TIBCO ActiveMatrix® Adapter for Files for Unix/Win

Configuration and Deployment

Software Release 6.2.1
July 2013
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 BusinessWorks, TIBCO Rendezvous, TIBCO Administrator, TIBCO Designer, TIBCO Runtime Agent, TIBCO Hawk, and TIBCO Enterprise Message Service 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.

Copyright © 1999-2013 TIBCO Software Inc. ALL RIGHTS RESERVED.

TIBCO Software Inc. Confidential Information
## Contents

### Chapters

**Chapter 1  Introduction.**  
1. Overview.......................... 2  
   Adapter Features.......................... 2  
2. Configuration.......................... 5  
   Projects.................................. 7  
   Version Control.......................... 8  
3. Deployment.......................... 10  
   TIBCO Administration Domain.............. 10  
   TIBCO Administration Server.............. 11  
   TIBCO Administrator GUI.................. 11  
4. Adapter Project Life Cycle............ 12

**Chapter 2  Getting Started.**  
1. Overview.................................. 16  
2. Configuring the Adapter Components..... 17  
3. Deploying the Runtime Adapter........... 24  
   Deploying the Runtime Adapter Using Adapter Tester.................. 24  
   Deploying the Runtime Adapter from the Command Line.............. 24

### Other TIBCO Documentation

- Related Documentation.................. xiii
- Connecting with TIBCO Resources....... xvi
  - How to Join TIBCOMMunity............... xvi
  - How to Access TIBCO Documentation....... xvi
  - How to Contact TIBCO Support.......... xvi

### Figures

- TIBCO ActiveMatrix Adapter for Files Documentation.......................... xvi
- Deploying the Runtime Adapter from the Command Line.......................... 24

### Tables

- xix

### Preface

- Changes from the Previous Release of This Guide.................. xii
- Related Documentation.......................... xiii
  - TIBCO ActiveMatrix Adapter for Files Documentation.......................... xiii
  - Other TIBCO Product Documentation.......................... xiii
- Typographical Conventions.......................... xiv

### How to Contact TIBCO Support

- How to Join TIBCOMMunity............... xvi
- How to Access TIBCO Documentation....... xvi
- How to Contact TIBCO Support.......... xvi

### How to Access TIBCO Documentation

- How to Access TIBCO Documentation....... xvi
- How to Contact TIBCO Support.......... xvi

### How to Join TIBCOMMunity

- How to Join TIBCOMMunity............... xvi
- How to Access TIBCO Documentation....... xvi
- How to Contact TIBCO Support.......... xvi

### How to Contact TIBCO Support

- How to Contact TIBCO Support.......... xvi
- How to Access TIBCO Documentation....... xvi
- How to Contact TIBCO Support.......... xvi

### Deploying the Runtime Adapter

- Deploying the Runtime Adapter........... 24  
  - Deploying the Runtime Adapter Using Adapter Tester.................. 24  
  - Deploying the Runtime Adapter from the Command Line.............. 24

### Required Platform and Software

- TIBCO Administration GUI.............. 11
- TIBCO Administration Server............ 11
- TIBCO Administration Domain............. 10
- Version Control.......................... 8
- Projects.................................. 7
- Configuration.......................... 5
- Overview.................................. 2

### Overview

- Overview.................................. 2
- Adapter Features.......................... 2

### Adapter Features

- Adapter Features.......................... 2
## Chapter 3 Adapter Configuration Options ......................................................... 27

Overview ............................................................................................................. 28
  Configuration Task Sequence ......................................................................... 28
  Operation Mode .............................................................................................. 28
  Repository ....................................................................................................... 29
  Saving a Project .............................................................................................. 30
  Testing the Adapter ......................................................................................... 31

Adapter Instance Options ................................................................................. 32
  Configuration Tab ............................................................................................ 32
  General Tab ..................................................................................................... 33
  Logging Tab ..................................................................................................... 34
  Startup Tab ...................................................................................................... 35
  Monitoring Tab ................................................................................................. 36

Publication Service Options ............................................................................. 38
  Configuration Tab ............................................................................................ 38
  Processing Tab ................................................................................................ 43
  Schema Tab ..................................................................................................... 47
  SFT Tab ............................................................................................................ 47
  Transport Tab .................................................................................................. 51
  Encoding Tab ................................................................................................... 54
  Advanced Tab .................................................................................................. 56

Subscription Service Options .......................................................................... 60
  Configuration Tab ............................................................................................ 60
  Processing Tab ................................................................................................ 63
  Schema Tab ..................................................................................................... 65
  Transport Tab .................................................................................................. 65
  Encoding Tab ................................................................................................... 68
  SFT Tab ............................................................................................................ 70
  Advanced Tab .................................................................................................. 72

Defining Schemas .............................................................................................. 74
  Defining Read Schemas ................................................................................. 74
  Defining Write Schemas ............................................................................... 81

## Chapter 4 Deploying and Starting an Adapter Using TIBCO Administrator .......... 91

Creating an EAR File in TIBCO Designer ......................................................... 92
Deploying the Project ......................................................................................... 93
Starting or Stopping the Adapter ....................................................................... 94
Monitoring the Adapter ..................................................................................... 95

## Chapter 5 Advanced Topics ................................................................. 97

Using the Adapter with a Revision Control System ........................................ 98
Chapter 6 Monitoring the Adapter Using TIBCO Hawk ................. 135

Overview .................................................. 136
Starting TIBCO Hawk Software ................................ 137
The Auto-discovery Process .................................. 138
Invoking Microagent Methods ................................ 139
Standard and Class Microagents ............................... 142
Custom Microagents ....................................... 145

activateTraceRole() ........................................ 146
defactivateTraceRole() ................................... 147
getActivityStatisticsByOperation(Operation) ............... 148
getActivityStatisticsBySchema(SchemaName) ................. 149
getActivityStatisticsByService(ServiceName) ................. 150
getAdapterServicesInformation() ............................ 151
getComponents() .......................................... 152
getConfig() .............................................. 153
## Figures

| Figure 1 | TIBCO ActiveMatrix Adapter for Files in Operation | 2 |
| Figure 2 | TIBCO Designer Panels | 5 |
| Figure 3 | Create a TIBCO Designer Project | 17 |
| Figure 4 | Define a Delimited File Record | 19 |
| Figure 5 | The Configuration Tab | 20 |
| Figure 6 | Associate the ReadSchema to the Publication Service | 21 |
| Figure 7 | Wire the Write Schema and the Read Schema | 22 |
| Figure 8 | Define a Layout File Record | 23 |
| Figure 9 | Delimited File Record Options | 76 |
| Figure 10 | Layout File Record Options | 85 |
| Figure 11 | Using TIBCO Designer to Specify JMS Message Selector | 126 |
# Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table 1</td>
<td>General Typographical Conventions</td>
<td>xiv</td>
</tr>
<tr>
<td>Table 2</td>
<td>Predefined Global Variables</td>
<td>105</td>
</tr>
<tr>
<td>Table 3</td>
<td>Data Validation Formats</td>
<td>110</td>
</tr>
<tr>
<td>Table 4</td>
<td>Unsupported Variations of the Container Record</td>
<td>113</td>
</tr>
<tr>
<td>Table 5</td>
<td>Date and Time Symbols</td>
<td>115</td>
</tr>
<tr>
<td>Table 6</td>
<td>Country and Language Codes</td>
<td>117</td>
</tr>
<tr>
<td>Table 7</td>
<td>Information for the Status Message</td>
<td>131</td>
</tr>
<tr>
<td>Table 8</td>
<td>Standard and Class Microagent Methods</td>
<td>142</td>
</tr>
<tr>
<td>Table 9</td>
<td>Performance Statistics Methods</td>
<td>145</td>
</tr>
<tr>
<td>Table 10</td>
<td>Tracing Fields</td>
<td>176</td>
</tr>
<tr>
<td>Table 11</td>
<td>Predefined Properties for TIBCO ActiveMatrix Adapter for Files</td>
<td>202</td>
</tr>
</tbody>
</table>
Preface

TIBCO ActiveMatrix Adapter for Files for Unix/Win (TIBCO ActiveMatrix Adapter for Files is still referenced as the short product name hereinafter in this document) processes data from text files and publishes the contents in real-time to the TIBCO environment. The adapter also listens for messages in the TIBCO environment and writes the contents to a file.

Topics

- Changes from the Previous Release of This Guide, page xii
- Related Documentation, page xiii
- Typographical Conventions, page xiv
- Connecting with TIBCO Resources, page xvi
Changes from the Previous Release of This Guide

This section itemizes the major changes from the previous release of this guide.

Support for SFT JMS Map Message Exchange

TIBCO ActiveMatrix Adapter for Files can exchange JMS map messages with mainframe Files adapters in Simple File Transfer (SFT) mode when you set adapter property `adfiles.SFTUseJMSMapMsg` to `ON` in the `adfilesagent.tra` file.

TIBCO ActiveMatrix Adapter for Files is still referenced as the short product name in this document to enhance readability.
Related Documentation

This section lists documentation resources you may find useful.

**TIBCO ActiveMatrix Adapter for Files Documentation**

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

- *TIBCO ActiveMatrix Adapter for Files Concepts*  
  Read this manual to gain an understanding of adapters in general, which you can apply to the various tasks you may undertake.

- *TIBCO ActiveMatrix Adapter for Files Installation*  
  Read this manual to learn how to install TIBCO ActiveMatrix Adapter for Files.

- *TIBCO ActiveMatrix Adapter for Files Configuration and Deployment*  
  This manual explains how to create and configure adapter projects. Information on deploying adapter projects is also included.

- *TIBCO ActiveMatrix Adapter for Files Examples*  
  Read this manual to work through the examples provided with the adapter.

- *TIBCO ActiveMatrix Adapter for Files Release Notes*  
  Read this document for information about new features, deprecated features, and open and closed issues.

**Other TIBCO Product Documentation**

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

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

The following typographical conventions are used in this manual.

Table 1  General Typographical Conventions

<table>
<thead>
<tr>
<th>Convention</th>
<th>Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>TIBCO_HOME</td>
<td>TIBCO products are installed into an installation environment. A product installed into an installation environment does not access components in other installation environments. Incompatible products and multiple instances of the same product must be installed into different installation environments.</td>
</tr>
<tr>
<td>ENV_NAME</td>
<td>An installation environment consists of the following properties:</td>
</tr>
<tr>
<td></td>
<td>• Name  Identifies the installation environment. This name is referenced in documentation as ENV_NAME. On Microsoft Windows, the name is appended to the name of Windows services created by the installer and is a component of the path to the product shortcut in the Windows Start &gt; All Programs menu.</td>
</tr>
<tr>
<td></td>
<td>• Path  The folder into which the product is installed. This folder is referenced in documentation as TIBCO_HOME.</td>
</tr>
<tr>
<td></td>
<td>A TIBCO configuration folder stores configuration data generated by TIBCO products. Configuration data can include sample scripts, session data, configured binaries, logs, and so on. This folder is referenced in documentation as CONFIG_HOME. The default location of the folder is USER_HOME/ENV_NAME/tibco/cfgmgmt/Product_Name. For example, on Windows, the default location is C:\Documents and Settings\UserName\Application Data\ENV_NAME\tibco\cfgmgmt\Product_Name.</td>
</tr>
<tr>
<td>code font</td>
<td>Code font identifies commands, code examples, filenames, pathnames, and output displayed in a command window. For example:</td>
</tr>
<tr>
<td></td>
<td>Use MyCommand to start the foo process.</td>
</tr>
<tr>
<td>bold code font</td>
<td>Bold code font is used in the following ways:</td>
</tr>
<tr>
<td></td>
<td>• In procedures, to indicate what a user types. For example: Type admin.</td>
</tr>
<tr>
<td></td>
<td>• In large code samples, to indicate the parts of the sample that are of particular interest.</td>
</tr>
<tr>
<td></td>
<td>• In command syntax, to indicate the default parameter for a command. For example, if no parameter is specified, MyCommand is enabled:</td>
</tr>
<tr>
<td></td>
<td>MyCommand [enable</td>
</tr>
<tr>
<td>Convention</td>
<td>Use</td>
</tr>
<tr>
<td>------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>italic font</td>
<td>Italic font is used in the following ways:</td>
</tr>
<tr>
<td></td>
<td>• To indicate a document title. For example: See TIBCO ActiveMatrix BusinessWorks Concepts.</td>
</tr>
<tr>
<td></td>
<td>• To introduce new terms. For example: A portal page may contain several portlets. Portlets are mini-applications that run in a portal.</td>
</tr>
<tr>
<td></td>
<td>• To indicate a variable in a command or code syntax that you must replace. For example: MyCommand PathName</td>
</tr>
<tr>
<td>Key combinations</td>
<td>Key names separated by a plus sign indicate keys pressed simultaneously. For example: Ctrl+C.</td>
</tr>
<tr>
<td></td>
<td>Key names separated by a comma and space indicate keys pressed one after the other. For example: Esc, Ctrl+Q.</td>
</tr>
<tr>
<td></td>
<td>The note icon indicates information that is of special interest or importance, for example, an additional action required only in certain circumstances.</td>
</tr>
<tr>
<td></td>
<td>The tip icon indicates an idea that could be useful, for example, a way to apply the information provided in the current section to achieve a specific result.</td>
</tr>
<tr>
<td></td>
<td>The warning icon indicates the potential for a damaging situation, for example, data loss or corruption if certain steps are taken or not taken.</td>
</tr>
</tbody>
</table>
Connecting with TIBCO Resources

How to Join TIBCOmmunity

TIBCOmmunity is an online destination for TIBCO customers, partners, and resident experts, 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 http://www.tibcommunity.com.

How to Access TIBCO Documentation

You can access TIBCO documentation here:

http://docs.tibco.com

How to Contact TIBCO Support

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

- For an overview of TIBCO Support, and information about getting started with TIBCO Support, visit this site:
  
  http://www.tibco.com/services/support

- If you already have a valid maintenance or support contract, visit this site:
  
  https://support.tibco.com

Entry to this site requires a username and password. If you do not have a username, you can request one.
Chapter 1  Introduction

This chapter provides an introduction to TIBCO ActiveMatrix Adapter for Files.

Topics

• Overview, page 2
• Configuration, page 5
• Deployment, page 10
• Adapter Project Life Cycle, page 12
Overview

TIBCO ActiveMatrix Adapter for Files processes data from text files and publishes the contents in real-time to the TIBCO environment. The adapter also listens for messages in the TIBCO environment and writes the contents to a file.

The adapter is compliant with the TIBCO ActiveEnterprise environment, and can be used in conjunction with adapters and products that are compliant with this environment, such as TIBCO ActiveMatrix Adapter for Database, and TIBCO ActiveMatrix Adapter for Siebel. A business process defined in TIBCO ActiveMatrix BusinessWorks allows the adapters to exchange messages.

Figure 1   TIBCO ActiveMatrix Adapter for Files in Operation

Adapter Features

The adapter provides the following features:

- **An Easy-to-use GUI for Configuring the Adapter** The adapter provides its own design-time component, namely, the adapter palette, which seamlessly integrates with TIBCO Designer. This easy-to-use interface allows you to quickly configure adapter-specific features. It also validates some configurations.
• **Support for Defining Schema**  An important palette operation is defining schema. Schemas capture the rules and relations that are used by the runtime component to process the contents of a file during publication or write data into a file during subscription. See Defining Schemas on page 74 for more information.

• **Support for Dual Message Transport**  The adapter supports two popular message transports, TIBCO Rendezvous and JMS.

• **File Recognition**  The adapter provides options to selectively recognize files that need to be processed for publication. See Recognition Method on page 41 for more information.

• **Support for Invoking External Scripts**  The adapter allows you to run external scripts on the files that are being processed. See Processing Tab on page 43 for publication service and Processing Tab on page 63 for subscription service.

• **Restart Capability**  The adapter provides restart capabilities when publishing files. See Checkpoint Restart on page 58 for more information.

• **Performance Tuning**  The adapter provides configuration options for performance tuning. When more than one publication service or subscription service is defined in an adapter configuration, or if a publication service is expected to cater for large files, options are provided to tune the configuration to perform optimally.

• **Flow Control**  The adapter supports flow control so that the publication service can cater to slower consumers. See Advanced Tab on page 56 for more information.

• **Adapter-specific Microagent**  In addition to the standard microagents that are available with any ActiveEnterprise-compliant adapter, the adapter provides its own microagent. This microagent provides methods for getting basic processing-related statistics, as well as methods for getting and setting some configuration parameters at runtime. See Chapter 6 on page 135 for more information.

• **Support for Globalization**  The adapter supports data containing date and time as type date and time. It also provides support for various locales. See Locales Supported for Date and Time on page 117 for more information.

• **Simple Transfer Mode**  If the intention is to merely transfer files across machines, the adapter can be configured to operate in SFT mode. The mode is available for both the TIBCO Rendezvous and JMS transports. See TIBCO ActiveMatrix Adapter for Files Concepts for more information.

• **Enhanced JMS Selective Routing**  This enables you to dynamically send files to selected subscribers by sending a JMS trigger message to the publisher and specifying the JMS message selector for each subscribers. See Using Selective Routing Over JMS on page 125 for more information.
- **ICU Regular Expression File Matching** This enables you to find matching input files by using ICU Regular Expression in filenames. You can use the ICU regular expressions in File Name.

- **Data Field Wrapped In Double Quotes** The publication service treats data fields that are wrapped in a pair of double quotes as one single field. See Delimited File Record Options on page 76 for more information.

- **Additional Checking When Validating A Record** When validating a delimited record, the publication service checks the field count in addition to the constant field value. When validating a positional record, the publication service checks the record length in addition to the constant field value. See Delimited File Record Options on page 76 for more information.

- **Delta Publishing** The publication service checks the input file on a pre-configured time interval, and copies any new data to a work file under the WIP directory, then processes and publishes the data. See Delta Publishing Mode on page 39 for more information.

- **Load Balancing** Multiple instances of the same publication service can be run to process separate files in parallel. See Load Balancing Mode on page 133 for more information.

- **Support for SFT JMS Map Message Exchange** TIBCO ActiveMatrix Adapter for Files can exchange JMS map messages with mainframe Files adapters in SFT mode when you set adapter property `adfiles.SFTUseJMSMapMsg` to `ON` in the `adfilesagent.tra` file. See TIBCO ActiveMatrix Adapter for Files Concepts for more information.
TIBCO Designer, the design-time component, is an easy-to-use graphic user interface. The TIBCO ActiveMatrix Adapter for SAP palette in TIBCO Designer is used to create adapter instances, configure adapter services, download schemas from your SAP system, and save the resulting configuration in a project.

Before using TIBCO Designer, make sure you read the TIBCO Designer product documentation. The following figure shows the TIBCO Designer interface.

**Figure 2   TIBCO Designer Panels**

---

**Project Panel**

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

Projects are the key organizational principle of the configuration information you specify. A *project* is a collection of all configured resources. *Resources* are the components of a project. For example, an adapter publication service is a resource. Resources can be complex and contain other resources, much like a folder can contain other folders on your computer file system. Together, these resources make up your integration project. The top-level (root) folder in the project tree panel represents a project. The top-level folder is initially named *Untitled* and is renamed to the name of the project when you save the project for the first time.
Most adapter resources have context-sensitive help available for the configuration of that resource. Right-click on the resource and choose What Is This? from the popup menu for more information on configuring the resource.

An adapter project contains the following folders:

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

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

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

**Palettes Panel**

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

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

**Design Panel**

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

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

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

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

**Projects**

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

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

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

**Enterprise Archive File**

An Enterprise Archive file contains information about the adapter instances and processes you wish to deploy. The format is used by TIBCO Administrator. The EAR file is imported into Administrator where you can deploy, start, and manage the adapter instance on the machines of your choice.
Local Repository
A project exported to a local repository is saved in .dat format. Projects saved in .dat format should only be used for development and testing. The format can be used where data is not shared by more than one adapter. It is possible to have a few local adapters accessing a local project in read-only mode. It is, however, not possible to have more than one local adapter accessing a local project in read and write mode.

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

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

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

Data are loaded on demand for server-based projects.

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

Version Control
TIBCO Designer allows multiple developers to work on the same project and to use file sharing (locking) or a version control system so that the same resource is not changed by two developers at the same time. Different users can then add resources to the project and lock the parts of the project they are working on.
TIBCO Designer creates a file that can be shared and locked for each top-level resource, such as an adapter configuration. It does not create a file for each resource. As a result, for example, you can lock an adapter configuration but cannot lock individual adapter services.

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

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

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

Using TIBCO Administrator, you can upload the EAR, deploy the adapter on the machine(s) of your choice, and set runtime options before deployment. Additionally, you can start and stop the adapter using TIBCO Administrator.

TIBCO Administrator also provides built-in tools to monitor and manage the adapter.

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

- **User Management**
  
  This module allows you to set permissions for adapter users. You define authentication, users and groups, and assign access control lists to users. This includes security for server-based projects at design-time and for deployed applications at runtime.

- **Resource Management**
  
  This module allows you to monitor machines and all running applications in a TIBCO administration domain. Alerts can be created, for example, to notify an administrator if the number of processes or disk usage exceed a certain number.

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

**TIBCO Administration Domain**

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

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

**TIBCO Administration Server**

The TIBCO Administrator Server provides a central storage and distribution point for configuration data and schema data needed by an adapter. The server is included in both Administrator editions.

Each administration domain has one and only one TIBCO Administration Server. The *TIBCO Administration Server* is the machine process that handles the stored project and requests to manage the TIBCO administration domain.

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

**TIBCO Administrator GUI**

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

This section describes the high-level steps required to configure and deploy an adapter. Each of these steps are described in details in subsequent chapters. Adapter projects are configured using TIBCO Designer.

Configuration

Task A  Define an Adapter Project

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

Task B  Set Global Variables

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

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

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

See Using Global Variables on page 104 for more information.

Task C  Define Schemas

If using the Record Transfer mode, you will define and use schemas. In the Simple File Transfer mode, you can skip this task.

See Defining Schemas on page 74 for more information about defining schemas.

Task D  Configure an Adapter Service

The adapter supports the following services: publication and subscription.
Deployment

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

**Task E  Generate and Import an Enterprise Archive File**

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

See Creating an EAR File in TIBCO Designer for more information.

**Task F  Specify Deployment Information**

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

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

See Deploying the Project for more information.

**Task G  Specify Monitoring Options**

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

- Specifying alerts or TIBCO Hawk rulebases for each machine.
- Specifying alerts and TIBCO Hawk rulebases for an adapter service.

Setting log file properties for an adapter service instance.

**Task H  Start the Adapter**

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

See Starting or Stopping the Adapter for more information.
Chapter 2  Getting Started

This chapter describes the basic steps that are required to configure and run the adapter by presenting an example.

Topics

- Overview, page 16
- Configuring the Adapter Components, page 17
- Deploying the Runtime Adapter, page 24
Overview

In this example, you will start by creating an adapter project and then create two adapter instances. One adapter instance is configured with a publication service and the other is configured with a subscription service. The subscription service listens to the messages that are published by the publication service.

The Record Transfer mode is used in this example. First you need to create and design a ReadSchema and associate the ReadSchema with the publication service. Second create a WriteSchema and wire it up to the ReadSchema. Finally, you need to associate the WriteSchema with the subscription service.

