` or `OFF`. The default value is `OFF`.
	When it is set to `ON`, the thread ID is traced in the Log4J log.
`adfiles.trace.log4j.properties`	Specifies the path for the Log4J properties. The default path is `%TIB_ADFILES_HOME%/bin/adfiles.log4j.properties`.

Properties Only for Publication Service

The following properties are for only applicable for Publication Service:

Property	Description
`adfiles.logSftProgressEveryNBlocks`	The default value is not to log progress status.
	The value is integer. When you set the value to `10`, the adapter prints a progress status every 10 messages.

Property	Description
adfiles.sendLotEndWithDataMsg	Either ON or OFF. The default value is ON.
	When it is set to ON and the Data Format is **Business Document**, Publication Service sends the **LotEnd** attribute either in an independent message or with the very last data message of a file.
	See *TIBCO ActiveMatrix Adapter for Files for Unix/Win Concepts* about the **LotEnd** attribute.
adfiles.schemaDiagnosticsFile Format	Either txt or xml. The default value is xml.
	When Publication Service finds records that do not match the schemas, Publication Service writes them to a file either as plain text or in XML-like format.
adfiles.matchFieldCountAndRec ordLength	Either ON or OFF. The default value is OFF.
	It enables or disables checking field count and record length when validating a record.
adfiles.quotedField	Either ON or OFF. The default value is OFF.
	If ON, the fields in double quotes are treated and processed as one field.
adfiles.JMSCompress	Either ON or OFF. The default value is OFF.
	When it is set to ON, the JMS messages are compressed before being sent.
adfiles.DeltaFlushInterval	The default value is 3.
	In Delta Publishing mode, when no new data is appended to an input file after a specified amount of polling, the data remaining in memory is considered as complete data and published.
adfiles.LBEnabled	Either ON or OFF. The default value is OFF.
	When it is set to ON, Load Balancing mode is enabled, so multiple adapter configurations of the same Publication Service can process separate files in parallel.
adfiles.NullNumberFieldAsZero	Either ON or OFF. The default value is OFF.
	It enables or disables treating null fields of integer type as zero.
adfiles.PublishEmptyField	Either ON or OFF. The default value is OFF.
	When it is set to ON, the empty fields are published. When it is set to OFF, the empty fields are skipped and not published.
adfiles.BDGroupingFactor	At runtime, this value overrides the Grouping Factor value specified at design time. This property only applies to the Business Document data format.

Property	Description
`adfiles.pub.multithread4Bigfile`	Either `ON` or `OFF`. The default value is `OFF`. When it is set to `ON`, multiple threads are used to process a file. Subscription Service writes messages randomly. This property is valid only for Record Transfer mode.
`adfiles.pub.multithreadCount`	Defines the number of threads that are used to process a file.

Properties Only for Subscription Service

The following properties are for only applicable for Subscription Service:

Property	Description
`adfiles.startSubscriberOnNewFileBoundary`	Either `ON` or `OFF`. The default value is `ON`. When it is set to `ON`, a subscriber discards partial files and starts processing only from the beginning of a new file upon restart. This property only applies to Simple File Transfer mode, JMS transport, and Durable Delivery mode.
`adfiles.dontAddEOLToFinalRecord`	Either `ON` or `OFF`. The default value is `OFF`. When it is set to `ON`, EOL is not added to the final record of a file. This property only applies to Simple File Transfer mode with TEXT File Transfer mode, and File Content Encoding is anything other than ASCII.
`adfiles.toggleChildRecordsOrdering`	Either `ON` or `OFF`. The default value is `OFF`. When it is set to `ON`, the sequence of the child orders is reversed in the generated file. When it is set to `OFF`, the sequence of the child orders does not change. This property takes effect only when no child records are reordered in TIBCO Business Studio.
`adfiles.sub.discardDuplicateMessages`	Either `ON` or `OFF`. The default value is `OFF`. When it is set to `ON`, this property prevents Subscription Service from writing duplicate messages to the output file, which is caused by Publication Service restart. This property only applies to the Business Document data format.

Property	Description
adfiles.useBDEmbeddedFileName	Either ON or OFF. The default value is OFF.
	When it is set to ON, Subscription Service uses the filename embedded in the **LotId** attribute in the incoming Business Document message when creating the output file instead of using the filename specified in TIBCO Business Studio. This property only applies to the Business Document data format.
	See *TIBCO ActiveMatrix Adapter for Files for Unix/Win Concepts* about the **LotId** attribute.
adfiles.io.sub.exit4issue	Either ON or OFF. The default value is ON.
	When it is set to ON, Subscription Service does not confirm messages and exits when an IO error occurs. When it is set to OFF, Subscription Service confirms messages, ignores IO errors, and continues to process other messages.

Trace Messages

Trace messages provide information about adapter activities. The messages are logged to the console where the runtime adapter is 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 includes the following fields:

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

You can check the following table for the fields and corresponding description:

Field	Description
Timestamp	The timestamp of occurrence. For example,`2014 Feb 22 20:14:51:718 GMT -8`.
Adapter Identifier	The name of the adapter that writes the trace message. This is a combination of the adapter acronym and adapter configuration name. For example, the application identifier, `ADFILES.publisher1` identifies a TIBCO ActiveMatrix Adapter for Files for Unix/Win (TIBCO Business Studio) service named `publisher1`.
Role	`Info`, `Warn`, `Error`, and `Debug` are available roles. • Info: indicates normal adapter operation. No action is necessary. A trace message tagged with info indicates that a significant processing step is reached and has been logged for tracking or auditing purposes. Only info messages preceding a tracking identifier are considered significant steps. • Warn: indicates an abnormal condition was found. Processing continues, but special attention from an administrator is required. • Error: indicates an unrecoverable error occurred. Depending on the error severity, the adapter continues with the next operation or stops altogether. • Debug: indicates a developer defined trace message. In normal operating conditions, debug messages are not displayed. When configuring the adapter, you can define what roles are logged, what roles are not logged. For example, to improve performance, you can decide not to log Info roles.