Required Platform and Software

This exercise is performed on a Windows XP operating system, using adapters provided with TIBCO ActiveMatrix Adapter for Files. The exercise is run using TIBCO Rendezvous as the transport.
Configuring the Adapter Components

A typical sequence of creating an adapter project and configuring an adapter instance and related services is as follows:

- Create a TIBCO Designer Project, page 17
- Create an Adapter Instance and Configure It with a Publication Service, page 18
- Create an Adapter Instance and Configure it with a Subscription Service, page 21

Task A Create a TIBCO Designer Project

TIBCO Designer is used to create projects and configure adapter instances. When starting TIBCO Designer, you must first create or select a project.

To create a project, complete the following steps:

1. Start TIBCO Designer.
2. Click the New Empty Project button in the TIBCO Designer startup panel.

Figure 3 Create a TIBCO Designer Project

3. In the Save Project dialog, click the Browse button to select the location of the project, in this example E:\MyProject, then click the OK button. A project is created.
Task B Create an Adapter Instance and Configure It with a Publication Service

To create an adapter instance and configure it with a publication service, complete the following steps:

1. Select the project created in Task A in the Project Explorer.

2. Drag and drop the `FileAdapterConfiguration` icon from the Palettes panel to the Design panel.

3. In the Configuration tab, change the adapter instance name to `PubServivceAdapter` then click the `Apply` button.

4. Create and design a read schema:
   a. In the Project Explorer, expand `PubServivceAdapter`, then select the `FileSchemas` folder.
   b. Drag and drop the `ReadSchema` icon from the Palettes panel to the Design panel.
   c. Double-click the `ReadSchema` icon.
   d. Drag and drop the `DelimitedFileRecord` icon from the Palettes panel to the Design panel.
   e. Double click the `DelimitedFileRecord` icon.
   f. In the Configuration tab, change the Delimited File Record name or use the default name. In this example, change the name to `Order`.
   g. Add attributes by clicking the `Add` button and choose their types. In this example, three attributes are added and their types are `String` as bellow figure shows. Click the `Apply` button to save. See Defining Read Schemas for more details.
5. Expand PubSerivceAdapter tree and select the AdapterServices folder in the Project panel.

6. Drag and drop the PublicationService icon from the Palettes panel to the Design panel.

7. In the Configuration tab, specify the Input Directory, in this example, type E:\data\reader\input. In this example, the Transport Type is Rendezvous, the publication service scans the input directory for files to be processed once every minute. The Record Transfer mode is used and the Recognition Method is By File Name.

8. Specify the filename, in this example, type text.txt in the File Name field as bellow figure shows. See Configuration Tab for detail.
9. In the Processing tab, specify the working directory, in this example as E:\data\reader\wip.

10. In the Processing tab, specify Post Processing as **Leave as is**.

11. Click the **Schema** tab, associate the ReadSchema with the publication service by clicking and then click the Apply Button as Figure 6 shows.
12. Click the **Transport** tab, specify the subject name in the Message Subject. In this example, the subject name is **fa**. This must be the same subject name the adapter listens to.

13. Click the **Apply** Button.

**Task C  Create an Adapter Instance and Configure it with a Subscription Service**

To create an adapter instance and configure it with a subscription service, complete the following steps:

1. Select the project created in **Task A** in the Project Explorer.

2. Drag and drop the **FileAdapterConfiguration** icon from the Palettes panel to the Design panel.

3. Rename the adapter instance to **SubServiceAdapter** and then click the **Apply** button.

   Go to **step 4** to create and define **WriteSchema(Classic)**, or go to **step 5** to create and define **WriteSchema(Manual)**.

4. Create and define **WriteSchema(Classic)**.
To create and define WriteSchema(Classic), complete the following steps:

a. In the Project Explorer, expand SubServiceAdapter, then select the File Schema folder.

b. Drag and drop the WriteSchema(Classic) icon from the Palettes panel to the Design panel.

c. Wire the write schema and the read schema created in step 4 by clicking as Figure 7 shows and then click the Save button and the Apply button.

Figure 7 Wire the Write Schema and the Read Schema


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

a. In the Project Explorer, expand SubServiceAdapter, then select the File Schema folder.

b. Drag and drop the WriteSchema(Manual) icon from the Palettes panel to the Design panel.

c. Double-click the WriteSchema(Manual) icon, and drag and drop the Layout File Record icon from the Palettes panel to the Design panel.

d. In the Configuration tab, change the Writer File Record name or use the default name. In this example, the name is changed to Order.

e. Add attributes by clicking the Add button and choose their types. In this example, three attributes are added and their types are String as Figure 8
shows. Click the **Apply** button to save. See **Defining Write Schemas** for more details.

**Figure 8 Define a Layout File Record**

6. Go back to the **MyProject** tree and select the **Adapter Services** folder.

7. Drag and drop the **Subscription Service** icon from the Palettes panel to the Design panel.

8. Associate the WriteSchema to the subscription service by clicking in the Schema tab.

9. Specify the subject name as **fa** in the Transport tab in this example.

10. Specify the working and output directories in the Processing tab, in this example, E:\data\writer\wip and E:\data\writer\output. See **Processing Tab on page 63** for more details.

11. Give a filename such as **output.txt** in the Configuration tab.

12. Click the **Apply** button.

Two operation modes are available for publication and subscription services. The steps required for configuring the publication or the subscription service depend on the operation mode. See **Operation Mode** for details.
Deploying the Runtime Adapter

This section describes how to deploy the runtime adapter from Adapter Tester and from the command line. In this situation, the two adapter instances that have already been created in the last section will be deployed.

Deploying the Runtime Adapter Using Adapter Tester

The adapter tester is a tool that is used for testing runtime adapters. Follow these steps to deploy the two adapter instances.

1. From the Tools menu, select Show Adapter Tester.
2. Specify the working directories for the two adapter instances in the Run Settings tab separately, as in this example they are the same E:\temp.
3. Start the two adapter instances.
4. You can see the progress of the publication and subscription services from the console.

Deploying the Runtime Adapter from the Command Line

1. Convert the project to a .dat file.
   a. From the Project menu, select Export Full Project.
   b. Browse and select the location of the directory where you want to save the .dat file in the Export Project dialog, for example e:\temp.
   c. Enter the name for the .dat file in the Project Name field in the Export Project dialog, for example, test.
   d. Click the OK button.
   e. The test.dat file will be generated and saved in e:\temp.
2. Create a TIBCO Runtime Agent file.
   The TIBCO Runtime Agent file is a runtime configuration file with the tra suffix.
   The installation program generates the TIBCO Runtime Agent file for the TIBCO ActiveMatrix Adapter for Files, and is called the adfilesagent.tra. On Windows, if the TIBCO Runtime Agent file generated by the installation program is copied and used by the runtime adapter (with the appropriate modifications) it will set the required environment for the adapter to run. On Unix however, some environment variables need to be set before running the adapter.
You need to make a copy of the .tra file and update the related variables.

a. Navigate to the $TIBCO_HOME\tibco\adapter\version_num\bin\ directory. Make a copy of the adfilesagent.tra file, and rename it, for example test.tra.

b. Update the related variables.

At a minimum, the following must be provided:

- tibco.repourl—pathname of the TIBCO Designer project
- tibco.configurl—name of the adapter configuration
- application.args—properties file to pass to the application

In this example, if you want to deploy the adapter instance that is configured with the subscription service, update the variables as follows:

tibco.repourl E:/temp/test.dat
tibco.configurl FileAdapterConfiguration1
application.args adfilesagent -system:propFile %TIB_ADFILES_HOME%/bin/test.tra

On Windows, if the TIBCO Runtime Agent file that is created by the installation program is used as the template it will set the environment required for the adapter to run.

On UNIX, while most of the environment is set by the generated TIBCO Runtime Agent, some of the environment variables need to be set before running the adapter.

The installation program generates the adfilesagent_env.csh (or adfilesagent_env.sh) under the File Adapter home/bin to set the environment variables that cannot be set by the TIBCO Runtime Agent, and to facilitate TIBCO Administrator to deploy the adapter on UNIX platforms. When running the adapter from the command line, the TIBCO Runtime Agent file needs to be sourced.

3. Deploy the adapter instance from the command line.

   a. Navigate to the $TIBCO_HOME\tibco\adapter\version_num\bin\ directory in a command line window.

   b. Type adfilesagent.exe --propFile filename.tra to start the audiotapes instance.

4. Repeat step 1 through step 3 to deploy the adapter instance that is configured with the publication instance.
Chapter 3  

**Adapter Configuration Options**

This chapter explains how to create an adapter instance and add services by configuring standard settings. All configuration tasks are performed in TIBCO Designer and the information is stored in a project that will be used later by the runtime adapter.

**Topics**

- Overview, page 28
- Adapter Instance Options, page 32
- Publication Service Options, page 38
- Subscription Service Options, page 60
- Defining Schemas, page 74
Overview

Read the following sections before starting to configure an adapter.

- Configuration Task Sequence on page 28
- Operation Mode on page 28
- Repository on page 29
- Saving a Project on page 30
- Testing the Adapter on page 31

Configuration Task Sequence

A typical sequence of creating and configuring an adapter is as follows:

1. Create a project using TIBCO Designer. Alternatively, you can work with an existing project. See the TIBCO Designer documentation for details.

2. Drag and drop the Files Adapter Configuration icon from the Palettes panel to the Design panel. This creates an adapter instance named, by default, FileAdapterConfiguration.

3. Configure the adapter instance. See Adapter Instance Options on page 32.

4. Add one or more services to the adapter instance by dragging a service icon from the Palettes panel and dropping it in the Design panel. See Publication Service Options on page 38 and Subscription Service Options on page 60 for details.

5. Define read and write schemas if using Record Transfer Mode. See Defining Schemas on page 74 for details.

6. Save the project.

Operation Mode

The Record Transfer and Simple File Transfer modes are available for publication and subscription services. The steps required for configuring the publication or the subscription service depend on the operation mode.

In the Record Transfer mode, you will define and use schemas.

In the Simple File Transfer mode, you start to configure a publication or subscription service by setting the desired file transfer configuration in the SFT tab and then define various options for file transfer. However, there is no need to define a schema.
See *TIBCO ActiveMatrix Adapter for Files Concepts* for details about the concept of operation mode.

**Repository**

You either can use an existing repository to configure an adapter or create a new repository. See *Projects* for details about repository.

**Using an Existing Repository to Configure an Adapter**

1. Start TIBCO Designer and select the **Administration** tab. Click **Convert DAT to Files**.

   The Convert DAT To File-Based Project dialog is displayed.

2. Enter the required information and click the **OK** button. TIBCO Designer converts the DAT file to a project.

3. Select the **Project** tab and click **Open Existing Project**. The Open Project dialog is displayed.

4. Browse to the location you specified in **step 2** and select the project. Click the **OK** button.
Creating a New Repository

1. Start TIBCO Designer and click **New empty project**. This opens a new repository.

2. Select **Project > Save**. The Save Project dialog is displayed.

3. Enter the appropriate information and click the **OK** button.

You have now successfully created a repository.

The runtime adapter can work with a local or remote repository. However, if you chose the Multi-File repository during design-time, it should be converted into either local or remote.

Saving a Project

Configuration information for an adapter and all other parameter settings related to the adapter are saved as a project. You can save the associated project at any time while configuring the adapter. Each time you save a project, any configuration information you have entered is saved as a project.

For detailed steps and information about exporting or importing projects to different formats (such as DAT), see *TIBCO Designer User’s Guide*. 
Testing the Adapter

You can use the Adapter Tester to verify that an adapter instance is configured correctly. The tester is invoked from the TIBCO Designer Tools menu and is documented in *TIBCO Designer Palette Reference*. 
Adapter Instance Options

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

- Configuration Tab on page 32
- General Tab on page 33
- Logging Tab on page 34
- Startup Tab on page 35
- Monitoring Tab on page 36

Configuration Tab

Instance Name

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

- An instance name must use alphanumeric characters. An underscore (_) character can be used. The entire instance name must be less than 80 characters. The space character cannot be used in an instance name.
- An instance name cannot use global variables.
- An instance name must be unique with respect to other adapter instances for the same adapter in the project. The same instance name can be used to name an adapter instance for a different adapter in the same project. For example, an R/3 adapter instance named TEST and a Siebel adapter instance named TEST can coexist in the same project.
- Each instance name must be unique per adapter within a project even if each instance is defined in a different folder. That is, configuring same-named adapter instances in different folders will not make their names unique.

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

Description

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

The version string indicates the ActiveEnterprise (AE) format in which the adapter instance is saved. An adapter instance can be saved in AE 4.0 or AE 5.0 format.

When a new adapter instance is created in TIBCO Designer 5.x, the version string is set to AE Version 5.0. When a 4.x adapter instance is opened in Designer 5.x, the Version field is set to AE Version 4.0.

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

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

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

  To change versions, click the Change Version button.

Message Filter

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

Show All Tabs

Check this checkbox to display the General tab for configuring advanced options.

General Tab

This tab displays only when the Show All Tabs button is selected in the Configuration tab.
Termination Subject or Topic

A message sent on the termination subject (if TIBCO Rendezvous is the transport) or topic (if JMS is the transport) stops the adapter. In most cases, you should use the default value.


Logging Tab

Use Advanced Logging

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

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

To create and configure the sinks, perform these steps:

1. Select the Log Sinks folder under the Advanced folder in the Project panel.
2. Drag and drop the Generic log sink icon from the Palettes panel to the Design panel.
3. From the Configuration panel, select the sink type. The following are the sink types available:
   - File—You can specify the file limit, file count, and the option to append or overwrite. By default, this is set to 30,000 bytes, 3, and append mode respectively.
   - Hawk—The Hawk sink uses the hawk session, created and used by the adapter for monitoring purposes, to send tracing messages to the TIBCO Hawk monitor or Display. The configuration for the Hawk sink involves specifying the MicroAgent Name that must be specified in the Configuration panel.
   - Network—Publishes tracing messages on TIBCO Rendezvous. The configuration for the network sink involves specifying the session and the subject on which the trace messages need to be published.
   - STDIO—You have an option to write to stdout or stderr. The default is stdout.

When File and STDIO sinks are created from Generic log sink, they offer further configuration options.
For all sinks, the name and description fields are optional.

**Log to Standard I/O**

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

**Log File**

Specify the name of the log file (log sink) to which trace messages are written. Global variables can be used to specify the location of the log file. See Using Global Variables on page 104 for more information.

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

**Log Info/Debug/Warning/Error Messages**

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

**Startup Tab**

**Show Startup Banner**

The startup banner displays the runtime adapter version, the infrastructure version on which the adapter is built, and copyright information in the console window when the adapter is started.

**Metadata Search URL**

This field is not used by the adapter.
Monitoring Tab

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

Enable Standard Microagent

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

Standard Microagent Name

This is the name for the standard microagent that will be registered with the TIBCO Hawk system. In most cases the default value is used. The InstanceId variable need not be set because it is automatically set at runtime by the runtime adapter.

Standard MicroAgent Timeout(ms):

The default value is 1000 milliseconds. TIBCO Hawk waits the feedbacks from the standard MicroAgent for the time that you specified.

Enable Class Microagent

Allows you to turn on or off the instance or class specific standard TIBCO Hawk Microagent.

Class Microagent Name

This is the name for the class microagent that will be registered with the TIBCO Hawk system. In most cases the default value is used. You can also configure the way to turn it on or off. By clicking the Globe icon, a standard checkbox or text value (true or false) can be used to turn the class microagent on or off.

Class Microagent Timeout(ms):

The default value is 1000 milliseconds. TIBCO Hawk waits for the feedbacks from the class MicroAgent for the amount of time that you specified.

Default Microagent Session

Specify the name of the TIBCO Rendezvous session that will be used by the standard, class, and custom microagents.
The session name and the corresponding session is automatically generated by TIBCO Designer. However, you can modify the session parameters if required. Navigate to the Sessions folder under the Advanced folder to modify the session parameters.

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

The following tabs can be used to define a publication service:

- Configuration Tab on page 38
- Processing Tab on page 43
- Schema Tab on page 47
- SFT Tab on page 47
- Transport Tab on page 51
- Encoding Tab on page 54
- Advanced Tab on page 56

Configuration Tab

This tab is available for both the Record Transfer mode and the Simple File Transfer mode of the adapter.

Name

The name you want this publication service to use. The name should be unique among all publishers in this adapter configuration. The name can contain only alphanumeric characters, including the underscore (_) character, and can be at most 80 characters long. The space character cannot be used in a name. Configuration names cannot use global variables.

Description

The description for the publication service that you are configuring. This field is optional.

Transport Type

Select the transport to be used by the runtime adapter, JMS or TIBCO Rendezvous. After selecting the transport, the transport-specific configuration fields display on the Transport Tab.

When JMS is used, the adfiles.JMSSequence property can be set to ON to enable JMS transport level compression in the adapter TRA file. See Appendix B, Adapter Properties File, on page 199 for more information.
The transport can be configured to use a trusted store and identity resource for use in SSL (Secure Sockets Layer) configurations. TIBCO Rendezvous sessions and JMS topics have an SSL configuration field which uses a dialog to perform SSL configuration.

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

**Transfer Mode**

This determines the operation for the publication service. Select **Record Transfer** or **Simple File Transfer** from the drop-down list. See *TIBCO ActiveMatrix Adapter for Files Concepts* for details about transfer mode.

**Preserve Undelivered**

This checkbox determines whether an undelivered JMS message should be preserved in the system queue.

**Delta Publishing Mode**

Delta Publishing mode only supports the Timer polling method. With Delta Publishing enabled, the publication service checks the input file on a pre-configured timer interval, and copies any new data to a work file under the WIP directory, then processes and publishes the new data.

When this checkbox is checked, several fields will be grayed out, as shown in the following table.

<table>
<thead>
<tr>
<th>Field/Checkbox</th>
<th>Read-only?</th>
<th>Default Value/Status</th>
<th>Tab</th>
</tr>
</thead>
<tbody>
<tr>
<td>Life Cycle</td>
<td>Yes</td>
<td>Repeating</td>
<td>Configuration</td>
</tr>
<tr>
<td>Polling Method</td>
<td>Yes</td>
<td>Timer</td>
<td>Configuration</td>
</tr>
<tr>
<td>Recognition Method</td>
<td>Yes</td>
<td>By file name</td>
<td>Configuration</td>
</tr>
<tr>
<td>Pre Processing Script File</td>
<td>Yes</td>
<td>None</td>
<td>Processing</td>
</tr>
<tr>
<td>Pre Processing Arguments</td>
<td>Yes</td>
<td>None</td>
<td>Processing</td>
</tr>
<tr>
<td>Post Processing</td>
<td>Yes</td>
<td>Leave as is</td>
<td>Processing</td>
</tr>
</tbody>
</table>
Delta Publishing mode supports the CheckPoint Restart feature, and the Document Delay feature.

The following two properties in the TRA file can be set for Delta Publishing mode:

- adfiles.DeltaFlushInterval
- adfiles.DeltaTrimInterval

If you want to stop Delta Publishing a file, you need to manually delete, or rename the file.

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

### Life Cycle

Specifies if the publication service needs to operate continuously (Repeating) or just one time (Once-only). Once-only means to start, process the files, and exit).

If more than one services is defined for the adapter instance, the first service cannot use Once-only.

### Polling Method

When selecting Repeating from the Life Cycle drop-down list, the polling method specifies how the publication service will be triggered to start processing the files. The publication service can be triggered using a timer or a message. The Timer method of triggering is available for both the transport types.

Triggering using a message depends on the transport type.
If the transport type is selected as TIBCO Rendezvous, then a TIBCO Rendezvous message can be used to trigger the publication service. If the transport type is selected as JMS, then a JMS message sent on a topic can be used to trigger the publication service.

**Polling Interval**

The amount of time in seconds until the next file scan is repeated. This is active when selecting Timer in the Polling Method drop-down list.

**Polling Subject or Polling Destination**

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. It’s the subject or topic name on which the TIBCO Rendezvous message or the JMS message is sent to trigger the publication service.

In the Simple File Transfer mode, special types of messages are defined. For the syntax and semantics of the special message, see Using Trigger Messages on page 122.

**Input Directory**

The publication service searches and processes the files in this directory, and then publishes the files.

This directory must be different from the directories specified for the Working Directory and Completion Directory fields. 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, the working, and the done or output directories need to be specified on the same file system.

Only the input directory is scanned for files that match the criteria. It is not recursively traversed.

**Recognition Method**

Specify the mechanism for finding the desired input file(s). This can be done using one of the following methods:

- **By File Name**—Process the file that exactly matches the value given in the File Name field.
• **By Wildcard via ICU Regular Expressions**—Process the file that matches the ICU regular expression specified in the File Name field.

• **By Prefix + Extension**—Process the files that match the criteria that you defined in the File Prefix and File Extension fields.

• **By Trigger**—Process the files that match the criteria that you have defined in the File Prefix, File Extension, and Trigger File Extension fields.

  This allows the adapter to process the input files only after they are ready. Without this, the adapter may process the files in the input directory before the third-party applications have created, written, and closed the files. The trigger file recognition method helps to avoid this situations. With the trigger recognition method, the adapter will only process the input files after the trigger files are created.

  The filename or file prefix cannot contain path information.

  For details with examples of the recognition method, see File Recognition Methods for Publication Service on page 108.

**File Name**

This field is enabled in the following cases:

• The By File Name item is selected in the Recognition Method drop-down list. In this case, the adapter processes the file that exactly matches the value given in this field.

• ICU regular expressions can be used in the File Name field when the By Wildcard via ICU Regular Expressions item is selected in the Recognition Method drop-down list.
The following are examples of using ICU regular expressions.

— Prepare 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` will be published.

— Prepare following files in input directory: `A6\.[01]\.[0-1]\.txt`, `A6\.[01]\.[0-1]\.log`, and `A6\.[01]\.[0-1]\.log`. If the input filename is `A6\.[01]\.[0-1]\.\(txt\|log\)`, the input files named `A6\.[01]\.[0-1]\.txt`, `A6\.[01]\.[0-1]\.log`, and `A6\.[01]\.[0-1]\.log` will be published.

Wildcard, which is different from regular expressions, is not supported. For example, `*\.
`.txt must be specified as `.*\.
txt` as in regular expressions format.

**File Prefix**

This prefix is used to locate the input file in the input directory. Any files matching the specified criteria will be processed. To activate the File Prefix, select **By prefix + extension** or **By trigger** in the Recognition Method drop-down list.

**File Extension**

To activate the File Extension, select **By prefix + extension** or **By trigger** in the Recognition Method drop-down list.

**Trigger File Extension**

To activate the Trigger File Extension, select **By trigger** in the Recognition Method drop-down list.

**Processing Tab**

This tab is available for both the Record Transfer Mode and the Simple File Transfer mode.

**Pre Processing Script File**

Name of the script that needs to be executed before the input file can be processed by the adapter. This allows you to perform processing on the input file before it is processed by the adapter. Click **Browse** to locate and load the script file.
Pre Processing Arguments

Arguments that need to be passed to the pre-processing script file. These are strings and are optional.

The command line syntax of the arguments that will be passed to the script is:

```
Script_filename Pre Processing Arguments
```

For example,

```
script.tcl inputFile0364.txt argument2 argument3 ...
```

where

```
script.tcl is the script filename, inputFile0364.txt is the name of the file that must be pre-processed and is the first argument. It is followed by other arguments.
```

The pre-processing script file that you use should read the input file, rename it, make the required modifications, and write to the original filename.

If there are five files in the input directory, the adapter will run 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 the number and alphabet of their names in ascending order. It’s also case sensitive. 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 will process the files in the following order:

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

During pre-processing, if the pre processing script finds the file unsuitable for processing, the adapter will not process the file. The adapter will log the feedbacks from the pre processing script.
**Pre and Post Processing Scripts**

Pre and post processing scripts have a way to communicate a processing status message back to the runtime adapter. The runtime adapter will write the processing status message to the log (Info role, AEFA-000081 for preprocessing status message and AEFA-000082 for postprocessing status message).

Additionally, the pre-processing script can inform the runtime adapter to skip processing the input file, by appending the string "Skip the input file" to the status message. The runtime adapter will then skip the input file and proceed to process the next file. If the error directory is specified, the runtime adapter will move the skipped file to the error directory.

To turn on this feature, the string %ScriptStatusFile% should be specified as the last argument in the Pre Processing or Post Processing Argument field.

At run time, the adapter will automatically substitute a valid temporary file name to this argument.

The script then will write the status message, if any, into this file.

The adapter will read the temporary file, and log the status message to the registered sinks.

**Working Directory**

The publication service uses this directory to process the files that match the criteria. Based on the Post Processing option that is selected, the file is either copied or moved into this directory.

If you select **Leave as is** in the Post Processing drop-down list, the file is copied. If you select **Delete** or **Move to**, the file is deleted or moved.

For adapter instances, if the files processed by the publication service(s) are independent of each other, the publication service(s) can share the Input, Working, Completion, and Error directories. Otherwise, the values for these directories must be unique.

**Completion Directory**

This is active only when you have selected **Move to** in the Post Processing drop-down list. After the file in the working directory is processed, it is moved to this directory.
Error Directory

When the transfer mode is Simple File Transfer, this field is mandatory. When the transfer mode is Record Transfer, this field is not mandatory. However, if specified, an XML file containing the name of the input file and the error details will be created in this directory. For details about the usage and contents of this directory, see Error Handling on page 100 for details.

Progress Directory

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

Post Processing

Specifies an action to apply to the file that is currently in the working directory after the publication service has processed the file. Post-processing actions can be one of the following:

- **Move to** — Move the file from the working directory to the completion directory.
- **Delete** — Delete the file from the working directory.
- **Leave as is** — Delete the file from the working directory (since the file in the working directory is a copy. The corresponding file in the input directory is left as is).

Add TimeStamp to File Name

This is an option to append date and time to the file that is moved to the completion directory. The format of the date and time is `YYYYMMDDHHMMSSmm`.

Post Processing Script File

Specify the name of the script that needs to be executed after the input file is processed by the adapter. Click the **Browse** button to locate and load the script.

Post Processing Arguments

These are arguments you want to pass to the post-processing script. These are strings and are optional. The sequence of the arguments passed to the post processing script are determined as follows:

- If the Transfer Mode is Record Transfer, the argument sequence will contain the name of the file, the arguments specified in the post processing arguments,
and then the status. The status will be succeeded if the publisher processes the file successfully. The status will be failed if the publisher has problems (for example, parsing) with processing the file.

- If the Transfer Mode is Simple File Transfer, and Explicit Confirmation Mode (ECM) is not selected in the SFT tab, the argument sequence will contain the name of the file and be followed by the arguments specified in the post processing arguments.

- If the Transfer Mode is set to Simple File Transfer, and Explicit Confirmation Mode (ECM) is selected in the SFT tab, the argument sequence will contain the name of the file, the arguments specified in the post processing arguments, and the status flag. The status will be succeeded if all the registered subscribers have successfully received the file transfer. The status will be failed if one or more registered subscribers have not successfully received the file transfer.

### Schema Tab

This tab is available only for the Record Transfer Mode of the adapter. Associate the ReadSchema(s) that are created for the publication service by clicking the Add Schema button and selecting the read schema from the pop-up dialog box. Repeat this to associate more than one ReadSchema.

The option for filtering the fields in the file records can be specified here. This is done by expanding the ReadSchema tree and the subsequent file record and deselecting the fields from the file records. A checkbox under the Use? column is provided for each field for this purpose.

When the fields are filtered, it will alter the wire schema.

Since TIBCO ActiveEnterprise applications use wire schema to exchange data, it is a good practice to inform the other adapter users about the change.

### SFT Tab

This tab is available only for the Simple File Transfer mode.

**Explicit Confirmation Mode (ECM)**

If checked, only the TIBCO Rendezvous reliable messaging quality of service is available. If ECM is not checked, files are transferred using TIBCO Rendezvous reliable or certified messaging quality of service.
If using ECM,

- On starting, the publication service, irrespective of the mode (strict ECM or flex ECM), performs a discovery of the preregistered subscription services. Discovery is performed to ascertain the status of the registered subscription services.

- When operating in strict ECM, discovery is performed until all the registered subscription services have responded and their state is marked as active.

- When operating in flex ECM, discovery is performed for the specified number of times during the specified intervals. For the registered subscription services that have not responded during such time, the publisher marks their status as inactive.

- Discovery is not performed during a restart. The ECM protocol does not support dynamic registration of subscription services. This means it does not support adding new subscription services at runtime.

- On restart, the publication service uses the progress file (the name and location of the progress file can be configured during design time) that was created when it was started for the first time. It uses the progress file as the reference for the list of subscription services and for getting their status. Removing the progress file before restarting the publication service has the same effect as starting the publication service for the first time.

- When operating in flex ECM, one or more registered subscription services can become inactive or may not have been started. An inactive pre-registered subscription service can request an activation. The publication service will honor the request and will turn the inactive subscription service to an active one when it begins a new file transfer.

- When operating in flex ECM, if the publication service determines that all preregistered subscription services are inactive, it switches to non-ECM. The service continues to operate in non-ECM until a preregistered subscription service becomes active.

- If a subscription service experiences an IO error during the transfer of a particular file, the publication service, irrespective of operating in Strict
ECM or Flex ECM, will mark the subscription service as inactive and continue with the transfer of that particular file.

If the publication service is operating in Strict ECM, it will activate all inactive subscription services before beginning transfer of the next file.

If the publication service is operating in Flex ECM, it will activate the inactive subscription service when that particular subscription service requests activation.

— When operating in ECM, if you want to transfer a file between two subnets over TIBCO Rendezvous routing daemon, the ECM protocol administration messages between the publication and subscription services are exchanged with the following subject names:

```
endpoint subject name.>
_TIBCO_AE_ADAPTER_FAFT_PUBTOSUB . endpoint subject name.>
_TIBCO_AE_ADAPTER_FAFT_SUBTOPUB . endpoint subject name.>
```

- If using non-ECM, there is no notion of registered subscription services and hence there is no discovery or waiting for acknowledgements. Features such as checksum verification, IO error issues at the subscription service, and error handling are not supported in this mode.

However the publication service provides an option for restarting from where it left off just before exiting normally or abnormally. This can be specified at configuration time as the progress file parameter.

On starting, the publication service scans the input directory and begins transferring files that match the file recognition criteria.

Upon restart, if a progress file is set, the publication service will continue from where it left off. Otherwise, it will scan the input directory and begin transferring files.

**Confirmation Timeout Period**

The amount of time in seconds that the publication service waits for:

- discovery confirmation from its registered subscribers
- confirmation from registered subscribers that each file chunk transfer was successful

**ECM Retry Semantics**

Active only when the ECM is selected.
**Times To Retry**

This field is available only if you select **Flex ECM** in the ECM Retry Semantics drop-down list. This is the number of times to retry sending either administrative messages or data messages before marking a registered subscription service as inactive.

**Pre-registered ECM Subscribers**

It is active only when Explicit Confirmation Mode is selected. A comma separates the list of subscription service names that are configured to receive file transfers from this publication service. Each subscription service operating in the ECM has a name.

**Transmission Buffer Size**

The buffer size to use for each data message chunk.

**Progress File Name**

If using the ECM, this parameter is mandatory. If not using ECM, this parameter is optional.

If the progress filename is left blank, the publication service will not restart from where it left off. If the progress filename is specified, the publication service will continue to process from where it left off.

The default progress filename is:

```
__InstanceNameServiceName.prg
```

If a path name is not specified as part of the progress filename, the progress file is created under the directory specified in the **Progress Directory** field of the Processing tab.

The progress filename can be specified using only the LATIN_1 or ASCII character set.

**File Transfer Mode**

Choose any one of the following methods for transferring files:

- **Binary** — This mode is used for opaquely transferring files and is used for any type of file where the file contents must not be altered during file transfer.

- **Text** — This mode is used for transferring text files. When transferring files across platforms, the file created on the receiving platform may 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.

Checksum Verification

Checksum is active only if you select the ECM checkbox and the Binary mode. If checksum verification is checked, each registered subscription service will compute the checksum at the end of the file transfer and verify it against the checksum computed by the publication service.

If there is a checksum mismatch, the subscription service notifies the publication service and appropriate error handling takes place.

Checksum computation is a time-consuming activity and selecting it will affect the throughput.

Retransmission Delay

Specify the number of times the publisher will attempt to publish a message before publishing an error message in this field.

The default value is 10.

Transport Tab

This tab is available for both the Record Transfer Mode and the Simple File Transfer mode of the adapter.

Message Subject (TIBCO Rendezvous Transport Only)

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

Reply Message Subject (TIBCO Rendezvous Transport Only)

Not currently used.

Quality of Service (TIBCO Rendezvous Transport Only)

Any of the following choices can be specified:

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

- **Reliable** (Reliable Message Delivery) ensures that each multicast or broadcast message is received as long as the physical network and packet recipients are working, and that the loss of a message is detected. This choice can compensate for brief network failures because it can retransmit a message on request if the first attempt failed. This choice is appropriate when message delivery is expected but some loss can be tolerated. When this quality of service is chosen, a Rendezvous session will be used.

### Wire Format

Choose one of the following wire formats for sending the message. Publishers and subscribers can only send and receive data if they agree on a specific wire format.

This field is only enabled in the Record Transfer mode.

In the Simple File Transfer mode, this field is read-only. Its value is Rendezvous Message by default for the Rendezvous transport type and XML Message by default for the JMS transport type.

In the Record Transfer mode, you can select **Rendezvous Message**, **ActiveEnterprise Message**, or **XML Message** for the Rendezvous transport type, and you can only select **XML Message** for the JMS transport type.

- **Rendezvous Message** — Control information for validation is *not* sent in the message. For TIBCO Rendezvous wire format, a message of type `RVMSG_RVMSG` (TIBCO Rendezvous 5.x) or `TIBRVMSG_MSG` (TIBCO Rendezvous 6.x) is always sent when this format is selected. If a publisher sends a simple `RVMSG_STRING` or `TIBRVMSG_STRING`, an exception is passed to the subscriber.

  This option is not available for the Record Transfer Mode of the adapter. It is available only for the Simple File Transfer mode.

- **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. This format allows ActiveEnterprise components to perform extra validation on messages sent or received.

  This option is not available for the Simple File Transfer mode of the adapter. It is available only for the 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.

See *TIBCO ActiveMatrix Adapter SDK Concepts* for details about the control information generated and sent with ActiveEnterprise messages.

**Destination (JMS Transport Only)**

By default, a service uses a dynamic destination that is generated using the Domain and Deployment global variables, the adapter acronym, the adapter instance name, and the service name. If you use this default dynamic destination, make sure the values for Domain and Deployment are not empty. You can override the default dynamic destination by specifying the static destination in this field. The static destination must be defined on the JMS server before it can be used by the runtime adapter.

**Connection Factory Type (JMS Transport Only)**

A message can be published to a *topic* or sent to a *queue*.

- **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 known as 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 may connect to the queue. The first receiver to access the queue gets the message. The other receivers do not. This messaging model is known as point-to-point.

**Delivery Mode (JMS Transport Only)**

For a publication service, a message is marked as *persistent* or *non persistent*.

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

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

Messages with the persistent delivery mode are always written to persistent storage, except when they are published to a topic that has no durable subscribers. When a topic has no durable subscribers, there are no subscribers that need messages resent in the event of a server failure. Therefore, messages do not need to be saved, and performance is improved because disk I/O is not required.
The semantics for these fields are somewhat more complex than the explanation given here. See *TIBCO Enterprise Message Service User’s Guide* for more information.

### Session Reference

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

### Endpoint Reference

You can drag a different endpoint, browse for another endpoint resource, go to the referenced endpoint to edit its properties, or delete the endpoint. Endpoint reference objects are explained in *TIBCO Designer Palette Reference*. The guide can be displayed from any endpoint icon.

### Encoding Tab

This tab is available for both the Record Transfer and the Simple File Transfer modes.

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, the repository encoding needs to be set to UTF-8. If the repository encoding field is not set correctly, messages may be corrupted. For details, see File and Content Encoding on page 128.

### 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
• Other

Selecting **Other** allows you to enter an encoding string that is not present in the list. For a list of encoding strings, see *TIBCO ActiveMatrix Adapter for Files Concepts*.

If an invalid or unsupported encoding value is specified, the adapter will display an error at runtime.

**File System Encoding Other**

This is active only when selecting **Other** in the File System Encoding drop-down list. Enter an encoding string that is not present in the list. For a list of encoding strings, see *TIBCO ActiveMatrix Adapter for Files Concepts*.

**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
- Other

If you want to transfer a file in ASCII encoding between open systems in the Simple File Transfer mode, set the File Content Encoding to ASCII.

If you want to transfer a file in ASCII encoding between an open system and a mainframe system in the Simple File Transfer mode, set the File Content Encoding to ISO8859-1.

Selecting **Other** allows you to enter an encoding string that is not present in the list. For a list of encoding strings, see *TIBCO ActiveMatrix Adapter for Files Concepts*. 
The File Content Encoding option is not available when transfer mode (set in the Configuration tab) is Simple File Transfer and the File Transfer Mode (set in the SFT tab) is Binary.

If an invalid or unsupported encoding string value is specified, the adapter will display an error at runtime.

File Content Encoding Other
This is active only when selecting Other in the File Content Encoding drop-down list.

By default, the repository encoding is set to ISO-8859-1. When the system or file encoding is set to any encoding other than ASCII or LATIN_1, the repository encoding needs to be set to UTF-8. If the repository encoding is not set correctly, messages may get corrupted. For details, see File and Content Encoding on page 128.

End of Line
Choose the method by which lines in the input file are separated.

- **System** — Use a carriage return (new line) to mark the end of a line.
- **User Defined** — Use 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** — Use a combination of carriage returns and custom characters to mark the end of a line.

User Defined EOL
It’s active when the Line Separator field is not System. Enter the characters to mark the end of a line.

The End of Line and User Defined EOL fields are not available when the transfer mode is Simple File Transfer.

Advanced Tab
This tab is available only for the adapter’s Record Transfer mode.
User Exit Endpoint Reference

Select the UserExitClient (endpoint) for the callout or the user exit operation. Using callouts or user exits is not supported by TIBCO ActiveMatrix Adapter for Files.

Document Delay

The adapter publication service can sometimes overwhelm the underlying messaging infrastructure by publishing messages faster than they can be consumed. Under these circumstances, flow control can be exercised on the publication service. Flow control can be turned on by setting this field. Specify the document delay in milliseconds. This is the delay that will be applied when publishing a message. The message is either the MInstance or the MBusinessDocument.

The publication service will publish an MInstance or MBusinessDocument every time a delay timer triggers. When flow control is turned on, the publication service is automatically set to the asynchronous mode.

Grouping Factor

Specify how many objects need to be grouped for a single publication of a Business document. Multiple file records can be grouped into one Business document message.

Objects belonging to the same file are grouped together.

Data Format

- Select the format of data or message that will be published. You can choose Object (MInstance) or Business document (MBusinessDocument).

  When selecting Business document, MInstances can be batched before sending. For a given size of MInstance, the number of MInstances per business document can be varied to achieve the desired throughput.

  On the subscription side, an improvement will be seen when it receives an MBusinessDocument.

Business Document Name

It’s active only when the data format is Business document. This is the name of the Business document that will be published. This is a required attribute for the MBusinessDocument format of messages.
Input Buffer Max Size

Specify the amount of data. The publication service will read the data from the input file every time it reads the file. TIBCO recommends that you make this a multiple of the operating system block size to avoid wasting memory when reading a file. Usually, this field can be left with the default value.

Synchronous Mode

Check this checkbox if you want the publication service to operate in the synchronous mode. Clear this checkbox if you want the publication service to operate in the asynchronous mode. See TIBCO ActiveMatrix Adapter for Files Concepts for details.

Checkpoint Restart

Check this checkbox to enable checkpoint restart capability for the publication service. It continues from where it stopped when it recovers from a crash. Clearing the checkbox means a higher throughput is possible, but you will need to manually restart the service.

The restart information is stored in a progress file under the directory where the runtime adapter is executed. The name of the progress file is

`__InstanceNamePublicationServiceName.prg`

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

`__InstanceNamePublicationServiceNameSubjectName.prg`

Since the publication service does not update the progress file and perform post processing on the file automatically, when the adapter exits abnormally, the following situations may happen:

- **Duplicate Messages**

  Since the publication service does not publish messages and update the progress files automatically, duplicate files may be created when the publication service resumes.

  The number of the duplicate files is one if Object (MInstance) or Business document is used and the grouping size is one. The number of the duplicate files is more than one if Business document is used and the grouping size is greater than one.

  Using the MInstance message format, the subscriber will occasionally write 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

There could be situations where the post-processing on the working file is completed but the progress file is not updated. Upon restart, the adapter will prompt that the file in the working directory is missing.

**Publication Semantics**

If checked, the publication service publishes an object only if all its associations have no errors. For example, if a ReadSchema is defined as,

\[ H \]
\[ D \]

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

\[ H \]
\[ \text{field}_1 \quad \text{String} \quad \text{(Header)} \]
\[ \text{field}_2 \quad \text{ui4} \]

\[ D \]
\[ \text{field}_1 \quad \text{String} \quad \text{(Detail)} \]
\[ \text{field}_2 \quad \text{ui4} \]

and the input filename `input.txt` contains,

Header, 10
Detail, -20

Since `Detail, -20` has a validation issue, the entire object will not be published if this option is turned on. If this option is turned off, then the object containing only `Header, 10` will be published.

Therefore, if an object should be published only if all its associations (children) have no issues, check the **Only publish an identified object whose associations, if defined, have no errors** checkbox under **Publication Semantics** in the Advanced tab of the publication service. Using this option will guarantee that an object will be published only if all the identified associations have no errors.
Subscription Service Options

The following tabs can be used to define a subscription service:

- Configuration Tab, page 60
- Processing Tab, page 63
- Schema Tab, page 65
- SFT Tab, page 70
- Transport Tab, page 65
- Encoding Tab, page 68
- Advanced Tab, page 72

Configuration Tab

This tab is available for both the Record Transfer mode and the Simple File Transfer mode of the adapter.

Name

The name you want this subscription service to use. The name should be unique among other subscribers assigned to this adapter configuration. The name can contain only alphanumeric characters, including the underscore (_) character, and can be at most 80 characters long. The space character cannot be used in a name. Configuration names cannot use global variables.

Description

The description of the subscription service that you are configuring. This field is optional.

Transport Type

Select the transport to be used by the runtime adapter, JMS or TIBCO Rendezvous. After selecting the transport, the transport-specific configuration fields display on the Transport Tab.

The transport can be configured to use a trusted store and identity resource for use in SSL (Secure Sockets Layer) configurations. TIBCO Rendezvous sessions and JMS topics have an SSL configuration field which uses a dialog to perform SSL configuration.
To enable and configure SSL, in the Project panel, expand the Advanced>Sessions tree. Select the TIBCO Rendezvous session or JMS topic and click Use SSL? The SSL configuration options are explained in the online help associated with the session dialog.

**Transfer Mode**

This determines the operation mode for the publication service. The options are Record Transfer and Simple File Transfer.

**Wip Creation Mode**

This specifies the creation mode of the file under the working directory. This option is inactive when the Simple File Transfer mode is selected. The subscription service uses the working directory to create the file. Based on the semantics of the Wip Creation mode options, the file is then moved to the output directory.

The options are:

- **Append messages to file, Close on Timer**—The received messages are appended to a file created in the working directory. The file will be closed and moved to the output directory on receiving the timer event specified in the Elapsed field. After that, a new file is created in the working directory.

- **Append messages to file, Close on rvMessage**—The received messages are appended to the file created in the working directory. The file will be closed and moved to the output directory on receiving a TIBCO Rendezvous message on the subject specified in the Closing Subscriber field. After that, a new file is created in the working directory.

- **Append messages to file, Close on jmsMessage**—The received messages are appended to the file created in the working directory. The file will be closed and moved to the output directory on receiving a JMS message. After that, a new file is created in the working directory.

- **Append messages to file, Close on Business Document Lot End**—Close the working file of the subscription service when the LotEnd field is true and a business document is received. After that, a new file is created in the output directory. Use this option when transferring files. See the transferReader and transferWriter example in *TIBCO ActiveMatrix Adapter for Files for Unix/Win Examples* for more information.

- **One message per file**—Each received message is written to a file and then moved to the output directory.
Output Creation Mode

- **Overwrite (always create a new file)** — When moving a file from the working directory to the output directory, overwrite the file.

- **Append (if same filename exits)** — When moving a file from the working directory to the output directory, append the text that received from the publication service to the file in the output directory.

File Name

Name of the output file that needs to be created. This is active only if you select **From Configuration** in the File Name Selection drop-down list.

The filename cannot contain path information.

File Name Selection

If using the Record Transfer mode, only the From Configuration option is available. You must specify a name in the File Name field. The subscription service will use the name that you specify in the File Name field as the output filename.

For the Simple File Transfer mode, you can select **From Configuration** or **From Message**. If you select **From Message**, the subscription service will use the same filename that is available in the message as the output filename.

Close file after time elapsed (minutes)

The time before the file is closed in the working directory. You can type a specific time in this field or use a global variable. This field is active only when selecting **Append messages to file, Close on timer** in the Wip Creation Mode drop-down list.

Message Time Out (minutes)

It’s active only if you select **Append messages to file, Close on timer** in the Wip Creation Mode drop-down 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 the subscription service closes the work-in-progress file and transfers it to the output directory. The default value is zero that indicates no timeout.
**Closing Subscriber**

It’s active only if you select *Append messages to file, Close on rvMessage* or *Append messages to file, Close on jmsMessage* in the Wip Creation Mode drop-down list. The subscriber used to receive and signify the output file should be closed.

**Processing Tab**

This tab is available for both the Record Transfer mode and the Simple File Transfer mode of the adapter.

**Working Directory**

Directory where contents for the actual output file is composed from the received messages.

If the files generated by the subscription service(s) are independent of each other, the service(s) can 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 need to be on the same file system.

**Output Directory**

The file in the working directory is moved to this directory.

**Error Directory**

For details about the contents of this directory, see Error Handling on page 100.

**Progress Directory**

The progress file is written to this directory. It applies to both the Record Transfer and Simple File Transfer modes. If no directory is specified in this field, the progress file is created under the directory where the adapter is started.

**Post Processing Script File**

Specify the name of the script that needs to be executed after the file is moved to the output directory. Click *Browser* to locate and load the script.
Post Processing Arguments

Pass the arguments to the post-processing script. The sequence of the arguments passed to the post processing script is determined as follows:

- If the Transfer Mode for the subscription service is Record Transfer or Simple File Transfer and Explicit Confirmation Mode (ECM) is not selected in the SFT tab, the sequence of the arguments contains the name of the file that is followed by the arguments specified in the Post Processing Arguments field.