Field	Description
Category	Trace message can be grouped as one of the following categories: • Adapter: the adapter is processing an event. • Configuration: the adapter is reading configuration information. • Palette: the adapter is interacting with the palette. • Publication: Publication Service is reporting this trace message. • Shutdown: the adapter is shutting down. • Startup: the adapter is starting. • Subscription: Subscription Service is reporting this trace message. • System: this category is not linked to a specific event process. The trace message might be related to a Windows Service related messages, memory allocation, file system error, and so on. • TibRvComm: the adapter is communicating with TIBCO Rendezvous.
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 General Status Messages.
Tracking Identifier	A unique identifier that is "stamped" on each message by the original adapter. The tracking identifier remains in effect from the beginning of a message 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 to be displayed.

General Status Messages

The following table lists status codes, detailed explanation of each error, where applicable, and ways to solve different errors.

Status Code and Status Message	Role	Category	Resolution
AEFA-000002 `Out of memory`	errorRole	System	Check the system to determine why it ran out of memory.
AEFA-000003 `Missing class definition for <class name> in the configuration`	errorRole	Configuration	Review the documentation. If that does not help, contact TIBCO support.

Status Code and Status Message	Role	Category	Resolution
AEFA-000004 `Missing attribute for <attribute name> in the configuration`	errorRole	Configuration	Review the documentation. If that does not help, contact TIBCO support.
AEFA-000005 `Attribute <attribute name> in the configuration has the wrong value <attribute name>`	errorRole	Configuration	Review the documentation. If that does not help, contact TIBCO support.
AEFA-000007 `Couldn't open file <filename>`	errorRole	System	Check the file system to ensure that all of the files exist and can be opened.
AEFA-000008 `Couldn't close file <filename>`	errorRole	System	Check the file system to determine why the file could not be closed.
AEFA-000009 `Couldn't read file <filename>`	errorRole	System	Check the file and the file system to determine why the error occurred.
AEFA-000011 `Pre Processing Script File directory <directory name> couldn't be found`	errorRole	Configuration	Make sure that the directory exists. If it does, correct the configuration to point to the correct directory for the Pre Processing Script File.
AEFA-000012 `Pre Processing Script File <filename> couldn't be found`	errorRole	Configuration	Make sure that the file exists. If it does, correct the configuration to point to the correct file and directory names for the Pre Processing Script.
AEFA-000013 `Post-processing script file <filename> couldn't be found`	errorRole	Configuration	Correct the configuration to point to the correct file and directory names for the Post Processing Script of the Publication Service.
AEFA-000014 `No access to Input Directory <directory name>`	errorRole	Configuration	Create the Input Directory or give access to it.

Status Code and Status Message	Role	Category	Resolution
AEFA-000015 `No access to Working Directory <directory name>`	errorRole	Configuration	Create the Working Directory or give access to it.
AEFA-000016 `No access to Completion Directory <directory name>`	errorRole	Configuration	Create the Completion Directory or give access to it.
AEFA-000017 `Working file <filename> already exists`	errorRole	Adapter	Remove the duplicate file from the Working Directory.
AEFA-000018 `Could not add file <filename> to Working Directory <directory name>. Reasons: 1. The Working Directory or Input Directory could be in read only mode. 2. The file specified does not exist anymore. 3. The file specified is locked by some other applications`	errorRole	System	1. If none of the files are processed, check the file system. Make sure the Working Directory and Input Directory exist and are in write mode. 2. If the file does not exist anymore then verify if you have other applications processing the input directory the adapter is using. 3. If the file is locked by some other application, the file will be processed in the next iteration. In such cases consider using trigger files.
AEFA-000019 `Cannot remove file <filename> from Working Directory <directory name>`	warnRole	System	Check the file system to determine why the error occurred.
AEFA-000020 `Cannot add TimeStamp to file <filename> in directory <directory name>`	warnRole	System	Check the file system to determine why the error occurred.
AEFA-000021 `Cannot move file <filename> to Completion Directory <directory name>`	warnRole	System	Check the file system to determine why the error occurred.

Status Code and Status Message	Role	Category	Resolution
AEFA-000022 `Cannot remove Trigger File <filename> from Input Directory <directory name>`	errorRole	System	Check the Trigger File rights and the file system state to determine why the error occurred.
AEFA-000023 `Cannot open data file <filename> in Working Directory <directory name>`	errorRole	System	Check the working data file rights and the file system state to determine why the error occurred.
AEFA-000024 `Cannot find input data file <filename> matching Trigger File <filename> in Input Directory <directory name>`	errorRole	Adapter	Check the existence of the input data file and the input data file rights.
AEFA-000025 `Cannot open input data file <filename> on CheckPoint Restart`	errorRole	Adapter	Check the existence of the input data file, the input data file rights, and the content of the CheckPoint Restart file.
AEFA-000026 `User exit failed for Publication Service <service name> Input file = <filename>; Message = <data message>`	errorRole	Adapter	Check the User exit server to determine why the error occurred.
AEFA-000027 `User exit timed out for Publication Service <service name>`	errorRole	Adapter	Check that the User exit server is running.
AEFA-000028 `User exit failed for Subscription Service <service name>`	errorRole	Adapter	Check the User exit server to determine why the error occurred.
AEFA-000029 `User exit timed out for Subscription Service <service name>`	errorRole	Adapter	Check that the User exit server is running.