- If the Transfer Mode is Simple File Transfer, and Explicit Confirmation Mode (ECM) is selected, the sequence of the arguments will contain the name of the file, name of the host, user ID, and then the arguments specified in the Post Processing Arguments field. The host name identifies the host name of the publisher, and the user ID contains the name of the user specified in the TIBCO Rendezvous message trigger of the publication service. The user ID field is empty if the TIBCO Rendezvous message trigger does not contain the user ID.

Options for modifying output Filename

Provides options to modify the name of the generated output file by appending a timestamp or sequence number.

- **Append**
  - None—Do not append anything to the filename.
  - Timestamp—Append a timestamp to the filename. The format is YYYYMMDDHHMMSSmm. Selecting this option will activate the Location field.
  - Sequence Number—Append a sequence number to the filename. Selecting this option will activate the Location and Sequence Number Width fields.

- **Location**
  - Prefix—Append the timestamp or sequence number to the filename as a prefix.
  - Suffix—Append the timestamp or sequence number to the filename as a suffix.

  The separator between the prefix and suffix is assumed to be a period (.)

- **Sequence Number Width**

  You can specify the width of the sequence number. For example, if you select 4 as the width of the sequence number, the number that is appended to the file will be 0000. The sequence number begins with 0 and will be incremented for
each file that is moved to the output directory. Upon reaching the maximum value for a given width, the sequence number will roll over.

If the adapter is restarted, the subscription service will remember the previous sequence number and continue from there.

**Schema Tab**

This tab is available only for the Record Transfer mode.

Associate the WriteSchema(s) that have been created to the subscription service by clicking the **add wire schema** button and selecting the write schema from the pop-up dialog box. Repeat this to associate more than one WriteSchema.

**Transport Tab**

This tab is available for the Record Transfer and the Simple File Transfer modes.

**Message Subject (TIBCO Rendezvous Transport only)**

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

**Quality of Service (TIBCO Rendezvous Transport only)**

Any of the following choices can be specified:

- **Certified**—Guarantees that every certified message reaches its intended recipient in the order sent. The message can be sent across network boundaries, and if a network fails, delivery attempts continue until delivery succeeds or until the message’s time limit expires. This is often called guaranteed delivery. When this quality of service is chosen, an RVCM session will be used.

- **Reliable**—Ensures that each multicast or broadcast message is received as long as the physical network and packet recipients are working, and that the loss of a message is detected. This choice can compensate for brief network failures because it can retransmit a message on request if the first attempt fails. This choice is appropriate when message delivery is expected but some loss can be tolerated. When this quality of service is chosen, a Rendezvous session will be used.
• **Distributed Queue**—A distributed queue is a group of cooperating transport objects, each in a separate process. To obtain load balancing among servers, the adapter uses distributed queues for one-of-n delivery of messages to a group of servers. Each member of a distributed queue listens for the same subject using the TIBCO Rendezvous Distributed Queue listener objects. Even though many members listen for each inbound message (or task), only one member processes the message. For details on distributed queues, see *TIBCO Rendezvous Concepts*.

Distributed Queue is supported only if the adapter configuration is a subscription service and its mode of operation is Record Transfer Mode.

**Wire Format**

Choose the expected wire format for arriving messages. Publishers and subscribers can only send and receive data if they agree on a specific wire format.

This field is only enabled in the Record Transfer mode.

In the Simple File Transfer mode, this field is read-only. Its value is Rendezvous Message by default for the Rendezvous transport type and XML Message by default for the JMS transport type.

In the Record Transfer mode, you can select **Rendezvous Message**, **ActiveEnterprise Message**, or **XML Message** for the Rendezvous transport type, and you can only select **XML Message** for the JMS transport type.

• **Rendezvous Message** — Control information for validation is *not* expected in the message. For TIBCO Rendezvous wire format, a message of type RVMSG_RVMSG (TIBCO Rendezvous 5.x) or TIBRVMSG_MSG (TIBCO Rendezvous 6.x) is always expected when this format is selected. If a publisher sends a simple RVMSG_STRING or TIBRVMSG_STRING, an exception is passed to the subscriber. This option is not available for the Record Transfer mode of the adapter. It is available only for the Simple File Transfer mode.

• **ActiveEnterprise Message** — Control information for validation is expected 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 TIBCO ActiveMatrix Adapter SDK set of data types. This format allows ActiveEnterprise components to perform extra validation on messages sent or received. This option is not available for the Simple File Transfer mode. It is available only for the Record Transfer mode.
• **XML Message** — It’s a text message which contains XML data that can be validated against an XSD. This option is available only for the Record Transfer mode.

See the TIBCO ActiveMatrix Adapter for SDK documentation for details about the control information generated and sent with ActiveEnterprise messages.

**Destination (JMS Transport only)**

By default, a service uses a dynamic destination that is generated using the **Domain** and **Deployment** global variables, the adapter acronym, the adapter instance name and the service name. If you use this default dynamic destination, make sure the values for **Domain** and **Deployment** are not empty. You can override the default dynamic destination by specifying the static destination in this field. The static destination must be defined on the JMS server before it can be used by the runtime adapter.

**Connection Factory Type (JMS Transport only)**

A subscriber can get messages that have been published to a topic or placed in a queue.

• **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 known as 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 may connect to the queue. The first receiver to access the queue gets the message. The other receivers do not. This messaging model is known as point-to-point. A subscription service that receives messages from a queue is always a durable subscriber.

**Delivery Mode (JMS Transport only)**

An adapter subscription service can be Durable or Non Durable.

• **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 consumed by the service. The service can be down and expect to receive its messages when it comes back up.

• **Non Durable** — Messages sent to a non-durable service are not held by the JMS server. If the subscription service is down, it will not receive the messages that arrived at the JMS server while the service was down.
The semantics for these fields are more complex than the explanation given here. See the TIBCO Enterprise Message Service documentation for more information.

**Session Reference**

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

**Endpoint Reference**

You can drag a different endpoint, browse for another endpoint resource, and go to the referenced endpoint to edit its properties or delete the endpoint. Endpoint reference objects are explained in the *TIBCO Designer Palette Reference*. The guide can be displayed from the endpoint icon.

**Encoding Tab**

This tab is available for both the Record Transfer mode and the Simple File Transfer mode of the adapter.

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, the repository encoding needs to be set to UTF-8. If the repository encoding field is not set correctly, messages may be corrupted. See *File and Content Encoding on page 128* for details.

**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
• Other

Selecting Other allows you to type an encoding string that is not present in the list. For a list of encoding strings, see TIBCO ActiveMatrix Adapter for Files Concepts.

If an invalid or unsupported encoding value is specified, the adapter will display an error at runtime.

**File System Encoding Other**

This is active only when you select Other in the File System Encoding drop-down list. Type an encoding string that is not present in the list. For a list of encoding strings, see TIBCO ActiveMatrix Adapter for Files Concepts.

**File Content Encoding**

The File Content Encoding option is not available when selecting Simple File Transfer in the Transfer Mode drop-down list (in the Configuration tab) and you select Binary in the File Transfer Mode drop-down list (in the SFT tab).

Provides aliases for the following commonly used encoding for file contents:

- ASCII
- ISO8859-1
- UTF16_BigEndian
- UTF_LittleEndian
- UTF-8
- Shift JIS (CP943)
- Shift JIS (TIBCO)
- EUC-JP
- Big5
- Other

Selecting Other allows you to type an encoding value that is not present in the list. This value should reflect the setting that is made for the subscription service. The subscriber is associated with the publication service. For a list of encoding strings, see TIBCO ActiveMatrix Adapter for Files Concepts.

If an invalid or unsupported encoding string value is specified, the adapter will display an error at runtime.
**File Content Encoding Other**

This is active only if you select **Other** in the File Content Encoding drop-down list.

**End of Line**

Specify the line separator to be used in the output file:

- **System** — The separator is runtime dependent. It will follow the DOS convention on Microsoft platforms or 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 the characters to mark the end of line. Specify the characters in the User Defined EOL field.

The End of Line option is not available when the transfer mode is Simple File Transfer.

**User Defined EOL**

It’s active when you select the User Defined item in the End of Line drop-down list. Enter the characters to mark the end of a line.

The End of Line and User Defined EOL fields are not available when the transfer mode is Simple File Transfer.

**Byte Order Mark**

It applies to UTF8, UTF16BE or UTF16LE file encoding. It is not selected by default. If not selected, a byte order mark is not inserted. If selected, a byte order mark is inserted at the beginning of an output file.

**SFT Tab**

This tab is available only for the Simple File Transfer mode.
Explicit Confirmation Mode (ECM)

If ECM is checked, only the TIBCO Rendezvous reliable messaging quality of service is available. If ECM is not checked, files are transferred using TIBCO Rendezvous reliable or certified messaging quality of service.

- If using ECM,
  - On starting, the subscription service, irrespective of the mode (strict ECM or flex ECM), waits to be discovered. This continues until it is discovered. Once discovered, the subscription service creates a progress file. The name and location of the progress file can be configured during design time.

    After discovery, the subscription service queries the publication service to determine if it (the subscription service) is active. If not active, it requests activation. Once it is active, the subscription service starts participating in the file transfer when a new file transfer begins.

    When a subscription service is restarted, it uses the progress file it had created and proceeds to query if it is active. If it is active, the subscription service continues from where it left off. If it is not active, the subscription service requests for activation and, once it is active, it starts receiving the file transfer, when a new transfer begins. If the subscriber was processing a file when it requested activation it will move the file to the error directory.

    When the subscription service is restarted, it does not wait to be discovered. The presence of a progress file indicates that the service is registered with a publication service on the network. Removing the progress file before starting the subscription service has the same effect as starting the service for the first time.

You can configure the adapter to send a TIBCO Rendezvous message when a registered inactive subscriber becomes active after requesting for an activation. The TIBCO Rendezvous message will contain a field with the subscriber’s name. To send this TIBCO Rendezvous message on a user-defined subject, create a network sink and role named `ecmSubActive`, and associate the role with the network sink.

To create user-defined roles such as `ecmSubActive` and to use network sink, advanced logging options for the adapter configuration should be turned on in TIBCO Designer.

- At any time, if the subscription service receives a discovery message from the publication service, the subscription service will be reset, that is, the subscription service will close the file it is currently processing, move it to the error directory, and wait for a new file transfer.

- If a subscription service experiences an IO error during the transfer of a particular file, the publication service, irrespective of operating in Strict
ECM or Flex ECM, will mark the subscription service as inactive and continue with the transfer of that particular file. After sending the IO error status, the subscription service will request for an activation. The subscription service will start receiving and processing files after it is activated by a publication service.

- If using non-ECM, the subscription service maintains no state information and features such as checksum verification and error handling are not supported. No matter when the subscription service is started, it always starts processing a Simple File Transfer on a new file boundary.

**ECM Subscriber Name**

Provide a name to identify the subscription service. This name must be entered in the Pre-registered ECM Subscribers field for the publication service that is initiating the file transfer. The name must be unique among the subscription services defined for the publication service that initiates the file transfer.

**Progress File Name**

Pathname to a file where details about the file transfer are recorded. If only one filename is entered, the progress file is created in the directory specified in the Progress Directory field in the Processing tab.

**File Transfer Mode**

Choose any one of the following methods for transferring files:

- **Binary** — This mode is used for opaquely transferring files and is used for any type of file where the file contents must not be altered during file transfer.

- **Text** — This mode is used for transferring text files. When transferring files across platforms, the file created on the receiving platform may 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.

This mode should reflect the setting that is made for the publication service. The subscriber is associated with the publication service.

**Advanced Tab**

This tab is available only for the Record Transfer mode.
User Exit Endpoint Reference

Select the UserExitClient (endpoint) that needs to be used to perform the callout or the user exit operation. Using callouts or user exits is not supported by TIBCO ActiveMatrix Adapter for Files.
Defining Schemas

Defining schemas is relevant when the adapter is operating in the Record Transfer mode. This section discusses the mechanics of creating read schemas and write schemas.

Defining or generating the read schema or write schema is done at the instance level. Therefore, the defined read schema(s) can be shared across publication services of that instance, and the generated or created write schemas(s) can be shared across subscription services of that instance.

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

Defining Read Schemas

In the project tree panel, click the File Schemas folder. In the Palettes panel, select and drag a Read Schema to the Design panel. Optionally, you can specify a new name for the schema and provide a description. Select Show Base Records Folder if you creating a file record that will contain a Record data type, that is, for Container records. For information on Container Records and Base Records, see TIBCO ActiveMatrix Adapter for Files Concepts.

For a read schema, you can create file records and define relations among them. The read schema can contain a combination of delimited file records or positional file records.

Defining Relations

After having created the first file record for the read schema, defining additional records for the read schema will automatically form a relation. The kind of relationship that is created among file records depends on how you create the file records. The first file record that is created under the read schema is referred to as the root record. There can be only one root record that can be defined for the read schema. All the other records that are created after the first file record are referred to as child records of the root.
The root record can have any number of children and each child can further have any number of children. The child file records that are defined immediately under the root record have a special attribute called Position from Header. This allows the immediate children of the root record in the actual data file to appear before or after the root record.

This special attribute is only available for children defined under the root record.

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

Order
  Customer
  Item

If the data file contains the records as,

Order
Customer
Item
Item
Order
Customer
Item

the Position from Header attribute for the Customer file record and the Item file record should be defined as Begin.

If the data file contains records as

Customer
Item
Item
Order
Customer
Item
Item
Order

the Position from Header attribute for the Customer file record and the Item file record should be defined as End.

The Position from Header attribute setting for the child records should be 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.
Delimited File Record Options

**Figure 9 Delimited File Record Options**

*Figure 9 Delimited File Record Options*

**Name** — File record names must only have alphanumeric characters and must be at most 80 characters long. Each name must be unique within the adapter instance.

**Position from Header** — The option is available only for records that are defined as immediate children of the root record. Select *Begin* or *End* based on the structure of the actual records in the data file. See *Defining Relations on page 74*.

**Strip Blanks** — When selected, blank spaces are removed if any, for each field identified for the record in data file.

**Repeating** — Specifies whether repeated delimiters should be ignored. If selected, then any repeated delimiters between fields are translated as only one field separator. If this checkbox is cleared, then repeated delimiters are translated as empty fields.

**Delimiter** — Specifies the separator between fields within each line of the file for the line processed. Choices include Space, Tab or Other. If you choose *Other*, the Delimiter Other field becomes active.
**Delimiter Other** — Active only if the Delimiter field has been assigned the Other value. Specify the delimiter to be used. Single or multiple characters can be specified as the delimiter.

To distinguish delimiter characters that are not actual delimiters but actual field values, you can enclose data fields including the delimiter characters in double quotes and set `adfiles.quotedField` to ON in the `your_runtime.tra` file.

For example, the comma-delimited record `John, "3301 Hillview Ave, Palo Alto CA"` will be treated as one single field because of the double quotes in the second through fourth fields.

The quoted field is supported with regular records only. Container records and End-of-line can not be enclosed in the double quotes.

**Identifier Type** — Option to specify the method to identify a record. Specify the Field Value (constant field value) or Number of Fields. If the field value is selected, then when defining the fields for the records the constant attribute for one of the fields needs to be specified.

When the Identifier Type is Field Value, the 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" will be accepted by the publication service. However, a record with a valid constant Order but containing four fields, such as "order, ID1234, Aug20, CA" will be rejected. The following error will be logged: "1 lines couldn't be interpreted."

To use this feature, you need to set the `adfiles.matchFieldCountAndRecordLength` property to ON in the `your_runtime.tra` file.

Currently, only one constant field value can be used to identify the record.

**Display Wizard** — If selected, it brings up a wizard text box which can be used to display the contents of the actual file for which the record is being created. This is a convenience provided to generate the fields for the record.

If you are not using the feature then files for the record can be created by using the Add button.
Attributes for the Fields

Name — The name of the filed that is automatically generated when Add is clicked is field, field1, field2 and so on. This can be edited to your choice of the field name.

Type — The type of the field need to be specified. This is the type that the actual data in the corresponding field in the file will be converted to before publication.

If using the dateTime type, you must specify the pattern and locale. This can be done by right-clicking on the field name and selecting Set locale and Pattern. This will bring up a window for specifying the locale. Use the drop-down list to select the various locales that are supported.

The record type is used for interpreting 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 will bring up a pop-up dialog that will display the base records that are defined. For more information on data validation, see Data Validation on page 110.

When using the record data type, it is recommended that you first define the base records.

Constant — This is active only if Identify Type is selected as Field Value. Enter the constant value for the appropriate field.

Sample Value — This attribute shows the sample value. This is set to <unknown> if Display Wizard is turned off, and shows the actual value if the display wizard is on and the line in the file is highlighted. Options to remove and shuffle the fields are provided. If the display wizard is selected, then creating the fields and the names of the fields (in some cases) becomes easier.

1. Select the Encoding of the file content. This is important and has to be correctly selected before the file is opened.

2. Use the Browse button to navigate to the file. After identifying the desired file, click Open. The file will be opened in the Display Wizard window.

3. Highlight the line you wish to create a file record and click Create.

This will create the fields corresponding to the line using the specified delimiter as the reference. The generated fields will be named by default as field, field1, field2, and so on.

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

**Name** — File record names must only have alphanumeric characters and must be at most 80 characters long. Each name must be unique within the adapter instance.

**Position from Header** — The option is available only for records that are defined as immediate children of the root record. Select `Begin` or `End` based on the structure of the actual records in the data file. For details, see Defining Relations on page 74.

**Strip Blanks** — When selected, blank spaces, if there are any are removed, for each field identified for the record in the data file.

**Identifier Type** — Option to specify the method to identify a record. Specify Field Value (constant field value) or Record length. If field value is selected, then when defining the fields for the records the constant attribute for one of the fields needs to be specified.

Currently, only one constant field value can be used to identify the record.

In addition to checking only the constant field value in validating a record, a publication service can also be configured to check the record length.

For example, a read schema defines a record to contain two string fields. The first field contains a constant string "Order". The second field contains the value 5, representing the length. (The record length is 10.) A record, `Order12345`, will be accepted with no error. A record, `Order12345678`, will be rejected since the record length is 13. The following error will be logged: "1 lines couldn’t be interpreted."

To use this feature, you need to set the `adfiles.matchFieldCountAndRecordLength` property to `ON` in the `your_runtime.tra` file.

**Length** — This is active only if identifier type is Record length.

**Display Wizard** — If checked, this brings up a wizard text box which can be used to display the contents of the file for which the record is being created. This allows you to specify the start and length of each of the fields.

If you are not using the feature then fields for the record can be created by using the Add button.
Attributes for the Fields

**Name** — The name of the field that is automatically generated when Add is clicked is field, field1, field2, and so on. This can be edited to your choice of the field name.

**Type** — The type of the field need to be specified. This is the type that the data in the corresponding filed in the file will be converted to before publication.

If using the Date Time type, you must specify the pattern and locale. This can be done by right-clicking on the field name and selecting **Set Locale and Pattern**. This will bring up a window for specifying the locale. Use the drop-down list to select the various locales that are supported. For more information on data validation, see **Data Validation on page 110**.

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 may not receive an error.

In order to be certain that the pattern used is correct, it is recommended that the date and time actually published be verified 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** — This is active only if Identifier Type is set to **Field Value**. Enter the constant value for the appropriate field.

**Sample Value** — This attribute shows the sample value. This is set to <unknown> if the Display Wizard is turned off and shows the actual value if the display wizard is on and the line in the file is highlighted.

Options to remove and shuffle the fields are provided.

If the display wizard is checked then specifying the start and length attributes of the fields becomes easier.

1. Select the Encoding of the file content. This is important and has to be correctly selected before the file is opened.
2. Use **Browse** to navigate to the file. After identifying the desired file, click **Open**.
3. The file will be opened in the Wizard window.
4. Highlight the line you wish to create a file record. This will show the line in the Sample Line box.
5. Using the **Add** button add a field.
6. Select the text in the sample line. This will set the start and the length for the fields.

7. Repeat step 5 and 6 for creating other fields.

After creating all the fields for the record, if the Identifier type is Record length then use the Update length to update the Length configuration parameter.

Defining Write Schemas

There are two types of write schemas from release 6.2.0: WriteSchema(Classic) and WriteSchema(Manual). This section describes how to define the two types of write schemas:

- Defining WriteSchema(Classic)
- Defining WriteSchema(Manual)
- Defining Relations
- Layout File Record Options
- Editing the File Record

Copying and pasting write schemas is not supported.

Defining WriteSchema(Classic)

Complete the following steps to define WriteSchema(Classic):

1. Add WriteSchema(Classic) to a subscription service.
   a. In the project tree panel, click the File Schemas folder.
   b. In the Palettes panel, select and drag the WriteSchema(Classic) icon to the design palette.
   c. Optionally specify a new name for the write schema and click the Apply button.
2. Wire WriteSchema(Classic) to an existing schema.
   a. In the Configuration tab, click the button. The Select a Resource dialog is displayed.
   b. Select the wire schema or the canonical schema from which WriteSchema(Classic) needs to be generated. Click the OK button to generate the write schema.
      The read schema defined by the adapter creates the wire schema or the canonical schema under the folder AESchemas/ae/FileAdapter/wire/adapter configuration name.
   c. Click the Apply button.
3. Optionally provide a description about the write schema.
4. The Write Schema Type field is displayed with read-only value Classic.
5. Optionally edit the configuration options of the generated write schema. See Layout File Record Options for more information.

WriteSchema(Classic) cannot be created from a wire schema that is created by a publication service under the same adapter instance.
There can be only one root record created for WriteSchema(Classic).
The root record can have any number of child records and each child record can have any number of child records.

Defining WriteSchema(Manual)

Complete the following steps to define WriteSchema(Manual):
1. Add WriteSchema(Manual) to a subscription service.
   a. In the project tree panel, click the File Schemas folder.
   b. In the Palettes panel, select and drag the WriteSchema(Manual) icon to the design palette.
   c. Optionally specify a new name for the write schema, and add a description about the write schema.
   d. The Write Schema Type field is displayed with read-only value Manual.
   e. Click the Apply button.
2. Create a root file record of the write schema.
   a. Double-click the **WriteSchema(Manual)** icon in the design palette.
   b. In the Palettes panel, select and drag the **Layout File Record** icon to the design palette.
   c. Specify the configuration options (see **Layout File Record Options**), and click the **Apply** button.

3. Create the child file record(s) of the root file record.
   a. In the design palette, double-click the root file record icon.
   b. In the Palettes panel, select and drag the **Layout File Record** icon to the design palette.
   c. Specify the configuration options (see **Layout File Record Options**), and click the **Apply** button.
   d. Repeat step a, b, and c for creating more child file records of the root file record.

There can be only one root record created for **WriteSchema(Manual)**.
The root record can have any number of child records and each child record can have any number of child records.

**Defining Relations**

To view the write schema, navigate to the Project panel and expand the write schema you created. This will show the file records and the corresponding relations as it was defined in the wire schema or the canonical schema, or **WriteSchema(Manual)**.

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

The root record can have any number of child records and each child record can have any number of child records. The child file records that are defined immediately under the root record have a special field called position from header. This field allows the immediate child records of the root record in the actual data file to be generated before or after the root record.

This special field is only available for child records that are defined under the root record.

If a write schema is defined as follows:
Order
  Customer
    Item

If the position from the header field for the Customer file record and Item file record is specified as Begin, the contents of the generated data file will appear as follows:

Order
  Customer
    Item
    Item
    Item
    Order
    Customer
    Item
    Order

If the position from the header field for the Customer file record and Item file record is specified as End, the contents of the generated data file will appear as follows:

Customer
  Item
  Item
  Order
  Customer
  Item
  Order

The position from the header field setting for the child records should be 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.

**Layout File Record Options**

Figure 10 shows the layout file record options.
Figure 10  Layout File Record Options

UI Display Name—By default, the name of the file record as given in the wire schema is displayed.

Name—By default, the name of the file record as given in the wire schema is displayed. This can be modified if necessary. File record names must only have alphanumeric characters and must be at most 80 characters long. Each name must be unique within the adapter.

Position From Header—The option is available only for records that are defined as immediate child records of the root record. Select Begin or End based on the structure you wish to generate in the output file.

Wire Schema Path—Wire schema that was used to create this write schema.

Parse—Specifies whether the generated output line should use a delimiter to separate fields or whether it should use absolute line positions. The following are the choices for this field:

- 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 will begin at a fixed position (offset from the start of the line).

Delimiter—Specifies the separator between fields within each line of the file for the line processed. Choices include Space, Tab, or Other. If you choose Other, the Delimiter Other field becomes available.

Delimiter Other—Active only if the Delimiter field has the Other value assigned. Specify the delimiter character(s) to be used.
Attributes for Write Schema

**Name**—The name of the attribute of the object from the incoming message. The value of this field is populated automatically when generated from the WireSchema for WriteSchema(Classic).

**Type**—The data type of the attribute. The value of this field is populated automatically when generated from the WireSchema for WriteSchema(Classic).

If using the Date Time type, you must specify the pattern and locale. This can be done by right-clicking on the field name and selecting **Set Locale and Pattern**. This will bring up a window for specifying the locale. Use the drop-down list to select the various locales that are supported. For more information on data validation, see **Data Validation on page 110**.

For WriteSchema(Classic), you will not see the record type after generating the write schema from the wire schema. However, you can promote a child record by right clicking the child record and clicking **Promote**. The promoted child record will be an attribute of its parent and the type will be Record. You can demote a child record by right clicking the child record in the Attributes table and selecting **Demote**.

**Width**—Number of characters used to represent the field. Zero (0) signifies the field should not be included in the output line. Negative one (-1) signifies that as many characters as needed (without padding) should be used (this is the default value for lines that use delimiters. Lines using the Positional method cannot specify -1).

**Alignment**—Specifies whether the data should be 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 prepend the positive sign before positive numbers in the output. Can be True if you wish the sign to be output. Otherwise, it should be False.

**Precision**—Specifies the number of digits after the decimal place. For example, with a precision of 3, the number 3.14159 would be displayed the following ways:

- Scientific notation: 3.142e+1
- Non-scientific notation: 3.142

A value of -1 indicates that the entire precision provided by the operating system should be used.

**Sc. Notation**—For real numbers only, used to determine whether scientific notation should be used for the output of this field. Set to True to use.
**Upper Case**—Only valid when Sc. Notation is set to True. If this field is set to True, the E in the scientific notation will be in uppercase. False is indicated by using lowercase.

**Editing the File Record**

You may need to change the fields of the attributes or change the number of the attributes.

To edit an attribute, click the attribute.

To add an attribute, click the **Add** button.

Use the **Remove** button to remove a field and use the **Up** and **Down** button to shuffle the fields.

If you need to make significant changes to `WriteSchema(Classic)`, it is highly recommended to regenerate the write schema rather than editing it.

**Modifying 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,

```
WriteSchema
   FileRecord_Parent
     FileRecord_Child_1
       FileRecord_Child_2
     ...
   ...
```

The sequence of 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 its order.
Once you change the order of the child records of a write schema, the runtime will write the child records to a file in the exact same order as you have configured for the write schema in the design time.

If you rearrange the order of the child records in TIBCO Designer and also set the `adfiles.toggleChildRecordsOrdering` property in the TRA file, the property setting will have no effect.

Projects, created in earlier versions of 6.2.0 with the `adfiles.toggleChildRecordsOrdering` property enabled, can be successfully run in this release.

In some situations, the child records may need to be written as part of the parent record. This can be achieved by promoting a child record. A promoted child becomes a field of type record in the parent.

Conversely, a promoted child record can be demoted.

**Promoting and Demoting a Child Record**

To promote a child record, right-click the child record and select **Promote**. A promoted child record is displayed as an attribute of its parent. The Type of the promoted child is Record.

To demote a child record, identify and select the attribute in the parent record which needs to be demoted. Right-click the attribute and select **Demote**. The attribute will be removed from the parent record and append to the end of the child records.

**Restrictions on Promoting Child Records**

- A child record can be promoted only to its immediate parent. For example:
  
  ```
  FileRecord_Parent
  FileRecord_Child_1
  FileRecord_Child_2
  
  can only be promoted to `FileRecord_Child_1`:
  ```
  FileRecord_Parent
  FileRecord_Child_1 [FileRecord_Child_2]
  ```

- It cannot be promoted to `FileRecord_Parent`:
  ```
  FileRecord_Parent [FileRecord_Child_2]
  FileRecord_Child_1
  ```

- Only a leaf child record can be promoted. For example,
  ```
  FileRecord_Parent
  ```
FileRecord_Child_1 [FileRecord_Child_2]
is allowed. However, the following is not allowed:
FileRecord_Parent FileRecord_Child_1 [FileRecord_Child_2]
Chapter 4  Deploying and Starting an Adapter Using TIBCO Administrator

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

Topics

- Creating an EAR File in TIBCO Designer, page 92
- Deploying the Project, page 93
- Starting or Stopping the Adapter, page 94
- Monitoring the Adapter, page 95
Creating an EAR File in TIBCO Designer

Generate an Enterprise Archive file (EAR) that contains information on what you wish to deploy. This could be one or more adapter services, one or more TIBCO ActiveMatrix BusinessWorks process engines, or both.

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

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

1. Configure the adapter services.
2. In the Project panel, select the project.
3. Drag the Enterprise Archive resource from the Palettes panel to the Design panel.
4. Double-click the Enterprise Archive resource you have created.
5. Drag the Process Archive resource from the Palettes panel to the Design panel. If there are any processes in your project, configure them using the Browse Resources button.
6. If there are any configured adapter services in your project, an Adapter Archive resource becomes available in the Palettes panel. Drag the Adapter Archive resource to the Design panel and specify information in the Configuration tab, then click the Apply button.
7. Go to the Enterprise Archive resource, and click the Build Archive button in the Configuration tab to create the EAR file.

See Also

See the TIBCO Designer documentation for more information about this procedure.
Deploying the Project

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

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

To deploy a project:

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

See Also

See the TIBCO Administrator documentation for an introduction to the TIBCO administration domain and detailed information about the above steps.
Starting or Stopping the Adapter

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

To start an adapter service from the module:

1. In the Administrator GUI left pane, expand Application Management > Application-Name > Service Instances.
2. In the Service Instances panel, select the checkbox next to the adapter service.
3. Click the Start Selected button.
   The status changes from Stopped to Starting Up to running.
4. To stop the adapter service, select it and click the Stop Selected button.

The adapter does not start if it is deployed in a cluster and it is configured to access a file in a mapped drive. If the mapped drive is changed to a local drive or a cluster shared drive, the adapter starts successfully.

See Also

See the TIBCO Administrator documentation for more information.
Monitoring the Adapter

TIBCO Administrator offers a number of monitoring options.

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

See Also

See the TIBCO Administrator documentation for information about configuring the above monitoring options.
Chapter 5  Advanced Topics

Topics

- Using the Adapter with a Revision Control System, page 98
- Error Handling, page 100
- Using Global Variables, page 104
- File Recognition Methods for Publication Service, page 108
- Data Validation, page 110
- Syntax for Specifying User-defined Date and Time Patterns, page 115
- Locales Supported for Date and Time, page 117
- Using Trigger Messages, page 122
- Using Selective Routing Over JMS, page 125
- Dynamically Changing Output File Names at Runtime, page 127
- File and Content Encoding, page 128
- Business Event Messages, page 130
- Simple File Transfer Status Message, page 131
- Load Balancing Mode, page 133
Using the Adapter with a Revision Control System

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

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

Project_root/AESchemas/ae/FileAdapter/FileAdapterConfiguration.aeschema
When the project is saved and a revision control system has been specified, the adapter displays a warning that additional files were created and should be added to the revision control system. This warning is displayed only when the files are created for the first time. The warning displays a Go To Resource button that helps in navigating to the resource. You should select the **Multi-User > Add Resources to RCS** menu to add these files to the revision control system.

![Warning: Add File To RCS](image)

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

**Copy, Cut, Paste, and Move Operations**

To successfully copy and paste a service from adapter `FileAdapterConfiguration` to `FileAdapterConfiguration1`, the adapter configuration and schema files for the `FileAdapterConfiguration1` must be checked out.

To successfully cut and paste a service from adapter `FileAdapterConfiguration` to `FileAdapterConfiguration1`, the adapter configuration and schema files for both `FileAdapterConfiguration` and `FileAdapterConfiguration1` must be checked out.

To successfully move a service from adapter `FileAdapterConfiguration` to `FileAdapterConfiguration1`, the adapter configuration and schema files for both `FileAdapterConfiguration` and `FileAdapterConfiguration1` must be checked out.
Error Handling

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

Using Error Handling with Record Transfer Mode

When the publication service detects a parsing error that violates the pre-defined schema, an XML file containing the name of the input file and the error details will be created. To use this feature, you must specify a valid directory in the Error Directory field in the Processing tab.

For example:

If a ReadSchema is defined as:

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

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 the publication service parses an input file `input.txt`, which contains the following records,

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

the line `Header -10` will fail validation. The xml file that is created is named `input.txt20030917151048000.xml`

```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 will be kept for analysis. The location and name is dependent on the post processing option specified in the Processing tab. If the input file is abc.txt, the following describes the name and location based on the various post-processing options that are available:

**Leave as is:**
1. abc.txt is retained in the input directory
2. abc.txt.timestamp is created in the error directory
3. abc.txt.timestamp.xml is created in the error directory

**Delete:**
1. abc.txt is deleted from the input directory
2. abc.txt.timestamp is also created in the error directory
3. abc.txt.timestamp.xml is created in the error directory

**Move to (without timestamp):**
1. abc.txt is moved to the done directory
2. abc.txt.timestamp.xml is created in the error directory
3. 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.

**Diagnostics File**

If the invalid records violate the schema and also contain invalid characters, all the invalid records and their line numbers will be created in a file under the specified error directory. The diagnostics file provides useful information for users to correct the invalid records and then re-publish them.

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

```
Order, ID41678, <10Apr2000
Item, GigaWidget, 60, $75
Item, MegaBucket, 48, $60
Customer, Hopkins Associates, ID36800
Order, ID41680, 20May2000
```

TIBCO ActiveMatrix Adapter for Files for Unix/Win Configuration and Deployment
The schema diagnostics file has the format of an XML file and has an “.xml” extension so it can be opened and viewed in a text editor as well as in an Internet browser. However there is a special case where a diagnostics file maybe deemed not well-formed XML file by an Internet browser and so cannot be opened. This error occurs when the invalid records contain special XML characters.

The XML specification states that characters ‘&’, ‘<’, and ‘>’ are special characters and these special characters when appear in certain part of an XML document must be “escaped” (i.e. replaced) by strings “&amp;”, “&lt;”, and “&gt;” respectively.

When invalid records containing special XML characters are written to the diagnostics file as-is without the proper conversion, the diagnostics file becomes an not 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 Files Adapter, you can create a global variable “adfiles.schemaDiagnosticsFileFormat” and use it to specify the desired format for the diagnostics file. The three valid values for the global variable 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 File Adapter by configuring or using the “.tra” file directly, they can add the property “adfiles.schemaDiagnosticsFileFormat” to their “.tra” file to specify the desired format for the diagnostics file. The three valid values for the global variable 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”.

### Using Error Handling with Simple File Transfer Mode

This section discusses how the adapter handles errors for the publication and subscription service when using the Simple File Transfer mode.
Publication Service

For every subscription service that is marked inactive, or if an active subscription service returns a checksum error for a given file transfer, the publication service maintains a subscription service specific log file (.txt) in the error directory specified during configuration. This file contains information that points to the file the subscription service had an issue with.

Based on the post processing option selected, files are moved to the appropriate place for future use. If the post processing option is set to:

- Leave as is or Delete and the publication service has detected an issue with a subscription service, the file is moved to the error directory with the date and timestamp appended to it.

- Move to and the publication service has detected an issue with a subscription service, the file is moved to the completion directory and, depending on the option to add date and time, the date and timestamp are appended to the file.

Subscription Service

When a subscription service,

- Detects a checksum error
- Is restarted but is no longer active with the publisher and therefore requests activation
- Is in the process of transferring a file and receives a fresh restart signal (discovery) from the publication service

the subscription service, if currently processing a file transfer, moves the file from the working directory to the error directory.

Additionally, if a subscription service had requested activation upon restart, the service can be configured to send a special TIBCO Rendezvous message on a user-defined subject. The message will contain the subscription service name. The purpose of this message is to indicate that a registered subscription service, which was inactive is now active. Using this message, you can inspect the subscription service-specific log file under the publication service error directory for further information. The subscription service-specific log file will help you identify the files that a subscription service missed when it was inactive.
Using Global Variables

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

Variable substitution allows you to accomplish the following.

- Substitute string variables specified in the project at startup time.
- Locally define the value for a variable for a specific project. The local value takes precedence over any global value.
- Specify the value for a variable in a properties file. This overrides the project repository and values set in code, but not variables set on the command line.
- Enforce the pre-defined variables listed in Predefined Global Variables on page 105.

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

Specifying Variables Using TIBCO Designer

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

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

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

You can also add and define global variables in the tra file. The global variable should follow the convention: `adfiles.instance-name_service-name_UserId`. 
To use global variables in your project, follow these steps:

1. In the Project panel, select the Global Variables tab and click the Open Advanced Editor button.

   You now have these choices:
   
   — To assign or change a variable value, select that region and triple-click the variable. The variable expands so you can change either the variable name or the variable value. Press Enter when you’re done.
   
   — To add a new global variable group, click . Specify the name of the group, then press Enter. With the group icon selected, you can click to add variables to the group.
   
   — To add a global variable, click . A new global variable item is added to the bottom of the list. Specify the variable name and, optionally, the value. Press Enter when you’re done.

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

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

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

   A number of global variables are predefined. See Predefined Global Variables on page 105 for information. You may add definitions of any variables you need to the predefined variables.

Predefined Global Variables

The next table lists and explains the predefined global variables. Some global variables are automatically used within the system when an adapter instance is configured.

Table 2  Predefined Global Variables (Sheet 1 of 3)

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deployment</td>
<td>Defaults to the TIBCO Designer project name. This value can be any string value. This global variable is used by the system to partially define the subject name defined for a service.</td>
</tr>
</tbody>
</table>
### Table 2  Predefined Global Variables (Sheet 2 of 3)

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DirLedger</td>
<td>Specifies the path name of the TIBCO Rendezvous certified messaging ledger file. The default is the root installation directory.</td>
</tr>
<tr>
<td>DirTrace</td>
<td>Specifies the path name for log file used by the adapter. The default is the root installation directory.</td>
</tr>
<tr>
<td>Domain</td>
<td>The default value for file-based local projects is MyDomain. The value for server-based projects is the domain to which the project was saved.</td>
</tr>
<tr>
<td>HawkEnabled</td>
<td>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.</td>
</tr>
<tr>
<td>JmsProviderUrl</td>
<td>Specifies where the JMS server is located. Setting this value mostly makes sense in early stages of a project, when only one JMS server is used.</td>
</tr>
<tr>
<td>RemoteRvDaemon</td>
<td>TIBCO Rendezvous routing daemon (rvrd) to be used. See TIBCO Administrator Server Configuration Guide for details about setting up a domain using rvrd.</td>
</tr>
<tr>
<td>RvDaemon</td>
<td>TIBCO Rendezvous daemon. Sessions use this daemon to establish communication. The default value is 7500.</td>
</tr>
<tr>
<td>RvNetwork</td>
<td>TIBCO Rendezvous network. This variable need only be set on computers with more than one network interface. If specified, the TIBCO Rendezvous daemon uses that network for all outbound messages. In most cases, you can leave the default.</td>
</tr>
<tr>
<td>RvService</td>
<td>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 with other transports. Unless you are using a non-default TIBCO Rendezvous configuration, you should leave the default (7500).</td>
</tr>
</tbody>
</table>
Table 2  Predefined Global Variables (Sheet 3 of 3)

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RvaHost</td>
<td>Computer on which the TIBCO Rendezvous agent runs. This variable is only relevant if you are using the TIBCO Rendezvous Agent (rva) instead of the TIBCO Rendezvous daemon, and if you have configured a non-default setup. See TIBCO Rendezvous Administration for details about specifying the rva parameters.</td>
</tr>
<tr>
<td>RvaPort</td>
<td>TCP port where the TIBCO Rendezvous agent (rva) listens for client connection requests. See TIBCO Rendezvous Administration for details about specifying the rva parameters. Defaults to 7501.</td>
</tr>
<tr>
<td>TIBHawkDaemon</td>
<td>TIBCO Rendezvous daemon used in the TIBCO Hawk session. See TIBCO Hawk Installation and Configuration for details about this parameter.</td>
</tr>
<tr>
<td>TIBHawkNetwork</td>
<td>TIBCO Rendezvous network used by the TIBCO Hawk session. See TIBCO Hawk Installation and Configuration for details about this parameter.</td>
</tr>
<tr>
<td>TIBHawkService</td>
<td>TIBCO Rendezvous service used by the TIBCO Hawk session. See TIBCO Hawk Installation and Configuration for details about this parameter.</td>
</tr>
</tbody>
</table>
**File Recognition Methods for Publication Service**

The next table shows various configuration setting combinations for a publication service and explains the results. The first four columns each indicate a field value. Note that an empty column indicates no value is specified in the corresponding field.

<table>
<thead>
<tr>
<th>Recognition Method</th>
<th>File Prefix</th>
<th>File Extension</th>
<th>Trigger File Extension</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>By prefix + extension</td>
<td></td>
<td></td>
<td></td>
<td>All files in the input directory are processed. This option is similar to using the <em>.</em> command.</td>
</tr>
<tr>
<td>delimited</td>
<td></td>
<td></td>
<td></td>
<td>Only files in the input directory that contain delimited in their file prefix are processed. This option is similar to using a delimited.* command. For example, delimited.txt, delimited123.txt, delimited456.csv.</td>
</tr>
<tr>
<td>delimited txt</td>
<td>delimited</td>
<td>txt</td>
<td></td>
<td>Only files in the input directory that contain delimited in their file prefix and have a .txt extension are processed.</td>
</tr>
<tr>
<td>txt</td>
<td></td>
<td></td>
<td></td>
<td>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.</td>
</tr>
</tbody>
</table>
### By Trigger

<table>
<thead>
<tr>
<th>Recognition Method</th>
<th>File Prefix</th>
<th>File Extension</th>
<th>Trigger File Extension</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>By Trigger</td>
<td></td>
<td>trg</td>
<td></td>
<td>This is equivalent to specifying that when a trigger file named, anyprefix.trg is created in the input directory, the adapter will search for a file named anyprefix, since the file extension is empty. For example, if a file named delimited.trg is created in the input directory, the adapter will only parse files for a file named delimited. If a file named payments.trg is created, the adapter will search for a file named payments.</td>
</tr>
<tr>
<td>delimited</td>
<td></td>
<td>trg</td>
<td></td>
<td>When a trigger file named, delimited.trg is created, the adapter will search for a file named delimited. Similarly if a file named delimited2.trg is created, the adapter will search for a file named delimited2. In this example, a file named payments.trg will be ignored by the adapter.</td>
</tr>
<tr>
<td>delimited</td>
<td>txt</td>
<td>trg</td>
<td></td>
<td>When a trigger file named delimited.trg is created, the adapter will search for a file named delimited.txt. If a file named delimited2.txt is present, it will not be processed. The file will be processed when delimited2.trg is created. In general, if the prefix name of the trigger file matches the prefix name of the file extension, the adapter will process the file.</td>
</tr>
<tr>
<td>txt</td>
<td></td>
<td>trg</td>
<td></td>
<td>When a trigger file named, anyprefix.trg is created, the adapter will search for a file named anyprefix.txt. Similarly, when a trigger file named payments.trg is created, the adapter will search for a file named payments.txt.</td>
</tr>
</tbody>
</table>
Data Validation

The runtime adapter performs data validation. Table 3 lists the valid format and range of each data type. If a field contains data that does not conform to the format or data whose value is outside the valid range, the entire record will be discarded and an error message will be logged.

Table 3  Data Validation Formats

<table>
<thead>
<tr>
<th>Data Type</th>
<th>Format</th>
<th>Valid Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>i1</td>
<td>[whitespace][{+</td>
<td>-}][digits]</td>
</tr>
<tr>
<td>i2</td>
<td>[whitespace][{+</td>
<td>-}][digits]</td>
</tr>
<tr>
<td>i4</td>
<td>[whitespace][{+</td>
<td>-}][digits]</td>
</tr>
<tr>
<td>i8</td>
<td>[whitespace][{+</td>
<td>-}][digits]</td>
</tr>
<tr>
<td>ui1</td>
<td>[whitespace][+][digits]</td>
<td>&lt;0,255&gt;</td>
</tr>
<tr>
<td>ui2</td>
<td>[whitespace][+][digits]</td>
<td>&lt;0,65535&gt;</td>
</tr>
<tr>
<td>ui4</td>
<td>[whitespace][+][digits]</td>
<td>&lt;0,4294967295&gt;</td>
</tr>
<tr>
<td>ui8</td>
<td>[whitespace][+][digits]</td>
<td>&lt;0,9223372036854775807&gt;</td>
</tr>
<tr>
<td>r4</td>
<td>[whitespace][sign][digits][. digits][{e</td>
<td>E}[sign][digits]</td>
</tr>
<tr>
<td>r8</td>
<td>[whitespace][sign][digits][. digits][{e</td>
<td>E}[sign][digits]</td>
</tr>
</tbody>
</table>
Base Record

A base record is a record with fields that can only be of scalar type. To define base records, follow these steps:

1. Select **Show Base Records Folder** in the Configuration panel of the **ReadSchema**. This will create a folder called Base Record Folder under the **ReadSchema**.

2. Highlight this folder and drag and drop the **DelimitedFileRecord** from the Palettes panel to the Design panel.

3. Configure the DelimitedFileRecord. If you need more than one base record repeat step 2 and 3.

Container Record

A container record is a record with fields that can be of scalar or record type. For example,

Department, Engineering, Employee, ID0005, Mary, Employee, ID0006, David, Employee, ID0008, John

where Employee is a base record that contains fields of String type

(Employee, ID00005, Mary)

---

Table 3  *Data Validation Formats*

<table>
<thead>
<tr>
<th>Data Type</th>
<th>Format</th>
<th>Valid Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>DateTime</td>
<td>For the pattern, there is a pre-defined list of patterns and an option to specify user-defined patterns. If the user-defined pattern option is selected then the text box named User-Defined Pattern can be used to specify the pattern. For syntax and semantics for specifying the pattern, see the <strong>Syntax for Specifying User-defined Date and Time Patterns</strong> on page 115.</td>
<td></td>
</tr>
</tbody>
</table>
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.
Unsupported Variations of the Container Record

The current release of the adapter does not support some variations of the container record, as described in Table 6.