Status Code and Status Message	Role	Category	Resolution
AEFA-000030 `Deserialisation failed for Subscription Service <service name>`	errorRole	Configuration	Check if the publisher is sending a MInstance.
AEFA-000033 `No formatting defined for class <class name>`	errorRole	Metadata	Configure the class in your Subscription Service or change its subject address.
AEFA-000034 `Received message couldn't be written to a file`	errorRole	System	Check the file system to determine why the error occurred.
AEFA-000035 `No access to Working Directory <directory name>`	errorRole	Configuration	Create the Working Directory or give access to it.
AEFA-000036 `No access to Completion Directory <directory name>`	errorRole	Configuration	Create the Completion Directory or give access to it.
AEFA-000037 `No access to Error Directory <directory name>`	errorRole	Configuration	Create the Error Directory or give access to it.
AEFA-000038 `Cannot write file <filename> in Working Directory <directory name>`	errorRole	System	Check the file system state to determine why the error occurred.
AEFA-000039 `Cannot create file <filename> in Working Directory <directory name>`	errorRole	System	Check the file system state to determine why the error occurred.
AEFA-000040 `Cannot move file <filename> to Output Directory <directory name>`	errorRole	System	Check the file system state to determine why the error occurred.

Status Code and Status Message	Role	Category	Resolution

AEFA-000041
 	
`Cannot move file`
`<filename> to Error`
`Directory <directory`
`name>`
	errorRole	System	Check the file system state to determine why the error occurred.

AEFA-000042
 	
`Cannot add TimeStamp to`
`file <filename> in`
`directory <directory`
`name>`
	errorRole	System	Check the file system state to determine why the error occurred.

AEFA-000043
 	
`Post Processing Script`
`file directory <directory`
`name> couldn't be found`
	errorRole	Configuration	Check that the directory for the Post Processing Script File of Subscription Service exists and is readable.

AEFA-000044
 	
`Post Processing Script`
`file <filename> couldn't`
`be found`
	errorRole	Configuration	Check that Subscription Service's Post Processing Script File exists and is readable.

AEFA-000045
 	
`Pre Processing Script`
`<command string> returned`
`error code <status`
`string>`
	warnRole	System	Check the Pre Processing script.

AEFA-000046
 	
`Post Processing script`
`<command string> returned`
`error code <status`
`string>`
	warnRole	System	Check Publication Service's Post Processing Script.

AEFA-000047
 	
`<class name> will not be`
`published since no`
`Publication Service could`
`be found`
	warnRole	Configuration	Assign a Read Schema to the File Record.

AEFA-000048
 	
`<input record> couldn't`
`be parsed`
	warnRole	Adapter	Either modify an existing File Record to match this input line or create a new one. Make sure that the File Record is linked to a Read Schema.

Status Code and Status Message	Role	Category	Resolution
AEFA-000049 `File <filename> has been parsed, <number of error records> lines couldn't be interpreted`	warnRole	Adapter	Check if the input line is valid. If it is, then modify the File Record to correctly interpret it.
AEFA-000051 `Subscription Service <service name> received an unexpected message. Type <event type>. Data <event data>.`	warnRole	TibRvComm	Check the format of data sent by other applications on the subject that this Subscription Service is listening on.
AEFA-000052 `Attribute <attribute name> of class <class name> is of an unsupported type`	warnRole	Metadata	Check for any discrepancy between the incoming class and the Write Schema.
AEFA-000053 `Attribute <attribute name> of class <class name> is of an unsupported type`	warnRole	Metadata	Change the input class to make sure that it only includes supported types.
AEFA-000054 `File <filename> already exists in Working Directory <directory name>`	warnRole	Adapter	Remove the file from the Input Directory. Check why the file was still there so the problem does not repeat.
AEFA-000055 `File <filename> has been created in Output Directory <directory name> due to an input message time out`	warnRole	Adapter	If Subscription Service was not supposed to have timed out then you have to check your publisher to find out why the publications were delayed.
AEFA-000056 `Post Processing script <command string> returned error code <status string>`	warnRole	System	Check the Post Processing Script File to make sure that it is valid and correctly returned an error.

Status Code and Status Message	Role	Category	Resolution
AEFA-000057 `<number of messages> messages have been published from file <filename> while <number of messages> have been written`	warnRole	Adapter	Check the content of your input and output data, check your logs for unsent data, and check if only one publisher is sending data.
AEFA-000058 `TIBCO ActiveMatrix Adapter for Files successfully initialized`	infoRole	Configuration	Indicates normal adapter information. No action necessary.
AEFA-000059 `Scanning Input Directory <directory name>`	infoRole	Adapter	Indicates normal adapter information. No action necessary.
AEFA-000060 `Processing input file <filename> in Input Directory <directory name>`	infoRole	Adapter	Indicates normal adapter information. No action necessary.
AEFA-000061 `Pre Processing script <command string> succeeded`	infoRole	Adapter	Indicates normal adapter information. No action necessary.
AEFA-000062 `Execute Post Processing script <command string>`	infoRole	Adapter	Indicates normal adapter information. No action necessary.
AEFA-000063 `Publication of file <filename> is finished.`	infoRole	Adapter	Indicates normal adapter information. No action necessary.
AEFA-000064 `File <filename> has been processed, all lines were interpreted`	infoRole	Adapter	Indicates normal adapter information. No action necessary.

Status Code and Status Message	Role	Category	Resolution
AEFA-000065 `Message containing class <class name> published on subject <subject name> (message is from file <filename>)`	infoRole	Adapter	Indicates normal adapter information. No action necessary.
AEFA-000067 `Message containing class <class name> received on subject <subject name>`	infoRole	Adapter	Indicates normal adapter information. No action necessary.
AEFA-000068 `Message containing class <class name> written to working file <filename in Working Directory <directory name>`	infoRole	Adapter	Indicates normal adapter information. No action necessary.
AEFA-000069 `<number of messages> messages have been published and received from file <filename>`	infoRole	Adapter	Indicates normal adapter information. No action necessary.
AEFA-000070 `File <filename> is moved to the Output Directory <directory name>`	infoRole	Adapter	Indicates normal adapter information. No action necessary.
AEFA-000071 `Post Processing Script <command string> succeeded`	infoRole	Adapter	Indicates normal adapter information. No action necessary.
AEFA-000072 `Multiple publisher mode used`	warnRole	Configuration	Contact TIBCO support if you experience any problem due to migration.
AEFA-000073 `Subscription Service <service name> received an empty MBusinessDocument in which the DataSection attribute was not set.`	infoRole	Adapter	Indicates normal adapter information. No action necessary.

Status Code and Status Message	Role	Category	Resolution
AEFA-000074 `The line "<input record>" contains an invalid field`	errorRole	Adapter	Correct the invalid field.
AEFA-000075 `Could not create simple datetime class. Received error code <error code>`	errorRole	Adapter	Unsupported locale is the main cause for this error.
AEFA-000076 `Failed to parse the datetime string <datetime string> for the pattern specified <datetime pattern>. Received error code <error code>`	errorRole	Adapter	Invalid pattern or locale mismatch is the main cause for this error.
AEFA-000077 `Failed to format the datetime value to the specified pattern <datetime pattern>. Received error code <error code>`	errorRole	Adapter	Invalid pattern or locale mismatch is the main cause for this error.
AEFA-000078 `Failed to set attribute <attribute name> for the class <class name> Received error code <error description>`	warnRole	Adapter	Missing attribute in the class, wrong value or wrong data type is the main cause for this error. The other one being filtering of the attributes of the schema.
AEFA-000079 `File <filename> has been parsed. Total: <number of total records> lines, Error: <number of error records> lines`	infoRole	Adapter	Indicates normal adapter information. No action necessary.
AEFA-000080 `Processing file <filename>...`	infoRole	Adapter	Indicates normal adapter information. No action necessary.

Status Code and Status Message	Role	Category	Resolution

AEFA-000081
 	
`Pre Processing Script`
`<command string> returned`
`message <status string>`
	infoRole	Adapter	Indicates normal adapter information. No action necessary.

AEFA-000082
 	
`Post Processing Script`
`<command string> returned`
`message <status string>`
	infoRole	Adapter	Indicates normal adapter information. No action necessary.

AEFA-000083
 	
`Post Processing Script`
`<command string> returned`
`message <status string>`
	infoRole	Adapter	Indicates normal adapter information. No action necessary.

AEFA-000084
 	
`Skip processing file`
`<filename> at the request`
`of the preprocessing`
`script.`
	infoRole	Adapter	Indicates normal adapter information. No action necessary.

AEFA-000094
 	
`ECM is flex, and times to`
`retry is over. The`
`following subscribers`
`have not responded and`
`will be de-activated.`
	warnRole	Adapter	

AEFA-000095
 	
`Registered Subscriber`
`name = <name string>`
	warnRole	Adapter	

AEFA-000099
 	
`Expected sequence does`
`not match incoming data`
	warnRole	Adapter	

AEFA-000100
 	
`Checksum mismatch with`
`the published file`
	warnRole	Adapter	

AEFA-000102 `IO error detected`	warnRole	Adapter	

Status Code and Status Message	Role	Category	Resolution
AEFA-000106 `Publisher is switching to ECM`	warnRole	Adapter	
AEFA-000107 `The admin confirmation timer is activated.`	warnRole	Adapter	
AEFA-000115 `The admin confirmation timer is activated.`	warnRole	Adapter	
AEFA-000116 `File <filename> is moved to the Error Directory <directory name>`	infoRole	Adapter	Indicates normal adapter information. No action necessary.
AEFA-000117 `Cannot process messages due to version mismatch. Product version = <version number>. Received message version = <version number>`	errorRole	Adapter	
AEFA-000118 `No access to Error Directory <directory name>`	errorRole	Configuration	Create the Error Directory or give access to it.
AEFA-000119 `File transfer failed for the subscriber <name string> because <error description>. Updated subscriber specific error log file <filename>`	errorRole	Configuration	
AEFA-000120 `Unable to open the subscriber specific error log file <filename>`	errorRole	Configuration	