Table 4  Unsupported Variations of the Container Record

<table>
<thead>
<tr>
<th>Record Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Container record that consists of repeating sub-structures that cannot be defined as a base record. It is also called a shadow container record.</td>
<td>Order, ID1010, pencil, 10, eraser, 6, note pad, 12</td>
</tr>
<tr>
<td></td>
<td>Although pencil, 10, eraser, 6, and note pad, 12 are repeating sub-structures, they lack a common constant identifier that can be used as a record identifier.</td>
</tr>
<tr>
<td>Container record that consists of a base record that appears any number of times anywhere within a container record. It is also called a floating base record.</td>
<td>Department, Finance, &lt;&lt;Employee, ID0003, Mike&gt;&gt;, Department, Finance, &lt;&lt;Employee, ID0005, Mary&gt;&gt;, Department, &lt;&lt;Employee, ID0003, Mike&gt;&gt;, Finance</td>
</tr>
<tr>
<td></td>
<td>The Employee base record appears at different places within the Department container record.</td>
</tr>
<tr>
<td>Container record that consists of nested container records.</td>
<td>Order, Id2002, Item, Monitor, Customer, John, Address, Madison Ave, NY, Item, Keyboard, Customer, Joe, Address, Pike Ave, NJ</td>
</tr>
<tr>
<td></td>
<td>If Address is a base record for Customer, it is considered a nested base record.</td>
</tr>
<tr>
<td>Positional container records.</td>
<td></td>
</tr>
<tr>
<td>Container record that consists of base records that may be partially formed.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>If the original Schema definition corresponds to the following structure,</td>
</tr>
<tr>
<td></td>
<td>Order, Id2002, Item, Monitor, Customer, John</td>
</tr>
<tr>
<td></td>
<td>with Item and Customer as the base records, but the actual line in the file 9 as,</td>
</tr>
<tr>
<td></td>
<td>Order, Id2002, Item, Customer, John</td>
</tr>
<tr>
<td></td>
<td>it is not supported.</td>
</tr>
</tbody>
</table>
Table 4  Unsupported Variations of the Container Record (Cont’d)

<table>
<thead>
<tr>
<th>Record Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>A container record which contains a base record that is not contiguous.</td>
<td>Order, Id2002, Item, Monitor, Customer, John, Item, Keyboard, Customer, Joe</td>
</tr>
<tr>
<td></td>
<td>Here the container record is ordered and the base records are Customer and Item. In this example, the Item record repetition is not contiguous.</td>
</tr>
</tbody>
</table>

The representation of the fields of type record on the wire schema is very similar to that of a file record that is a child. They both are represented as a sequence.
Syntax for Specifying User-defined Date and Time Patterns

Table 5 lists the date and time symbols that can be used as patterns to interpret and generate date and time in various formats.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
<th>Presentation</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>G</td>
<td>era designator</td>
<td>Text)</td>
<td>AD</td>
</tr>
<tr>
<td>y</td>
<td>year</td>
<td>(Number)</td>
<td>1996</td>
</tr>
<tr>
<td>Y</td>
<td>year/week of year</td>
<td>(Number)</td>
<td>1996</td>
</tr>
<tr>
<td>M</td>
<td>month in year</td>
<td>(Text &amp; Number)</td>
<td>July &amp; 07</td>
</tr>
<tr>
<td>d</td>
<td>day in month</td>
<td>(Number)</td>
<td>10</td>
</tr>
<tr>
<td>h</td>
<td>hour in am/pm (1~12)</td>
<td>(Number)</td>
<td>12</td>
</tr>
<tr>
<td>H</td>
<td>hour in day (0~23)</td>
<td>(Number)</td>
<td>0</td>
</tr>
<tr>
<td>m</td>
<td>minute in hour</td>
<td>(Number)</td>
<td>30</td>
</tr>
<tr>
<td>s</td>
<td>second in minute</td>
<td>(Number)</td>
<td>55</td>
</tr>
<tr>
<td>S</td>
<td>millisecond</td>
<td>(Number)</td>
<td>978</td>
</tr>
<tr>
<td>E</td>
<td>day of week</td>
<td>(Text)</td>
<td>Tuesday</td>
</tr>
<tr>
<td>e</td>
<td>day of week/local (1~7)</td>
<td>(Number)</td>
<td>2</td>
</tr>
<tr>
<td>D</td>
<td>day of year</td>
<td>(Number)</td>
<td>189</td>
</tr>
<tr>
<td>F</td>
<td>day of week in month</td>
<td>(Number)</td>
<td>2 (2nd Wed in July)</td>
</tr>
<tr>
<td>w</td>
<td>week in year</td>
<td>(Number)</td>
<td>27</td>
</tr>
<tr>
<td>W</td>
<td>week in month</td>
<td>(Number)</td>
<td>2</td>
</tr>
<tr>
<td>a</td>
<td>am/pm marker</td>
<td>(Text)</td>
<td>PM</td>
</tr>
<tr>
<td>k</td>
<td>hour in day (1~24)</td>
<td>(Number)</td>
<td>24</td>
</tr>
<tr>
<td>K</td>
<td>hour in am/pm (0~11)</td>
<td>(Number)</td>
<td>0</td>
</tr>
</tbody>
</table>
### Table 5  Date and Time Symbols (Cont’d)

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
<th>Presentation</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>z</code></td>
<td>time zone</td>
<td>(Text)</td>
<td>Pacific Standard Time</td>
</tr>
<tr>
<td><code>''</code></td>
<td>escape for text</td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>&quot;&quot;</code></td>
<td>single quote</td>
<td><code>'</code></td>
<td></td>
</tr>
</tbody>
</table>
## Locales Supported for Date and Time

Table 6  Country and Language Codes (Sheet 1 of 5)

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Country/Language</th>
</tr>
</thead>
<tbody>
<tr>
<td>ar_AE</td>
<td>Arabic United Arab Emirates</td>
</tr>
<tr>
<td>ar_BH</td>
<td>Arabic Bahrain</td>
</tr>
<tr>
<td>ar_DZ</td>
<td>Arabic Algeria</td>
</tr>
<tr>
<td>ar_EG</td>
<td>Arabic Egypt</td>
</tr>
<tr>
<td>ar_IQ</td>
<td>Arabic Iraq</td>
</tr>
<tr>
<td>ar_JO</td>
<td>Arabic Jordan</td>
</tr>
<tr>
<td>ar_KW</td>
<td>Arabic Kuwait</td>
</tr>
<tr>
<td>ar_LB</td>
<td>Arabic Lebanon</td>
</tr>
<tr>
<td>ar_LY</td>
<td>Arabic Libya</td>
</tr>
<tr>
<td>ar_MA</td>
<td>Arabic Morocco</td>
</tr>
<tr>
<td>ar_OM</td>
<td>Arabic Oman</td>
</tr>
<tr>
<td>ar_QA</td>
<td>Arabic Qatar</td>
</tr>
<tr>
<td>ar_SA</td>
<td>Arabic Saudi Arabia</td>
</tr>
<tr>
<td>ar_SD</td>
<td>Arabic Sudan</td>
</tr>
<tr>
<td>ar_SY</td>
<td>Arabic Syria</td>
</tr>
<tr>
<td>ar_TN</td>
<td>Arabic Tunisia</td>
</tr>
<tr>
<td>ar_YE</td>
<td>Arabic Yemen</td>
</tr>
<tr>
<td>be_BY</td>
<td>Byelorussian Byelorussia</td>
</tr>
<tr>
<td>bg_BG</td>
<td>Bulgarian Bulgaria</td>
</tr>
<tr>
<td>ca_ES</td>
<td>Catalan Spain</td>
</tr>
</tbody>
</table>
Table 6  Country and Language Codes (Sheet 2 of 5)

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Country/Language</th>
</tr>
</thead>
<tbody>
<tr>
<td>ca_ES_EURO</td>
<td>Catalan Spain</td>
</tr>
<tr>
<td>cs_CZ</td>
<td>Czech Republic</td>
</tr>
<tr>
<td>da_DK</td>
<td>Danish Denmark</td>
</tr>
<tr>
<td>de_AT</td>
<td>German Austria</td>
</tr>
<tr>
<td>de_AT_EURO</td>
<td>German Austria</td>
</tr>
<tr>
<td>de_CH</td>
<td>German Switzerland</td>
</tr>
<tr>
<td>de_DE</td>
<td>German Germany</td>
</tr>
<tr>
<td>de_DE_EURO</td>
<td>German Germany</td>
</tr>
<tr>
<td>de_LU</td>
<td>German Luxembourg</td>
</tr>
<tr>
<td>de_LU_EURO</td>
<td>German Luxembourg</td>
</tr>
<tr>
<td>el_GR</td>
<td>Greek Greece</td>
</tr>
<tr>
<td>en_AU</td>
<td>English Australia</td>
</tr>
<tr>
<td>en_BE</td>
<td>English Belgium</td>
</tr>
<tr>
<td>en_CA</td>
<td>English Canada</td>
</tr>
<tr>
<td>en_GB</td>
<td>English United Kingdom</td>
</tr>
<tr>
<td>en_IE</td>
<td>English Ireland</td>
</tr>
<tr>
<td>en_IE_EURO</td>
<td>English Ireland</td>
</tr>
<tr>
<td>en_NZ</td>
<td>English New Zealand</td>
</tr>
<tr>
<td>en_US</td>
<td>English United States</td>
</tr>
<tr>
<td>en_ZA</td>
<td>English South Africa</td>
</tr>
<tr>
<td>es_AR</td>
<td>Spanish Argentina</td>
</tr>
<tr>
<td>es_BO</td>
<td>Spanish Bolivia</td>
</tr>
<tr>
<td>es_CL</td>
<td>Spanish Chile</td>
</tr>
</tbody>
</table>
### Table 6  Country and Language Codes (Sheet 3 of 5)

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Country/Language</th>
</tr>
</thead>
<tbody>
<tr>
<td>es_CO</td>
<td>Spanish Colombia</td>
</tr>
<tr>
<td>es_CR</td>
<td>Spanish Costa Rica</td>
</tr>
<tr>
<td>es_DO</td>
<td>Spanish Dominican Republic</td>
</tr>
<tr>
<td>es_EC</td>
<td>Spanish Ecuador</td>
</tr>
<tr>
<td>es_ES</td>
<td>Spanish Spain</td>
</tr>
<tr>
<td>es_ES_EURO</td>
<td>Spanish Spain</td>
</tr>
<tr>
<td>es_GT</td>
<td>Spanish Guatemala</td>
</tr>
<tr>
<td>es_HN</td>
<td>Spanish Honduras</td>
</tr>
<tr>
<td>es_MX</td>
<td>Spanish Mexico</td>
</tr>
<tr>
<td>es_NI</td>
<td>Spanish Nicaragua</td>
</tr>
<tr>
<td>es_PA</td>
<td>Spanish Panama</td>
</tr>
<tr>
<td>es_PE</td>
<td>Spanish Peru</td>
</tr>
<tr>
<td>es_PR</td>
<td>Spanish Puerto Rico</td>
</tr>
<tr>
<td>es_PY</td>
<td>Spanish Paraguay</td>
</tr>
<tr>
<td>es_SV</td>
<td>Spanish El Salvador</td>
</tr>
<tr>
<td>es_UY</td>
<td>Spanish Uruguay</td>
</tr>
<tr>
<td>es_VE</td>
<td>Spanish Venezuela</td>
</tr>
<tr>
<td>et_EE</td>
<td>Estonian Estonia</td>
</tr>
<tr>
<td>fi_FI</td>
<td>Finnish Finland</td>
</tr>
<tr>
<td>fi_FI_EURO</td>
<td>Finnish Finland</td>
</tr>
<tr>
<td>fr_BE</td>
<td>French Belgium</td>
</tr>
<tr>
<td>fr_BE_EURO</td>
<td>French Belgium</td>
</tr>
<tr>
<td>fr_CA</td>
<td>French Canada</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Country/Language</td>
</tr>
<tr>
<td>--------------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td>fr_CH</td>
<td>French Switzerland</td>
</tr>
<tr>
<td>fr_FR</td>
<td>French France</td>
</tr>
<tr>
<td>fr_FR_EURO</td>
<td>French France</td>
</tr>
<tr>
<td>fr_LU</td>
<td>French Luxembourg</td>
</tr>
<tr>
<td>fr_LU_EURO</td>
<td>French Luxembourg</td>
</tr>
<tr>
<td>hr_HR</td>
<td>Croatian Croatia</td>
</tr>
<tr>
<td>hu_HU</td>
<td>Hungarian Hungary</td>
</tr>
<tr>
<td>is_IS</td>
<td>Icelandic Iceland</td>
</tr>
<tr>
<td>it_CH</td>
<td>Italian Switzerland</td>
</tr>
<tr>
<td>it_IT</td>
<td>Italian Italy</td>
</tr>
<tr>
<td>it_IT_EURO</td>
<td>Italian Italy</td>
</tr>
<tr>
<td>iw_IL</td>
<td>Hebrew Israel</td>
</tr>
<tr>
<td>ja_JP</td>
<td>Japanese Japan</td>
</tr>
<tr>
<td>ko_KR</td>
<td>Korean Korea</td>
</tr>
<tr>
<td>lt_LT</td>
<td>Lithuanian Lithuania</td>
</tr>
<tr>
<td>lv_LV</td>
<td>Latvian Latvia</td>
</tr>
<tr>
<td>mk_MK</td>
<td>Macedonian Macedonia</td>
</tr>
<tr>
<td>nl_BE</td>
<td>Dutch Belgium</td>
</tr>
<tr>
<td>nl_BE_EURO</td>
<td>Dutch Belgium</td>
</tr>
<tr>
<td>nl_NL</td>
<td>Dutch Netherlands</td>
</tr>
<tr>
<td>nl_NL_EURO</td>
<td>Dutch Netherlands</td>
</tr>
<tr>
<td>no_NO</td>
<td>Norwegian (Nynorsk) Norway</td>
</tr>
<tr>
<td>no_NO_NY</td>
<td>Norwegian (Bokmål) Norway</td>
</tr>
</tbody>
</table>
Table 6  Country and Language Codes (Sheet 5 of 5)

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Country/Language</th>
</tr>
</thead>
<tbody>
<tr>
<td>pl_PL</td>
<td>Polish Poland</td>
</tr>
<tr>
<td>pt_BR</td>
<td>Portuguese Brazil</td>
</tr>
<tr>
<td>pt_PT</td>
<td>Portuguese Portugal</td>
</tr>
<tr>
<td>pt_PT_EURO</td>
<td>Portuguese Portugal</td>
</tr>
<tr>
<td>ro_RO</td>
<td>Romanian Romania</td>
</tr>
<tr>
<td>ru_RU</td>
<td>Russian Russia</td>
</tr>
<tr>
<td>sh_YU</td>
<td>Serbo-Croatian Yugoslavia</td>
</tr>
<tr>
<td>sk_SK</td>
<td>Slovakian Slovakia</td>
</tr>
<tr>
<td>sl_SI</td>
<td>Slovenian Slovenia</td>
</tr>
<tr>
<td>sq_AL</td>
<td>Albanian Albania</td>
</tr>
<tr>
<td>sr_YU</td>
<td>Serbian (Cyrillic) Yugoslavia</td>
</tr>
<tr>
<td>sv_SE</td>
<td>Swedish Sweden</td>
</tr>
<tr>
<td>th_TH</td>
<td>Thai Thailand</td>
</tr>
<tr>
<td>tr_TR</td>
<td>Turkish Turkey</td>
</tr>
<tr>
<td>uk_UA</td>
<td>Ukranian Ukraine</td>
</tr>
<tr>
<td>vi_VN</td>
<td>Vietnamese Vietnam</td>
</tr>
<tr>
<td>zh_CN</td>
<td>Chinese (Simplified) China</td>
</tr>
<tr>
<td>zh_HK</td>
<td>Chinese Hong Kong</td>
</tr>
<tr>
<td>zh_TW</td>
<td>Chinese (Traditional) Taiwan</td>
</tr>
</tbody>
</table>
Using Trigger Messages

A Rendezvous or JMS trigger message can contain the USERID, inputDirectory, fileName, filePrefix, and fileExtension properties. When triggering a publication service using a message that has these properties, the JMS trigger message properties will override the corresponding properties that you set at design time. The changes made during runtime do not change the configuration in the repository.

If the By File Name item is selected in the Recognition Method drop-down list, the fileName property can only contain filename and not ICU regular expressions.

When an ICU regular expression is used in trigger messages, if the ICU regular expression contains a comma, the comma needs to be escaped.

For example, the trigger message “fileName=go{1,5}gle\.txt” contains a comma. This comma is not a property separator, so an "escaping" comma in front of the comma needs to be added, for example, "fileName=go{1,,5}gle\.txt".

You may send multiple trigger messages at one time, each with its own has its properties. After the publication service gets the trigger messages, the messages will be queued up. The publication service will process them one by one.

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

TIBCO ActiveMatrix Adapter for Files will make the best attempt to persist the trigger messages but there is no guarantee.

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. There are no spaces between properties.
  
  For example, "inputDirectory=reader/input,fileName=foo.txt".

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

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

- If the trigger message is empty, the publication service uses the properties that have been defined at design time.

- The input directory of the publication service can be changed at runtime 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 drop-down list for the publication service, it can be changed at runtime 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 filename "abc.txt" plus one.

  — JMS message:
    
    "fileName=abc.txt"

If a JMS trigger message contains no file name property, the publication service will use the file name specified in the last JMS trigger message, or the file name configured in TIBCO Designer 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 the By prefix + extension item in the Recognition Method drop-down list for the publication service, it can be changed at runtime 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"

• If the publication service gets an invalid trigger message, for example “fileName=foo.txt,=John” or “reader/input,fileName=foo.txt”, the publication service will discard the message.

• The publication service can receive more than one trigger message at a time but will process them one by one. Trigger messages waiting to be processed are saved. If the publication service stops and then restarts, it will continue to process the saved trigger messages.
Using Selective Routing Over JMS

When using JMS trigger messages to trigger a publication service, you can send different file(s) to selected subscriber(s) dynamically using Selective Routing.

Syntax

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

**Tip**

The USERID, inputDirectory, fileName, filePrefix, and fileExtension properties are reserved for TIBCO ActiveMatrix Adapter for Files. The message selector properties that you define must not use the same names. All the properties that you define will be included in the properties section of every outgoing JMS data message.

Configuration

To use selective routing, you need to specify the JMS message selector on the subscribers, you want to send messages to. You can specify message selectors in TIBCO Designer.

**Setting JMS Message Selectors in TIBCO Designer**

To do this, follow these steps:

1. Start the EMS server.
2. Open the project containing the subscription services that you are going to use in TIBCO Designer.
   a. In the Sessions folder of your subscription service for example, mysub1, expand the DefaultJMSTopicSession tree and select mysub1EndPoint.
   b. In the Configuration tab of the subscription service endpoint, 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 SalesVolume='ALL'), as shown in Figure 11.

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 a subscription service named mysub2.

Figure 11 Using TIBCO Designer to Specify JMS Message Selector

4. Select the Simple File Transfer item in the Transfer Mode drop-down list and use non-ECM for the subscription services.

5. Start the subscription service and connect to the EMS server.

6. Start a publication service and connect to the EMS server. The publication service uses the Simple File Transfer and non-ECM mode. The polling method is JMS (topic) message.

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

After receiving the trigger message, the publication service will start transferring the files to the subscription services specified in the JMS message selectors in step 3. The fileName, Branch, SalesUpper, and SalesLower properties will be included in the outgoing data messages. In this example, mysub1 matches the conditions in the JMS message. The publication service will route the messages to mysub1.

See the TIBCO Enterprise Message documentation for details about configuring EMS servers, creating EMS routes, constructing JMS messages, and specifying JMS message selectors.
Dynamically Changing Output File Names at Runtime

In the Record Transfer mode, the subscription service by default uses the filenames configured at the design time when creating the output files to store the incoming messages. However sometimes it's desirable to have the subscription service use different filenames dynamically from what's configured at the design time when creating the output file. This can be achieved by using the Business Document data format in the data messages and embedding the new filenames in the data messages. One scenario of such application is when the publisher is a publication service and you want the subscription service to create the output file using the exact same filenames when files are being published.

To do this, follow these steps:

1. Select **Business document** in the Data Format drop-down list in the Advanced tab of the 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** in the Wip Creation Mode drop-down list in the Configuration tab of the subscription service.
4. Add `adfiles.useBDEmbeddedFileName ON` to the `your_runtime.tra` file or create a global variable named `adfiles.useBDEmbeddedFileName`. The value of the variable is `ON`. This is for the subscription service.

If using TIBCO Administrator to deploy and run the adapter, create an `adfiles.useBDEmbeddedFileName` global variable and set the value to `ON`. You also need to check the **Service** checkbox in the global variable dialog. When creating the EAR file, you must also check the **Include all service level global variable** checkbox.
Instance encoding for the repository needs to be correctly set for the adapter to interoperate with other TIBCO ActiveEnterprise applications.

**File and Directory Encodings**

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

This parameter has to be set appropriately at the design time based on the platform encoding the runtime adapter will run on.

**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 will cater 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 the instance encoding**

- For Remote Repository:
  
  Use the remote repository initialization file (`tibcoadmin<domain>.tra`) and set the following property `repo.encoding` to UTF-8 or Latin-1.

- For Local Repository:
  
  If an adapter configuration is saved in a local project, the inter-communication 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 to be identical. The encoding value is set on the root project folder, under the `Project`
Settings tab. Select the appropriate repository instance encoding (ISO-8859-1 or UTF-8).

Click **Apply** to save the settings.
Business Event Messages

TIBCO ActiveMatrix Adapter for Files can provide business-event level notifications for TIBCO Business Events.

To configure TIBCO ActiveMatrix Adapter for Files to create business-event messages, perform these steps:

1. Create a Network Sink in TIBCO Designer:
   a. Drag the Network Sink object from the Adapter Resources palette.
   b. 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.EEMEnabled** property:
   - If you are using TIBCO Administrator, from the Global Variables tab, add a new global variable named ***adfiles.EEMEnabled*** 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, the publication service sends or logs a status message based on the log roles selected. Similarly, at the end of each file transfer, the subscription service sends or logs a status message based on the log roles selected. The status message contains the following information:

Table 7 Information for the Status Message

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>fileName</td>
<td>is the name of the file that is being transferred.</td>
</tr>
<tr>
<td>hostName</td>
<td>is the name of the host on which the publication service is running.</td>
</tr>
<tr>
<td>instanceName</td>
<td>is the name of the adapter configuration.</td>
</tr>
<tr>
<td>serviceName</td>
<td>is the name of the publication service.</td>
</tr>
<tr>
<td>trackingId</td>
<td>is the unique tracking ID generated for that polling instance every time the polling trigger activates the publisher. Note: You can set the tracking ID if you have selected TIBCO Rendezvous message as the polling method. For details, Using Trigger Messages on page 122.</td>
</tr>
<tr>
<td>fileTransferDuration</td>
<td>is the time taken to transfer the file.</td>
</tr>
<tr>
<td>currentDateT</td>
<td>is the date and time at the end of the file transfer.</td>
</tr>
<tr>
<td>status</td>
<td>In ECM, 0 indicates that all the registered subscribers successfully participated in the file transfer. 1 indicates that one or more registered subscribers did not successfully participate in the file transfer. In non-ECM, this value is set to n/a, or not applicable.</td>
</tr>
</tbody>
</table>

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

To send the status message as an exclusive TIBCO Rendezvous message on a user-defined subject, create a network sink, created a role named ecmPubStatus for a publication service or created a role named ecmSubStatus for a subscription service and then associate the role with the network sink. For details about creating a network sink, see the TIBCO Designer documentation.
To create user-defined roles such as `ecmPubStatus` or `ecmSubStatus` and to use network sink, advanced logging options for the adapter configuration should be turned on in TIBCO Designer. Otherwise, to log the status message, turn on `debugRole` for the sinks that are currently being used.
In previous releases, you could only start one instance of a particular publication service. From release 6.2.0, you can run multiple instances of a publication service to process separate files in parallel.