Status Code and Status Message	Role	Category	Resolution
AEFA-000125 `Cannot reset input directory to <directory name>. The directory does not exist or cannot be accessed`	errorRole	Configuration	Create the Input Directory or give access to it.
AEFA-000126 `Input directory is successfully changed to <directory name>`	infoRole	Configuration	Indicates normal adapter information. No action necessary.
AEFA-000127 `filename to process is reset to <filename>`	infoRole	Configuration	Indicates normal adapter information. No action necessary.
AEFA-000128 `File prefix to process is reset to <file prefix>`	infoRole	Configuration	Indicates normal adapter information. No action necessary.
AEFA-000129 `File extension to process is reset to <file extension>`	infoRole	Configuration	Indicates normal adapter information. No action necessary.
AEFA-000132 `Subscriber is started Simple File Transfer mode`	infoRole	Configuration	Indicates normal adapter information. No action necessary.
AEFA-000133 `Publisher is started Simple File Transfer mode`	infoRole	Configuration	Indicates normal adapter information. No action necessary.
AEFA-000134 `The progress file <filename> is corrupt`	errorRole	Configuration	Remove the progress file and restart the adapter.
AEFA-000135 `Received advisory message. Role: <role name>, Subject: <subject name>, Message: <message string>`	errorRole	Adapter	Check the advisory message for cause.

Status Code and Status Message	Role	Category	Resolution
AEFA-000136 `Cannot add sequence number to file <filename> in directory <directory name>`	errorRole	System	Check the file system state to determine why the error occurred.
AEFA-000137 `Polling subscriber endpoint is not defined`	errorRole	Adapter	The file palette generates the polling subscriber endpoint. Name of the endpoint is FAPollingSubscriberServiceName.
AEFA-000138 `Subscriber <subscription service name> is experiencing IO Error. The publisher will deactivate the subscriber for the current file transfer and will reactivate the subscriber on a new file transfer`	errorRole	Adapter	If the IO error is temporary, the subscriber will automatically come back up; however, if it is permanent, you need to stop the subscriber and resolve the IO error, and then restart the subscriber.
AEFA-000139 `Progress filename is empty. Verify configuration and set a valid progress filename`	errorRole	Adapter	Set a valid progress filename using the designer and restart the publisher.
AEFA-000140 `IOError received when accessing progress file. Verify if progress filename is valid`	errorRole	Adapter	Set a valid progress filename using the designer and restart the publisher. Only LATIN_1 encoded filenames are valid.
AEFA-000143 `Cannot open the data file <filename> specified in the progress file on publisher restart`	errorRole	Adapter	Check the existence of the data file, the data file rights, and the content of the progress file.
AEADFILES_910 006 `Exit subscriber is not defined. The Repository URL is <repourl> and the Configuration URL is <configurl>`	warnRole	Startup	

Status Code and Status Message	Role	Category	Resolution

AEADFILES_920 001
 	
`Subscription error.`
`<service name, subject`
`name, repourl, configurl>`
`received an invalid`
`event.`
	errorRole	Subscription	Check the configuration of the application that is publishing the event and make sure that it matches the inbound event definition for the above subscription service. See *TIBCO ActiveMatrix Adapter for Files for Unix/Win (TIBCO Business Studio) User's Guide* on the configuration of Subscription Service.

AEADFILES_920 002
 	
`Subscription error.`
`<service name, subject`
`name, repourl, configurl>`
`failed to deserialize the`
`received event. SDK`
`exception thrown is`
`<error description>`
	errorRole	Subscription	Check the configuration of the application that is publishing the event and make sure that it matches the inbound event definition for the above subscription service. See *TIBCO ActiveMatrix Adapter for Files for Unix/Win TIBCO Business Studio) User's Guide* on the configuration of Subscription Service.

AEADFILES_920 003
 	
`Subscription error.`
`<service name, subject`
`name, repourl, configurl>`
`received inbound event`
`with null data.`
	errorRole	Subscription	Check the configuration of the application that is publishing the event and make sure that it matches the inbound event definition for the above subscription service. See *TIBCO ActiveMatrix Adapter for Files for Unix/Win (TIBCO Business Studio) User's Guide* on the configuration of Subscription Service.

AEADFILES_920 004
 	
`Subscription Service`
`<subscription service`
`name> received an`
`MBusinessDocument`
`<business document> in`
`which the DataSection`
`attribute was not set.`
	infoRole	Subscription	Indicates normal adapter information. No action necessary.

Status Code and Status Message	Role	Category	Resolution
AEADFILES_920 005 `Subscription error. <error message> could not deserialize the inbound event to MBusinessDocument <name of the business document>`	errorRole	Subscription	Check the configuration of the application that is publishing the event and make sure that it matches the inbound event definition for the above subscription service. See *TIBCO ActiveMatrix Adapter for Files for Unix/Win (TIBCO Business Studio) User's Guide* on the configuration of Subscription Service.
AEADFILES_920 007 `Subscription error. Subscription Service <service name> listening on subject <subject name> could not get the class description of <class name>. <repoUrl and configUrl parameters>.`	errorRole	Subscription	Check the repository configuration for this service. See *TIBCO ActiveMatrix Adapter for Files for Unix/Win (TIBCO Business Studio) User's Guide* on how to configure, run and test Subscription Service.
AEADFILES_920 009 `Subscription error. <service name, subject name, repourl, configurl> received event with invalid value <attribute value> for property <attribute name> in class <class name>.`	errorRole	Subscription	Check the configuration of the application that is publishing the event and make sure that it matches the inbound event definition for the above subscription service. See *TIBCO ActiveMatrix Adapter for Files for Unix/Win (TIBCO Business Studio) User's Guide* on the configuration of Subscription Service.
AEADFILES_920 010 `Subscription error. <service name, subject name, repourl, configurl> received event with missing attribute <attribute name> in class <class name>.`	errorRole	Subscription	Check the configuration of the application that is publishing the event and make sure that it matches the inbound event definition for the above subscription service. See *TIBCO ActiveMatrix Adapter for Files for Unix/Win (TIBCO Business Studio) User's Guide* on the configuration of Subscription Service.

Status Code and Status Message	Role	Category	Resolution

AEADFILES_920 011
 	
`Subscription error.`
`<service name, subject`
`name, repourl, configurl>`
`received event with`
`missing association`
`<association name> for`
`class <class name>.`
	errorRole	Subscription	Check the configuration of the application that is publishing the event and make sure that it matches the inbound event definition for the above subscription service. See *TIBCO ActiveMatrix Adapter for Files for Unix/Win (TIBCO Business Studio) User's Guide* on the configuration of Subscription Service.

AEADFILES_920 012
 	
`Subscription error.`
`<service name, subject`
`name, repourl, configurl>`
`received`
`MBusinessDocument`
`<business document name>`
`with NULL value for`
`attribute <attribute`
`name>.`
	errorRole	Subscription	Check the configuration of the application that is publishing the event and make sure that it matches the inbound event definition for the above subscription service. See *TIBCO ActiveMatrix Adapter for Files for Unix/Win (TIBCO Business Studio) User's Guide* on the configuration of Subscription Service.

AEADFILES_920 015
 	
`Subscription error.`
`Subscription Service`
`<service name> listening`
`on subject <subject name>`
`failed due to target`
`application invocation`
`error <error`
`description>. Target`
`application is FILES. The`
`target application`
`specific commands and`
`parameters are <command`
`name>`
	errorRole	Subscription	Make sure that the directory or file exists and the permission is set properly.

AEADFILES_920 020
 	
`Subscription error.`
`<service name, subject`
`name, repourl, configurl>`
`received an event from`
`the wire but encountered`
`error <error description>`
`in pre-processing user`
`exit invocation. The User`
`exit is <userexit client`
`name>. The event details`
`are <class name>.`
	errorRole	Subscription	Make sure that the parameters passed to the UserExit are valid and the User Exit can be invoked by the adapter.

Status Code and Status Message	Role	Category	Resolution
AEADFILES_930 002 `Publication error. Publication Service <service name> with publication subject <subject name> encountered error <error description>.`	errorRole	Publication	Make sure that the directory or file exists and the permission is set correctly.
AEADFILES_930 003 `Publication error. Publication Service <service name> with publishing subject as <subject name> received event from target application FILES. It failed while converting event to MInstance as it could not get the class description for <class name>. <repourl, configurl>.`	errorRole	Publication	Verify the configuration of Publication Service and check that the schema/class definitions are present in the repository. See *TIBCO ActiveMatrix Adapter for Files for Unix/Win (TIBCO Business Studio) User's Guide* on how to configure a Publication Service.
AEADFILES_930 004 `Publication error. <service name, subject name, repourl, configurl> received event from target application FILES. It failed while converting event to MInstance as it could not find property <attribute name> in class <class name>.`	errorRole	Publication	Verify the configuration of Publication Service and check that the schema definitions are present in the repository. See *TIBCO ActiveMatrix Adapter for Files for Unix/Win (TIBCO Business Studio) User's Guide* on how to configure a Publication Service.

Status Code and Status Message	Role	Category	Resolution

AEADFILES_930 005
 	
`Publication error.`
`<service name, subject`
`name, repourl, configurl>`
`received event from`
`target application FILES.`
`It failed while`
`converting event to`
`MInstance as property`
`<attribute name> of class`
`<class name> has invalid`
`value <attribute value>.`
	errorRole	Publication	Verify the configuration of Publication Service and check that the schema definitions are present in the repository. See *TIBCO ActiveMatrix Adapter for Files for Unix/Win (TIBCO Business Studio) User's Guide* on how to configure a Publication Service.

AEADFILES_930 006
 	
`Publication error.`
`<service name, subject`
`name, repourl, configurl>`
`It failed while`
`converting event to`
`MInstance. Attribute`
`<attribute name> of class`
`<class name> is missing.`
	errorRole	Publication	Verify the configuration of Publication Service and check that the schema definitions are present in the repository. See *TIBCO ActiveMatrix Adapter for Files for Unix/Win (TIBCO Business Studio) User's Guide* on how to configure a Publication Service.

AEADFILES_930 007
 	
`Publication error.`
`<service name, subject`
`name, repourl, configurl>`
`received event from`
`target application but`
`could not create the`
`business document`
`<business document name>.`
`The target application`
`details are <class name>.`
	errorRole	Publication	Verify the configuration of Publication Service and check that the schema definitions for the MBusinessDocument maps properly to the event received from the target application. See *TIBCO ActiveMatrix Adapter for Files for Unix/Win (TIBCO Business Studio) User's Guide* on how to configure a Publication Service.

Status Code and Status Message	Role	Category	Resolution

AEADFILES_930 008
 	
`Publication error.`
`<service name, subject`
`name, repourl, configurl>`
`received SDK Exception`
`<error description> while`
`converting the event`
`received from target`
`application to`
`BusinessDocument. The`
`exception occurred while`
`setting the attribute`
`<attribute name>. The`
`target application`
`details are <class name>.`
	errorRole	Publication	Verify the configuration of Publication Service and check that the schema definitions for the MBusinessDocument maps properly to the event received from the target application. See *TIBCO ActiveMatrix Adapter for Files for Unix/Win (TIBCO Business Studio) User's Guide* on how to configure a Publication Service.