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

Before a publication service instance processes an input file, the publication service instance locks the file by creating a `.lck` file. Therefore other publication service instances will 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. If **Leave as is** is selected, the error message "Notice: Load Balance not support the post processing mode leave-as-is, please correct it" will appear after starting the publication service.

Each publication service instance should have one unique message subject.

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

Load Balancing mode supports the Checkpoint Restart feature. See **Checkpoint Restart on page 58** for more information.
Chapter 6  Monitoring the Adapter Using TIBCO Hawk

This chapter explains how to use TIBCO Hawk microagents to monitor and manage the adapter.

Topics

- Overview, page 136
- Starting TIBCO Hawk Software, page 137
- The Auto-discovery Process, page 138
- Invoking Microagent Methods, page 139
- Standard and Class Microagents, page 142
- Custom Microagents, page 145
Overview

TIBCO Hawk is a sophisticated tool for enterprise-wide monitoring and managing of all distributed applications and systems. System administrators can use it to monitor adapters in a wide area network of any size. TIBCO Hawk can be configured to monitor system and adapter parameters and to take actions when predefined conditions occur. The actions include: sending alarms that are graphically displayed in the TIBCO Hawk display, sending email, paging, running executables, or modifying the behavior of a managed adapter.

Unlike other monitoring applications, TIBCO Hawk relies on a purely distributed intelligent agent architecture using publish or subscribe to distribute alerts. TIBCO Hawk uses TIBCO Rendezvous for all messaging and thus gains the benefits and scalability of the TIBCO Rendezvous features of publish or subscribe, subject name addressing, interest-based routing, and reliable multicast.

TIBCO Hawk is a purely event-based system that uses alerts. The agents are configured with rules that instruct them on everything from what and how to monitor to what actions to take when problems are discovered. Therefore, the workload is fully distributed throughout the enterprise. Every agent is autonomous in that it does not depend on other components to perform its functions.

The TIBCO Hawk Enterprise Monitor consists of these components:

- **Display** — A GUI front end that displays alarms and provides editors to create rule bases, create tests, view messages, and invoke microagents to request information or initiate an action.

- **Agents** — Intelligent processes that perform monitoring and take actions as defined in rules.

- **Rulebases** — Rules that are loaded by agents to determine agent behavior.

- **Application Management Interface (AMI)** — Manages network applications via TIBCO Rendezvous and supports communication between a network application and monitoring TIBCO Hawk agents, including the ability to examine application variables, invoke methods, and monitor system performance.

- **Microagents** — Feeds information back to TIBCO Hawk and expose action methods to rulebases.

For more information, see the TIBCO Hawk documentation.
Starting TIBCO Hawk Software

The TIBCO Hawk agent can be configured to start automatically during the system boot cycle. See the TIBCO Hawk documentation for information about starting TIBCO Hawk and how to start the TIBCO Hawk Display.
The Auto-discovery Process

After you start an instance of TIBCO Hawk Display, it continually discovers machines running TIBCO Hawk agents on your network. Container icons are created for each agent, and arranged hierarchically in clusters. By default, agent icons are clustered according to subnets.

At first, the Agents container is empty. Its counter displays a value of zero and, on the right, the Discovered counter is also at zero. Both icons are initially green in color to show that no alerts, or warning messages, are in effect. As agents are discovered, the counters increment to reflect the current number of discovered agents:

Monitored network nodes are arranged in a hierarchical tree of containers. Clicking a container in the left panel displays nested items on the right.

Icon colors change to reflect the highest level of alert found on discovered agents. For explanations of icon elements and characteristics, see TIBCO Hawk Administrator’s Guide.
Invoking Microagent Methods

A set of default microagents, platform-specific and platform-independent, is loaded when a TIBCO Hawk agent is started. When you install and start the TIBCO ActiveMatrix Adapter for Files, microagents for the adapter are dynamically added to the local agent.

The following steps describe how to invoke a microagent method by specifying a microagent, method name, and optional method arguments.

To invoke a microagent method on a TIBCO Hawk agent:

1. In the TIBCO Hawk Display, right-click on the agent icon and select **Get Microagents**.

   If TIBCO Hawk security is implemented on your system and you do not have access to microagents on this agent, an error dialog displays. Select another agent, or contact your system administrator to obtain access.

2. The Microagents, Methods and Arguments dialog displays. The panel on the upper left lists microagents you can access on the current agent.

   ![Microagents, Methods and Arguments dialog](image)

   This dialog has two modes, Invoke and Subscribe. Invoking a method immediately returns a single set of current results. Subscribing provides updates of current results at regular intervals. Radio buttons at the bottom of the dialog control these modes.

3. Click a microagent name, such as **Self**, to display a list of associated methods and text descriptions in the panels below.
4. Click the name of the method to invoke, such as `getComponentInfo`.

If the method accepts arguments, fields for each argument display in the upper right panel. Detailed help text displays in the lower panel.

5. Specify any arguments for the method invocation.

6. Verify that the Invoke radio button is selected.

7. Click the **Invoke** button to invoke the selected method.

The Invocation Results dialog displays the results returned by the method.

8. Click **Done** to close the dialog.
These steps describe how to interactively invoke a microagent method and receive a single set of results in TIBCO Hawk Display. You can also use a microagent method as the data source of a TIBCO Hawk rule. Rules automatically receive method results, apply tests to evaluate them, then take action if necessary. For more information on building TIBCO Hawk rules and rule bases, see *TIBCO Hawk Administrator's Guide*.
Standard and Class Microagents

The following table lists each standard and class microagent method available for the adapter and the page on which the method is explained.

Table 8  Standard and Class Microagent Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>activateTraceRole()</td>
<td>Activates a mapping of a role to a sink at runtime.</td>
<td>146</td>
</tr>
<tr>
<td>deactivateTraceRole()</td>
<td>Deactivates a mapping of a roles to sinks at runtime.</td>
<td>147</td>
</tr>
<tr>
<td>getActivityStatisticsByOperation(Operation)</td>
<td>Returns statistics about one operation.</td>
<td>148</td>
</tr>
<tr>
<td>getActivityStatisticsBySchema(SchemaName)</td>
<td>Returns statistics about any activities on a particular object or schema.</td>
<td>149</td>
</tr>
<tr>
<td>getActivityStatisticsByService(ServiceName)</td>
<td>Returns statistics about the data handled by a particular adapter service since it was started.</td>
<td>150</td>
</tr>
<tr>
<td>getAdapterServicesInformation()</td>
<td>Returns information about the services implemented by this adapter.</td>
<td>151</td>
</tr>
<tr>
<td>getComponents()</td>
<td>Returns information about the publisher, subscriber and IODescriptor.</td>
<td>152</td>
</tr>
<tr>
<td>getConfig()</td>
<td>Returns basic configuration information. More specific information is accessed by the more specific methods.</td>
<td>153</td>
</tr>
<tr>
<td>getConfigProperties()</td>
<td>Returns all attributes and elements for the given repository object.</td>
<td>154</td>
</tr>
<tr>
<td>getDocumentDelay()</td>
<td>Returns the document delay setting for a given publication service.</td>
<td>155</td>
</tr>
<tr>
<td>getHostInformation()</td>
<td>Returns standard and extended application information.</td>
<td>156</td>
</tr>
<tr>
<td>getPollingInterval()</td>
<td>Returns the current polling interval setting.</td>
<td>157</td>
</tr>
<tr>
<td>Method (Cont’d)</td>
<td>Description (Cont’d)</td>
<td>Page</td>
</tr>
<tr>
<td>---------------------------------------</td>
<td>--------------------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td><code>getQueueStatistics()</code></td>
<td>Returns the current count of elements in any internal queue used by the adapter.</td>
<td>158</td>
</tr>
<tr>
<td><code>getRvConfig()</code></td>
<td>Returns information about all TIBCO Rendezvous sessions defined.</td>
<td>159</td>
</tr>
<tr>
<td><code>getStatus()</code></td>
<td>Returns general status information, such as the number of TIBCO Rendezvous messages received and published, the number of errors since the last call, the PID of the application, and more.</td>
<td>160</td>
</tr>
<tr>
<td><code>getTraceSinks()</code></td>
<td>Returns information about sinks to which traces currently go.</td>
<td>161</td>
</tr>
<tr>
<td><code>getVersion()</code></td>
<td>Returns the configuration ID, application name, version, and date for this adapter instance.</td>
<td>162</td>
</tr>
<tr>
<td><code>_onUnsolicitedMsg()</code></td>
<td>Displays alert messages sent to the current adapter.</td>
<td>163</td>
</tr>
<tr>
<td><code>preRegisterListener()</code></td>
<td>Preregisters an anticipated listener.</td>
<td>164</td>
</tr>
<tr>
<td><code>resetActivityStatistics()</code></td>
<td>Resets all the counts for the activity statistics.</td>
<td>165</td>
</tr>
<tr>
<td><code>reviewLedger()</code></td>
<td>Returns information retrieved from the ledger file of a certified messaging session for a publisher adapter.</td>
<td>166</td>
</tr>
<tr>
<td><code>setDocumentDelay()</code></td>
<td>Sets the document delay for a publication service.</td>
<td>168</td>
</tr>
<tr>
<td><code>setPollingInterval()</code></td>
<td>Sets the polling interval for the publication service.</td>
<td>169</td>
</tr>
<tr>
<td><code>setTraceSinks()</code></td>
<td>Adds a role or changes the file limit of a previously specified sink.</td>
<td>170</td>
</tr>
<tr>
<td><code>stopApplicationInstance()</code></td>
<td>Stops the running adapter instance.</td>
<td>171</td>
</tr>
</tbody>
</table>
### Table 8  Standard and Class Microagent Methods (Cont’d)

<table>
<thead>
<tr>
<th>Method (Cont’d)</th>
<th>Description (Cont’d)</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>unRegisterListener()</code></td>
<td>Unregisters a currently preregistered listener.</td>
<td>172</td>
</tr>
</tbody>
</table>
Custom Microagents

The following table lists each custom microagent method that is available for the adapter, and the page on which the method is explained.

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>getDocumentDelay()</td>
<td>Returns the document delay setting for a given publication service.</td>
<td>155</td>
</tr>
<tr>
<td>setDocumentDelay()</td>
<td>Sets the document delay for a publication service.</td>
<td>168</td>
</tr>
<tr>
<td>getPollingInterval()</td>
<td>Returns the current polling interval setting.</td>
<td>157</td>
</tr>
<tr>
<td>setPollingInterval()</td>
<td>Sets the polling interval for the publication service.</td>
<td>169</td>
</tr>
<tr>
<td>getActivityStatisticsByOperation(Operation)</td>
<td>Returns statistics about one operation.</td>
<td>148</td>
</tr>
<tr>
<td>getActivityStatisticsBySchema(SchemaName)</td>
<td>Returns statistics about any activities on a particular object or schema.</td>
<td>149</td>
</tr>
<tr>
<td>getActivityStatisticsByService(ServiceName)</td>
<td>Return information about the services implemented by this adapter.</td>
<td>150</td>
</tr>
<tr>
<td>resetActivityStatistics()</td>
<td>Reset all the counts for the activity statistics.</td>
<td>165</td>
</tr>
</tbody>
</table>

The `getActivityStatisticsBySchema(SchemaName)`, `getActivityStatisticsByOperation(Operation)`, `getActivityStatisticsByService(ServiceName)`, and `resetActivityStatistics()` methods are deprecated. You must use the standard microagent to get equivalent methods. When the value of `adfiles.addCustomHawkMethodsToStdMAgent` is set to `ON` in the TIBCO Runtime Agent file, these methods get added to the standard microagent.
activateTraceRole()

Activates a mapping of a role to a sink at runtime. This replaces the now-deprecated `setTraceSink()` TIBCO Hawk method.

<table>
<thead>
<tr>
<th>Input Parameters</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Role Name</td>
<td>string</td>
<td>Name of the role to activate.</td>
</tr>
<tr>
<td>Sink Name</td>
<td>string</td>
<td>Name of the sink for which to activate the role.</td>
</tr>
</tbody>
</table>
Deactivates a mapping of a roles to sinks at runtime.

<table>
<thead>
<tr>
<th>Input Parameters</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Role Name</td>
<td>string</td>
<td>Name of the role to activate.</td>
</tr>
<tr>
<td>Sink Name</td>
<td>string</td>
<td>Name of the sink for which to activate the role.</td>
</tr>
</tbody>
</table>
getActivityStatisticsByOperation(Operation)

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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operation</td>
<td>string</td>
<td>Type of operation - read or write.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Returns</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Service Name</td>
<td>string</td>
<td>Name of the service that is associated with the specified operation.</td>
</tr>
<tr>
<td>Total</td>
<td>string</td>
<td>Total number of objects processed for this schema for a publication service.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Total number of objects received for this schema for a subscription service.</td>
</tr>
<tr>
<td>Success</td>
<td>string</td>
<td>The number of objects that were successfully identified for this schema which will be published or written to a file.</td>
</tr>
<tr>
<td>Error</td>
<td>string</td>
<td>The number of objects that were identified for this schema but were not published because the header of the schema failed validation for the publication service, or was written to a file because the schema was not associated with the subscriber for a subscription service.</td>
</tr>
</tbody>
</table>
getActivityStatisticsBySchema(SchemaName)

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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Schema Name</td>
<td>string</td>
<td>Name of the schema.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Returns</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Service Name</td>
<td>string</td>
<td>Name of the service that is associated with the specified schema.</td>
</tr>
<tr>
<td>Total</td>
<td>string</td>
<td>Total number of objects processed for this schema for a publication service.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Total number of objects received for this schema for a subscription service.</td>
</tr>
<tr>
<td>Success</td>
<td>string</td>
<td>The number of objects that were successfully identified for this schema which will be published or written to a file.</td>
</tr>
<tr>
<td>Error</td>
<td>string</td>
<td>The number of objects that were identified for this schema but were not published because the header of the schema failed validation for the publication service, or was written to a file because the schema was not associated with the subscriber for a subscription service.</td>
</tr>
</tbody>
</table>
### `getActivityStatisticsByService(ServiceName)`

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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Service Name</td>
<td>string</td>
<td>Name of the service.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Returns</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operation</td>
<td>string</td>
<td>Type of operation that the service performs.</td>
</tr>
<tr>
<td>Schema Name</td>
<td>string</td>
<td>Name of the schema that is associated with the service.</td>
</tr>
<tr>
<td>Total</td>
<td>string</td>
<td>Number of objects processed for this schema for a publication service.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Number of objects received for this schema for a subscription service.</td>
</tr>
<tr>
<td>Success</td>
<td>string</td>
<td>The number of objects that were successfully identified for this schema which will be published or written to a file.</td>
</tr>
<tr>
<td>Error</td>
<td>string</td>
<td>The number of objects that were identified for this schema but were not published because the header of the schema failed validation for the publication service, or was written to a file because the schema was not associated with the subscriber for a subscription service.</td>
</tr>
</tbody>
</table>
getAdapterServicesInformation()

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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Service Name</td>
<td>string</td>
<td>Name of the service from which to get information. Default is ALL.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Returns</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Line</td>
<td>integer</td>
<td>Sequential row number.</td>
</tr>
<tr>
<td>Service Name</td>
<td>string</td>
<td>Name of the service as defined at design-time.</td>
</tr>
<tr>
<td>Endpoint Name</td>
<td>string</td>
<td>Name of the endpoint used for this service.</td>
</tr>
<tr>
<td>Type</td>
<td>string</td>
<td>Type of the endpoint, for example, publisher or subscriber.</td>
</tr>
<tr>
<td>Quality of Service</td>
<td>string</td>
<td>Quality of service for the endpoint. For example RVCM or JMS Persistent.</td>
</tr>
<tr>
<td>Subject</td>
<td>string</td>
<td>Subject defined for this endpoint.</td>
</tr>
<tr>
<td>Class</td>
<td>string</td>
<td>Class associated with the endpoint.</td>
</tr>
<tr>
<td>Number of Messages</td>
<td>integer</td>
<td>Number of messages processed for this endpoint.</td>
</tr>
</tbody>
</table>
getComponents()

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

<table>
<thead>
<tr>
<th>Input Parameters</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Component Name</td>
<td>string</td>
<td>Name of the component. If no value is enter, all components display.</td>
</tr>
<tr>
<td>Component Type</td>
<td>string</td>
<td>Any of Publisher, Subscriber, Timer, or IODescriptor. The default value is All.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Returns</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Instance ID</td>
<td>string</td>
<td>Name of this adapter instance as defined at design-time.</td>
</tr>
<tr>
<td>Adapter Name</td>
<td>string</td>
<td>Name of the adapter.</td>
</tr>
<tr>
<td>Component Name</td>
<td>string</td>
<td>Name of the component.</td>
</tr>
<tr>
<td>Component Type</td>
<td>string</td>
<td>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.</td>
</tr>
<tr>
<td>Session Name</td>
<td>string</td>
<td>Name of the session.</td>
</tr>
<tr>
<td>Description</td>
<td>string</td>
<td>Information about this component, for example, time interval, signal type, and validating the publisher or subscriber.</td>
</tr>
</tbody>
</table>
getConfig()

Retrieves generic configuration information. More specific configuration information is accessed through separate methods.

<table>
<thead>
<tr>
<th>Returns</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Instance ID</td>
<td>string</td>
<td>Configuration ID of this adapter.</td>
</tr>
<tr>
<td>Adapter Name</td>
<td>string</td>
<td>Name of the adapter.</td>
</tr>
<tr>
<td>Repository Connection</td>
<td>string</td>
<td>URL of the repository used for adapter instance.</td>
</tr>
<tr>
<td>Configuration URL</td>
<td>string</td>
<td>Location of the adapter project; either a filename or configuration URL.</td>
</tr>
<tr>
<td>Command</td>
<td>string</td>
<td>Command line arguments used to start the adapter.</td>
</tr>
</tbody>
</table>
getConfigProperties()

Returns all attributes and elements for the given repository object.

<table>
<thead>
<tr>
<th>Input Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property</td>
<td>string</td>
<td>Name of the property for which elements (tags) and attributes are desired. For example,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>agentone/startup.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If no value is given, all properties are returned.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Returns</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Element Name</td>
<td>string</td>
<td>Repository directory for the property.</td>
</tr>
<tr>
<td>Attribute Name</td>
<td>string</td>
<td>Name of the repository object attribute.</td>
</tr>
<tr>
<td>Attribute Value</td>
<td>string</td>
<td>Value of the repository object attribute.</td>
</tr>
<tr>
<td>Line</td>
<td>integer</td>
<td>Line number in which this property is defined in the project file.</td>
</tr>
</tbody>
</table>
getDocumentDelay()

Returns the document delay setting for a given publication service. If the publication service is not given, a list of publication services is returned with their corresponding polling interval.
**getHostInformation()**

Returns standard and extended application information set. It returns the following information.

<table>
<thead>
<tr>
<th>Returns</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>string</td>
<td>Name of the property.</td>
</tr>
<tr>
<td>Value</td>
<td>string</td>
<td>Value of the property.</td>
</tr>
</tbody>
</table>

getPollingInterval()

Returns the current polling interval setting.

<table>
<thead>
<tr>
<th>Returns</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PollingInterval</td>
<td>integer</td>
<td>Polling interval in milliseconds</td>
</tr>
</tbody>
</table>
getQueueStatistics()

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

<table>
<thead>
<tr>
<th>Returns</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>QueueID</td>
<td>string</td>
<td>A unique identification of a particular queue.</td>
</tr>
<tr>
<td>QueueType</td>
<td>string</td>
<td>A type or key that will match this queue to a thread or connection.</td>
</tr>
<tr>
<td>QueueCount</td>
<td>integer</td>
<td>Current number of elements in the queue.</td>
</tr>
<tr>
<td>MaxQueueSize</td>
<td>integer</td>
<td>Maximum number of elements in the queue.</td>
</tr>
</tbody>
</table>
getRvConfig()

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

<table>
<thead>
<tr>
<th>Input Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Session Name</td>
<td>string</td>
<td>Name of the TIBCO Rendezvous session for which configuration is required. If not given, information about all sessions is returned. The default is all.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Returns</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Instance ID</td>
<td>string</td>
<td>Configuration ID of this adapter.</td>
</tr>
<tr>
<td>Adapter Name</td>
<td>string</td>
<td>Name of the adapter.</td>
</tr>
<tr>
<td>Session Name</td>
<td>string</td>
<td>Name of the session.</td>
</tr>
<tr>
<td>Service</td>
<td>string</td>
<td>Service parameter for this session.</td>
</tr>
<tr>
<td>Daemon</td>
<td>string</td>
<td>Daemon parameter for this session.</td>
</tr>
<tr>
<td>Network</td>
<td>string</td>
<td>Network parameter for this session.</td>
</tr>
<tr>
<td>Synchronous?</td>
<td>boolean</td>
<td>Returns 1 if this is a synchronous session, 0 otherwise.</td>
</tr>
<tr>
<td>Session Type</td>
<td>string</td>
<td>Type of session; one of M_RV, M_RVCM, or M_RVCMQ.</td>
</tr>
<tr>
<td>Certified Name</td>
<td>string</td>
<td>Name of this certified session.</td>
</tr>
<tr>
<td>Ledger File</td>
<td>string</td>
<td>Ledger file for this certified messaging session. Returns the empty string for sessions that are not certified messaging sessions.</td>
</tr>
<tr>
<td>CM Timeout</td>
<td>string</td>
<td>Timeout for this certified messaging session. Returns the empty string for sessions that are not certified messaging sessions.</td>
</tr>
</tbody>
</table>
getStatus()  

Retrieves basic status information about the adapter.  
This information is fairly limited; for more detail, additional methods are provided (getConfig() on page 153 and getRvConfig() on page 159).

<table>
<thead>
<tr>
<th>Returns</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Instance ID</td>
<td>string</td>
<td>Configuration ID for this adapter instance.</td>
</tr>
<tr>
<td>Adapter Name</td>
<td>string</td>
<td>Name of the adapter.</td>
</tr>
<tr>
<td>Uptime</td>
<td>integer</td>
<td>Number of seconds since startup.</td>
</tr>
<tr>
<td>Messages Received</td>
<td>integer</td>
<td>Number of TIBCO Rendezvous messages received.</td>
</tr>
<tr>
<td>Messages Sent</td>
<td>integer</td>
<td>Number of TIBCO Rendezvous messages published.</td>
</tr>
<tr>
<td>New Errors</td>
<td>integer</td>
<td>Number of errors since the last call to this method.</td>
</tr>
<tr>
<td>Total Errors</td>
<td>integer</td>
<td>Total number of errors since startup.</td>
</tr>
<tr>
<td>Process ID</td>
<td>integer</td>
<td>Process ID of the application.</td>
</tr>
<tr>
<td>Host</td>
<td>string</td>
<td>Name of host machine on which this adapter is running.</td>
</tr>
</tbody>
</table>
getTraceSinks()

Returns information about sinks to which traces currently go.