AEADFILES_930 009
 	
`Publication error.`
`<service name, subject`
`name, repourl, configurl>`
`received an event from`
`the target application`
`but encountered error`
`<error description> in`
`pre-processing user exit`
`invocation. The User exit`
`is <userexit client`
`name>. The target`
`application details are`
`<class name>.`
	errorRole	Publication	Make sure that the parameters passed to the UserExit are valid and the User Exit can be invoked by the adapter.

AEADFILES_930 014
 	
`Publication error.`
`<service name, subject`
`name, repourl, configurl>`
`received error while`
`sending event over the`
`wire. The Publish`
`endpoint details are`
`<error description>`
	errorRole	Publication	Check the repository settings for a valid configuration of the publish endpoint for this service. See *TIBCO ActiveMatrix Adapter for Files for Unix/Win (TIBCO Business Studio) User's Guide* on setting up a Publication Service and a publish endpoint.

AEADFILES_930 015
 	
`Publication error.`
`<service name, subject`
`name, repourl, configurl>`
`Field count or record`
`length does not match the`
`schema.`
	errorRole	Publication	Correct the record according to the schema.

Status Code and Status Message	Role	Category	Resolution

AEADFILES_930 016
 	
`Publication error.`
`<service name, subject`
`name, repourl, configurl>`
`Publication Service`
`cannot create RegEx`
`matcher for the`
`configured pattern.`
	errorRole	Publication	Check the configured pattern for validity.

AEADFILES_930 017
 	
`Publication Service`
`notification.`
	infoRole	Publication	Indicates normal adapter information. No action necessary.

AEADFILES_980 001
 	
`<function name> Running`
`out of memory when trying`
`to create a new object.`
`Shutting down...`
	errorRole	System	Check the system memory usage.

AEADFILES_990 002
 	
`Shutdown error. SDK`
`cleanup exception =`
`<error description>`
	errorRole	Shutdown	Check the system resource.

AEADFILES_990 003
 	
`Shutdown error. Failed to`
`cleanup Hawk microagent.`
`SDK exception = <error`
`description>`
	errorRole	Shutdown	Check the system resource.

Deprecated Status Messages

The following table lists the deprecated status codes, detailed explanation of each error, where applicable, and ways to solve different errors.

Any use of a deprecated status message is discontinued as it might be removed in a future release. You have to avoid becoming dependent on deprecated status messages.

Status Code and Status Message	Role	Category	Resolution
AEFA-000002 `Out of memory`	errorRole	System	Check the system and find why it ran out of memory.

AEFA-000007
 	
`Couldn't open file`
`<filename>`
	errorRole	System	Check the file system to ensure that all of the files exist and can be opened.

Status Code and Status Message	Role	Category	Resolution
AEFA-000008 `Couldn't close file <filename>`	errorRole	System	Check the file system to find why the file was not closed.
AEFA-000009 `Couldn't read file <filename>`	errorRole	System	Check the file and the file system and find why the error occurred.
AEFA-000011 `Pre Processing Script File directory <directory name> couldn't be found`	errorRole	Configuration	Make sure that the directory exists. If it does, correct the configuration to point to the correct directory for the Pre Processing Script File.
AEFA-000012 `Pre Processing Script File <filename> couldn't be found`	errorRole	Configuration	Make sure that the file exists. If it does, correct the configuration to point to the correct file and directory names for the Pre Processing Script.
AEFA-000013 `Post-processing script file <filename> couldn't be found`	errorRole	Configuration	Correct the configuration to point to the correct file and directory names for the Post Processing Script of the Publication Service.
AEFA-000014 `No access to Input Directory <directory name>`	errorRole	Configuration	Create the Input Directory or give access to it.
AEFA-000015 `No access to Working Directory <directory name>`	errorRole	Configuration	Create the Working Directory or give access to it.
AEFA-000016 `No access to Completion Directory <directory name>`	errorRole	Configuration	Create the Completion Directory or give access to it.
AEFA-000027 `User exit timed out for Publication Service <service name>`	errorRole	Adapter	Check that the User exit server is running.

Status Code and Status Message	Role	Category	Resolution
AEFA-000028 `User exit failed for Subscription Service <service name>`	errorRole	Adapter	Check the User exit server and find why the error occurred.
AEFA-000029 `User exit timed out for Subscription Service <service name>`	errorRole	Adapter	Check that the User exit server is running.
AEFA-000034 `Received message couldn't be written to a file`	errorRole	System	Check the file system to and find why the error occurred.
AEFA-000043 `Post Processing Script file directory <directory name> couldn't be found`	errorRole	Configuration	Check that the directory for the Post Processing Script File of Subscription Service exists and is readable.
AEFA-000047 `<class name> will not be published since no Publication Service could be found`	warnRole	Configuration	Assign a read schema to the file record.
AEFA-000072 `Multiple publisher mode used`	warnRole	Configuration	Contact TIBCO support if you experience any problem due to migration.
AEFA-000077 `Failed to format the datetime value to the specified pattern <datetime pattern>. Received error code <error code>`	errorRole	Adapter	Invalid pattern or locale mismatch is the main cause for this error.
AEFA-000080 `Processing file <filename>...`	infoRole	Adapter	Indicates normal adapter information. No action necessary.
AEFA-000085 `Subscriber name = [%1], subscriber status = [%2]`	debugRole	Adapter	

Status Code and Status Message	Role	Category	Resolution
AEFA-000087 `Simple file transfer is running in ECM mode.`	debugRole	Adapter	
AEFA-000097 `Subscriber is in restart mode and is using [%1] progress file`	debugRole	Adapter	
AEFA-000098 `[%1]: expected sequence [%2] matches incoming data`	debugRole	Adapter	
AEFA-000099 `Expected sequence does not match incoming data`	warnRole	Adapter	
AEFA-000100 `Checksum mismatch with the published file`	warnRole	Adapter	
AEFA-000103 `[%1]: beginning transfer of new file`	debugRole	Adapter	
AEFA-000104 `[%1] received message`	debugRole	Adapter	
AEFA-000118 `No access to Error Directory <directory name>`	errorRole	Configuration	Create the Error Directory or give access to it.
AEFA-000124 `[%1] is discarding duplicate message`	debugRole	Configuration	
AEFA-000130 `The message contains no data field as this may be the last packet for the file being transfered`	debugRole	Configuration	

Status Code and Status Message	Role	Category	Resolution
AEFA-000132 `Subscriber is started Simple File Transfer mode`	infoRole	Configuration	Indicates normal adapter information. No action necessary.
AEFA-000134 `The progress file <filename> is corrupt`	errorRole	Configuration	Remove the progress file and restart the adapter.
AEFA-000138 `Subscriber <subscription service name> is experiencing IO Error. The publisher will deactivate the subscriber for the current file transfer and will reactivate the subscriber on a new file transfer`	errorRole	Adapter	If the IO error is temporary, the subscriber will automatically come back up; however, if it is permanent, you need to stop the subscriber and resolve the IO error, and then restart the subscriber.
AEFA-000139 `Progress filename is empty. Verify configuration and set a valid progress filename`	errorRole	Adapter	Set a valid progress filename using the designer and restart the publisher.
AEFA-000140 `IOError received when accessing progress file. Verify if progress filename is valid`	errorRole	Adapter	Set a valid progress filename using the designer and restart the publisher. Only LATIN_1 encoded filenames are valid.
AEFA-000143 `Cannot open the data file <filename> specified in the progress file on publisher restart`	errorRole	Adapter	Check the existence of the data file, the data file rights, and the content of the progress file.
AEFA-000150 `Processing script [%1] not found.`	warnRole	Adapter	Check the file system and find out the reason.
AEFA-000156 `The trigger message [%s] was invalid and was discarded.`	infoRole	Adapter	

Status Code and Status Message	Role	Category	Resolution

AEADFILES_920010
 	
`Subscription error.`
`<service name, subject`
`name, repourl, configurl>`
`received event with`
`missing attribute`
`<attribute name> in class`
`<class name>.`
	errorRole	Subscription	Check the configuration of the application that is publishing the event and make sure that it matches the inbound event definition for the above subscription service. See *TIBCO ActiveMatrix Adapter for Files for Unix/Win (TIBCO Business Studio) User's Guide* on the configuration of Subscription Service.

AEADFILES_920020
 	
`Subscription error.`
`<service name, subject`
`name, repourl, configurl>`
`received an event from`
`the wire but encountered`
`error <error description>`
`in pre-processing user`
`exit invocation. The User`
`exit is <userexit client`
`name>. The event details`
`are <class name>.`
	errorRole	Subscription	Make sure that the parameters passed to the UserExit are valid and the User Exit can be invoked by the adapter.

AEADFILES_920021
 	
`[%1] set JMS Message`
`Selector \"[%2]\"`
	errorRole	Subscription	

AEADFILES_930004
 	
`Publication error.`
`<service name, subject`
`name, repourl, configurl>`
`received event from`
`target application FILES.`
`It failed while`
`converting event to`
`MInstance as it could not`
`find property <attribute`
`name> in class <class`
`name>.`
	errorRole	Publication	Verify the configuration of Publication Service and check that the schema definitions are present in the repository. See *TIBCO ActiveMatrix Adapter for Files for Unix/Win (TIBCO Business Studio) User's Guide* on how to configure a Publication Service.

Status Code and Status Message	Role	Category	Resolution

AEADFILES_930005
 	
`Publication error.`
`<service name, subject`
`name, repourl, configurl>`
`received event from`
`target application FILES.`
`It failed while`
`converting event to`
`MInstance as property`
`<attribute name> of class`
`<class name> has invalid`
`value <attribute value>.`
	errorRole	Publication	Verify the configuration of Publication Service and check that the schema definitions are present in the repository. See *TIBCO ActiveMatrix Adapter for Files for Unix/Win (TIBCO Business Studio) User's Guide* on how to configure a Publication Service.

AEADFILES_930009
 	
`Publication error.`
`<service name, subject`
`name, repourl, configurl>`
`received an event from`
`the target application`
`but encountered error`
`<error description> in`
`pre-processing user exit`
`invocation. The User exit`
`is <userexit client`
`name>. The target`
`application details are`
`<class name>.`
	errorRole	Publication	Make sure that the parameters passed to the UserExit are valid and the User Exit can be invoked by the adapter.

AEADFILES_930016
 	
`Publication error.`
`<service name, subject`
`name, repourl, configurl>`
`Publication Service`
`cannot create RegEx`
`matcher for the`
`configured pattern.`
	errorRole	Publication	Check the configured pattern for validity.

AEADFILES_980 001
 	
`<function name> Running`
`out of memory when trying`
`to create a new object.`
`Shutting down...`
	errorRole	System	Check the system memory usage.