<table>
<thead>
<tr>
<th>Input Parameters</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sink Name</td>
<td>string</td>
<td>Name of the sink for which you need information. If no name is specified, information about all sinks is returned. Default is all.</td>
</tr>
<tr>
<td>Role Name</td>
<td>string</td>
<td>Name of the role for which you need information for the specified sink or sinks. Default is all.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Returns</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Instance ID</td>
<td>string</td>
<td>Name of this adapter instance as a string.</td>
</tr>
<tr>
<td>Adapter Name</td>
<td>string</td>
<td>Name of the application for this sink.</td>
</tr>
<tr>
<td>Sink Name</td>
<td>string</td>
<td>Name of the sink.</td>
</tr>
<tr>
<td>Sink Type</td>
<td>string</td>
<td>Type of this sink. One of fileSink, rvSink, hawkSink, stderrSink.</td>
</tr>
<tr>
<td>Roles</td>
<td>string</td>
<td>Roles this sink supports, as a string. For example warning, error, debug.</td>
</tr>
</tbody>
</table>
getVersion()  

Retrieves version information for the current application. Two lines may be returned, one for the TIBCO Adapter SDK, one for the adapter.

<table>
<thead>
<tr>
<th>Returns</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Instance ID</td>
<td>Configuration ID as a string, for example SDK.</td>
</tr>
<tr>
<td>Adapter Name</td>
<td>Name of the adapter as a string, for example agentone.</td>
</tr>
<tr>
<td>Version</td>
<td>Version number as a string, for example 5.5.</td>
</tr>
</tbody>
</table>
_onUnsolicitedMsg()

Displays all alert messages sent from the adapter or an error if not successful.
**preRegisterListener()**

Preregister an anticipated subscription service. Some sending applications can anticipate requests for certified delivery even before the listening applications start running. In such situations, the publication service can preregister subscription services, so TIBCO Rendezvous software begins storing outbound messages in the publication service ledger. If the listening correspondent requires old messages, it receives the backlogged messages when it requests certified deliver.

<table>
<thead>
<tr>
<th><strong>Input Parameters</strong></th>
<th><strong>Type</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Session Name</td>
<td>string</td>
<td>Name of the session that anticipates the listener.</td>
</tr>
<tr>
<td>Publisher Name</td>
<td>string</td>
<td>Name of the component for which the listener should be preregistered.</td>
</tr>
<tr>
<td>Listener Session Name</td>
<td>string</td>
<td>Name of the subscription service to preregister.</td>
</tr>
</tbody>
</table>

Returns OK if the subscription service was preregistered successfully, false otherwise.
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, ensure that the certified messaging publisher adapter has established a certified delivery agreement with its subscriber agents.

<table>
<thead>
<tr>
<th>Input Parameters</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Session Name</td>
<td>string</td>
<td>Name of the TIBCO Rendezvous session for which ledger information is desired (default is all).</td>
</tr>
<tr>
<td>Subject</td>
<td>string</td>
<td>Name of the subject for which ledger information is desired.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Returns</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Session Name</td>
<td>string</td>
<td>Name of the TIBCO Rendezvous CM session to which this information applies.</td>
</tr>
<tr>
<td>Subject</td>
<td>string</td>
<td>Subject name for this session.</td>
</tr>
<tr>
<td>Last Sent Message</td>
<td>integer</td>
<td>Sequence number of the most recently sent message with this subject name.</td>
</tr>
<tr>
<td>Total Messages</td>
<td>string</td>
<td>Total number of pending messages with this subject name.</td>
</tr>
<tr>
<td>Total Size</td>
<td>integer</td>
<td>Total storage (in bytes) occupied by all pending messages with this subject name.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If the ledger contains ten messages with this subject name, then this field sums the storage space over all of them.</td>
</tr>
<tr>
<td>Listener Session Name</td>
<td>string</td>
<td>Within each listener submessage, the Listener Session Name field contains the name of the delivery-tracking listener session.</td>
</tr>
</tbody>
</table>
Within each listener submessage, the Last Confirmed field contains the sequence number of the last message for which this listener session confirmed delivery.

<table>
<thead>
<tr>
<th>Returns (Cont’d)</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Last Confirmed</td>
<td>string</td>
<td>Row number in ledger file.</td>
</tr>
<tr>
<td>Line</td>
<td>integer</td>
<td>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.</td>
</tr>
</tbody>
</table>
**setDocumentDelay()**

Set the document delay for a publication service.

<table>
<thead>
<tr>
<th>Input Parameters</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DocumentDelay</td>
<td>integer</td>
<td>Document delay time in milliseconds.</td>
</tr>
<tr>
<td>ServiceName</td>
<td>string</td>
<td>Name of publication service to set.</td>
</tr>
</tbody>
</table>
setPollingInterval()

Set the polling interval for the publication service.

<table>
<thead>
<tr>
<th>Input Parameters</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PollingInterval</td>
<td>integer</td>
<td>Polling interval in milliseconds.</td>
</tr>
<tr>
<td>ServiceName</td>
<td>string</td>
<td>Name of service where the polling interval is set.</td>
</tr>
</tbody>
</table>
**setTraceSinks()**

Adds a role or changes the file limit of a previously specified sink.

<table>
<thead>
<tr>
<th>Input Parameters</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sink Name</td>
<td>string</td>
<td>Name of the sink for which you want to add a role or change the file limit.</td>
</tr>
<tr>
<td>Role Name</td>
<td>string</td>
<td>Name of the role you want to add to this sink (warning, error, debug, or user defined). Default is all.</td>
</tr>
<tr>
<td>File Size</td>
<td>integer</td>
<td>Maximum file size for this sink. This parameter is ignored if the sink specified by sinkName is not a file sink.</td>
</tr>
</tbody>
</table>

Returns OK if successful or an error if not successful.
stopApplicationInstance()

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

Unregister a currently preregistered subscription service.

<table>
<thead>
<tr>
<th>Input Parameters</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Session Name</td>
<td>string</td>
<td>Name of the session that anticipates the subscription service.</td>
</tr>
<tr>
<td>Publisher Name</td>
<td>string</td>
<td>Name of the publication service to which the subscription service is preregistered.</td>
</tr>
<tr>
<td>Listener Session Name</td>
<td>string</td>
<td>Name of the subscription service to unregister.</td>
</tr>
</tbody>
</table>

This method returns true if the subscription service was unregistered successfully. Otherwise, it returns false.
Appendix A Trace Messages

This appendix explains the trace messages that are logged to a location specified at configuration time.

Topics

- Overview, page 174
- Trace Message Fields, page 176
- Status Messages, page 178
Overview

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

Each trace message can include the following fields:

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

The above fields are explained in the Trace Message Fields on page 176. The following diagram shows an example trace message and calls out the fields.

Example Trace Messages

The following trace messages were written during a session where TIBCO ActiveMatrix Adapter for Files received an object from TIBCO ActiveMatrix Adapter for R/3, then processed the object.

The first message indicates that TIBCO ActiveMatrix Adapter for Files has started. The timestamp indicates when the adapter started, and the role indicates that the trace message is informational, which means the activity is normal for the adapter. The category is identified, and the corresponding status code is displayed. The status code indicates that the adapter started successfully.

2003 Feb 22 20:14:51:718 GMT -8
FileAdapter.FileAdapterConfiguration Info [Configuration] AEFA-000058 TIBCO ActiveMatrix Adapter for Files successfully initialized
The next set of trace messages indicates the adapter received an object that was sent on the TIBCO Rendezvous subject, FROM.SAP. The #MU3oTJ/WWCV1MU96J0zzwA9kzzw# tracking identifier included in the trace message uniquely identifies the message. The adapter (TIBCO ActiveMatrix Adapter for R/3) from which the message originated provided the identifier.

2003 Feb 22 20:15:12:937 GMT -8
Message containing class /tibco/public/class/ae/Customer received on subject FROM.SAP tracking=#MU3oTJ/WWCV1MU96J0zzwA9kzzw#

2003 Feb 22 20:15:12:937 GMT -8
Message containing class /tibco/public/class/ae/Customer written to working file customers.txt in Working Directory F:\ca\integration\001\data_sets\files\wip
tracking=#MU3oTJ/WWCV1MU96J0zzwA9kzzw#

The final trace message indicates the object has been moved to the output directory, which completes the adapter’s interaction with the object. Because the trace message is the termination point, the tracking identifier is not displayed.

2003 Feb 22 20:15:42:812 GMT -8
File customers.txt is moved to the Output Directory F:\ca\integration\001\data_sets\files\solutions\output
## Trace Message Fields

Each trace message includes the following fields:

### Table 10  Tracing Fields

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adapter Identifier</td>
<td>Name of the adapter that wrote the trace message. This is a combination of the adapter acronym and adapter configuration name. For example, the application identifier, ADFILES.publisher1 identifies a TIBCO ActiveMatrix Adapter for Files service named publisher1.</td>
</tr>
<tr>
<td>Role</td>
<td>A role can be:</td>
</tr>
<tr>
<td></td>
<td>• Info. Indicates normal adapter operation. No action is necessary. A tracing message tagged with Info indicates that a significant processing step was reached and has been logged for tracking or auditing purposes. Only info messages preceding a tracking identifier are considered significant steps.</td>
</tr>
<tr>
<td></td>
<td>• Warn. An abnormal condition was found. Processing will continue, but special attention from an administrator is recommended.</td>
</tr>
<tr>
<td></td>
<td>• Error. An unrecoverable error occurred. Depending on the error severity, the adapter may continue with the next operation or may stop altogether.</td>
</tr>
<tr>
<td></td>
<td>• Debug. A developer-defined tracing message. In normal operating conditions, debug messages should not display.</td>
</tr>
</tbody>
</table>

When configuring the adapter you define what roles should or should not be logged. For example, you may decide not to log Info roles to increase performance.
### Table 10  Tracing Fields

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Category</td>
<td>One of the following:</td>
</tr>
<tr>
<td></td>
<td>• Adapter. The adapter is processing an event.</td>
</tr>
<tr>
<td></td>
<td>• Configuration. The adapter is reading configuration information.</td>
</tr>
<tr>
<td></td>
<td>• Palette. The adapter is interacting with the palette.</td>
</tr>
<tr>
<td></td>
<td>• Publisher Service. The publication service is reporting this trace message.</td>
</tr>
<tr>
<td></td>
<td>• Shutdown. The adapter is shutting down.</td>
</tr>
<tr>
<td></td>
<td>• Startup. The adapter is starting.</td>
</tr>
<tr>
<td></td>
<td>• Subscription Service. The subscription service is reporting this trace message.</td>
</tr>
<tr>
<td></td>
<td>• System. This category is not linked to a specific event process. The trace message may be related to a Windows service related messages, memory allocation, file system error, and so on.</td>
</tr>
<tr>
<td></td>
<td>• TibRvComm. The adapter is communicating with TIBCO Rendezvous.</td>
</tr>
<tr>
<td>Status Code</td>
<td>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 warning role, the status code documentation includes a resolution. See Status Messages on page 178 for details.</td>
</tr>
<tr>
<td>Tracking Identifier</td>
<td>A unique identifier that is &quot;stamped&quot; on each message by the originating adapter. The tracking identifier remains in effect from a message’s beginning to its completion as it is exchanged by TIBCO applications. If the adapter is the termination point of the message, the tracking identifier is not displayed in the trace message. You cannot modify the tracking identifier format or configure what information is displayed.</td>
</tr>
</tbody>
</table>
# Status Messages

<table>
<thead>
<tr>
<th>Status Code</th>
<th>Role</th>
<th>Category</th>
<th>Resolution</th>
</tr>
</thead>
<tbody>
<tr>
<td>AEFA-000001</td>
<td>Exit</td>
<td>warnRole</td>
<td>Exit Subscriber is not defined</td>
</tr>
<tr>
<td></td>
<td>Role</td>
<td>Configuration</td>
<td></td>
</tr>
<tr>
<td>AEFA-000002</td>
<td>Out of memory</td>
<td>errorRole</td>
<td>System Check the system to determine why it ran out of memory.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>System</td>
<td></td>
</tr>
<tr>
<td>AEFA-000003</td>
<td>Missing class</td>
<td>errorRole</td>
<td>Please review the documentation. If that does not help, contact TIBCO support.</td>
</tr>
<tr>
<td></td>
<td>definition</td>
<td>Configuration</td>
<td></td>
</tr>
<tr>
<td>AEFA-000004</td>
<td>Missing</td>
<td>errorRole</td>
<td>Please review the documentation. If that does not help, contact TIBCO support.</td>
</tr>
<tr>
<td></td>
<td>attribute</td>
<td>Configuration</td>
<td></td>
</tr>
<tr>
<td>AEFA-000005</td>
<td>Attribute</td>
<td>errorRole</td>
<td>Please review the documentation. If that does not help, contact TIBCO support.</td>
</tr>
<tr>
<td></td>
<td>in the</td>
<td>Configuration</td>
<td></td>
</tr>
<tr>
<td></td>
<td>configuration</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Attribute</td>
<td>errorRole</td>
<td>Please review the documentation. If that does not help, contact TIBCO support.</td>
</tr>
<tr>
<td></td>
<td>in the</td>
<td>Configuration</td>
<td></td>
</tr>
<tr>
<td></td>
<td>configuration</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>has the wrong</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>value</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AEFA-000007</td>
<td>Couldn’t open</td>
<td>errorRole</td>
<td>Check the file system to ensure that all of the files exist and can be opened.</td>
</tr>
<tr>
<td></td>
<td>file</td>
<td>System</td>
<td></td>
</tr>
<tr>
<td>AEFA-000008</td>
<td>Couldn’t close</td>
<td>errorRole</td>
<td>Check the file system to determine why the file could not be closed.</td>
</tr>
<tr>
<td></td>
<td>file</td>
<td>System</td>
<td></td>
</tr>
<tr>
<td>AEFA-000009</td>
<td>Couldn’t read</td>
<td>errorRole</td>
<td>Check the file and the file system to determine why the error occurred.</td>
</tr>
<tr>
<td></td>
<td>file</td>
<td>System</td>
<td></td>
</tr>
<tr>
<td>Status Code</td>
<td>Role</td>
<td>Category</td>
<td>Resolution</td>
</tr>
<tr>
<td>---------------</td>
<td>-----------------------</td>
<td>----------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>AEFA-000011</td>
<td>Pre Processing Script File directory <code>&lt;directory name&gt;</code> couldn’t be found</td>
<td>errorRole</td>
<td>Make sure that the directory exists. If it does, correct the configuration to point to the correct directory for the Pre Processing Script File.</td>
</tr>
<tr>
<td>AEFA-000012</td>
<td>Pre Processing Script File <code>&lt;filename&gt;</code> couldn’t be found</td>
<td>errorRole</td>
<td>Make sure the file exists. If it does, correct the configuration to point to the correct file and directory names for the Pre Processing Script.</td>
</tr>
<tr>
<td>AEFA-000013</td>
<td>Post-processing script file <code>&lt;filename&gt;</code> couldn’t be found</td>
<td>errorRole</td>
<td>Correct the configuration to point to the correct file and directory names for the Publication Service’s Post Processing Script.</td>
</tr>
<tr>
<td>AEFA-000014</td>
<td>No access to Input Directory <code>&lt;directory name&gt;</code></td>
<td>errorRole</td>
<td>Create the Input Directory or allow access to it.</td>
</tr>
<tr>
<td>AEFA-000015</td>
<td>No access to Working Directory <code>&lt;directory name&gt;</code></td>
<td>errorRole</td>
<td>Create the Working Directory or allow access to it.</td>
</tr>
<tr>
<td>AEFA-000016</td>
<td>No access to Completion Directory <code>&lt;directory name&gt;</code></td>
<td>errorRole</td>
<td>Create the Completion Directory or allow access to it.</td>
</tr>
<tr>
<td>AEFA-000017</td>
<td>Working file <code>&lt;filename&gt;</code> already exists</td>
<td>errorRole</td>
<td>Remove the duplicate file from the Working Directory.</td>
</tr>
<tr>
<td>Status Code</td>
<td>Role</td>
<td>Category</td>
<td>Resolution</td>
</tr>
<tr>
<td>--------------</td>
<td>---------</td>
<td>----------</td>
<td>------------</td>
</tr>
<tr>
<td>AEFA-000018</td>
<td>errorRole</td>
<td>System</td>
<td>1. If none of the files are getting processed then 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.</td>
</tr>
<tr>
<td>AEFA-000019</td>
<td>warnRole</td>
<td>System</td>
<td>Check the file system to determine why the error occurred.</td>
</tr>
<tr>
<td>AEFA-000020</td>
<td>warnRole</td>
<td>System</td>
<td>Check the file system to determine why the error occurred.</td>
</tr>
<tr>
<td>AEFA-000021</td>
<td>warnRole</td>
<td>System</td>
<td>Check the file system to determine why the error occurred.</td>
</tr>
<tr>
<td>AEFA-000022</td>
<td>errorRole</td>
<td>System</td>
<td>Check the Trigger File rights and the file system state to determine why the error occurred.</td>
</tr>
<tr>
<td>AEFA-000023</td>
<td>errorRole</td>
<td>System</td>
<td>Check the working data file rights and the file system state to determine why the error occurred.</td>
</tr>
<tr>
<td>Status Code</td>
<td>Role</td>
<td>Category</td>
<td>Resolution</td>
</tr>
<tr>
<td>---------------</td>
<td>----------</td>
<td>--------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>AEFA-000024</td>
<td>errorRole</td>
<td>Adapter</td>
<td>Check the existence of the input data file and the input data file rights.</td>
</tr>
<tr>
<td>AEFA-000025</td>
<td>errorRole</td>
<td>Adapter</td>
<td>Check the existence of the input data file, the input data file rights, and the content of the CheckPoint Restart file.</td>
</tr>
<tr>
<td>AEFA-000026</td>
<td>errorRole</td>
<td>Adapter</td>
<td>Check the User exit server to determine why the error occurred.</td>
</tr>
<tr>
<td>AEFA-000027</td>
<td>errorRole</td>
<td>Adapter</td>
<td>Check that the User exit server is running.</td>
</tr>
<tr>
<td>AEFA-000028</td>
<td>errorRole</td>
<td>Adapter</td>
<td>Check the User exit server to determine why the error occurred.</td>
</tr>
<tr>
<td>AEFA-000029</td>
<td>errorRole</td>
<td>Adapter</td>
<td>Check that the User exit server is running.</td>
</tr>
<tr>
<td>AEFA-000030</td>
<td>errorRole</td>
<td>Configuration</td>
<td>Check if the publisher is sending an MInstance.</td>
</tr>
<tr>
<td>AEFA-000033</td>
<td>errorRole</td>
<td>Metadata</td>
<td>Configure the class in your Subscription Service or change its subject address.</td>
</tr>
<tr>
<td>AEFA-000034</td>
<td>errorRole</td>
<td>System</td>
<td>Check the file system to determine why the error occurred.</td>
</tr>
<tr>
<td>Status Code</td>
<td>Role</td>
<td>Category</td>
<td>Resolution</td>
</tr>
<tr>
<td>-------------</td>
<td>-------------</td>
<td>-------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>AEFA-000035</td>
<td>errorRole</td>
<td>Configuration</td>
<td>Create the Working Directory or allow access to it.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>AEFA-000036</strong> No access to Completion Directory &lt;directory name&gt;</td>
</tr>
<tr>
<td></td>
<td>errorRole</td>
<td>Configuration</td>
<td>Create the Completion Directory or allow access to it.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>AEFA-000037</strong> No access to Error Directory &lt;directory name&gt;</td>
</tr>
<tr>
<td></td>
<td>errorRole</td>
<td>Configuration</td>
<td>Create the Error Directory or allow access to it.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>AEFA-000038</strong> Cannot write file &lt;filename&gt; in Working Directory &lt;directory name&gt;</td>
</tr>
<tr>
<td></td>
<td>errorRole</td>
<td>System</td>
<td>Check the file system state to determine why the error occurred.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>AEFA-000039</strong> Cannot create file &lt;filename&gt; in Working Directory &lt;directory name&gt;</td>
</tr>
<tr>
<td></td>
<td>errorRole</td>
<td>System</td>
<td>Check the file system state to determine why the error occurred.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>AEFA-000040</strong> Cannot move file &lt;filename&gt; to Output Directory &lt;directory name&gt;</td>
</tr>
<tr>
<td></td>
<td>errorRole</td>
<td>System</td>
<td>Check the file system state to determine why the error occurred.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>AEFA-000041</strong> Cannot move file &lt;filename&gt; to Error Directory &lt;directory name&gt;</td>
</tr>
<tr>
<td></td>
<td>errorRole</td>
<td>System</td>
<td>Check the file system state to determine why the error occurred.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>AEFA-000042</strong> Cannot add TimeStamp to file &lt;filename&gt; in directory &lt;directory name&gt;</td>
</tr>
<tr>
<td></td>
<td>errorRole</td>
<td>System</td>
<td>Check the file system state to determine why the error occurred.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>AEFA-000043</strong> Post Processing Script file directory &lt;directory name&gt; couldn’t be found</td>
</tr>
<tr>
<td></td>
<td>errorRole</td>
<td>Configuration</td>
<td>Check that the directory for the Subscription Service’s Post Processing Script File exists and is readable.</td>
</tr>
<tr>
<td>Status Code</td>
<td>Role</td>
<td>Category</td>
<td>Resolution</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------------</td>
<td>-----------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>AEFA-000044</td>
<td>Post Processing Script file <code>&lt;filename&gt;</code> couldn’t be found</td>
<td>errorRole</td>
<td>Configuration Check that the Subscription Service’s Post Processing Script File exists and is readable.</td>
</tr>
<tr>
<td></td>
<td><code>&lt;command string&gt;</code> returned error code <code>&lt;status string&gt;</code></td>
<td>warnRole</td>
<td>System Check the Pre Processing script.</td>
</tr>
<tr>
<td></td>
<td>Post Processing script <code>&lt;command string&gt;</code> returned error code <code>&lt;status string&gt;</code></td>
<td>warnRole</td>
<td>System Check the Publication Service’s Post Processing Script.</td>
</tr>
<tr>
<td></td>
<td><code>&lt;class name&gt;</code> will not be published since no Publication Service could be found</td>
<td>warnRole</td>
<td>Configuration Assign a Read Schema to the File Record.</td>
</tr>
<tr>
<td></td>
<td><code>&lt;input record&gt;</code> couldn’t be parsed</td>
<td>warnRole</td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td>File <code>&lt;filename&gt;</code> has been parsed, <code>&lt;number of error records&gt;</code> lines couldn’t be interpreted</td>
<td>warnRole</td>
<td>Adapter Check if the input line is valid. If it is, then modify the File Record to correctly interpret it.</td>
</tr>
<tr>
<td></td>
<td>Subscription Service <code>&lt;service name&gt;</code> received an unexpected message. Type <code>&lt;event type&gt;</code>. Data <code>&lt;event data&gt;</code>.</td>
<td>warnRole</td>
<td>TibRvComm Check the format of data sent by other applications on the subject that this Subscription Service is listening on.</td>
</tr>
<tr>
<td></td>
<td>Attribute <code>&lt;attribute name&gt;</code> of class <code>&lt;class name&gt;</code> is of an unsupported type</td>
<td>warnRole</td>
<td>Metadata Check for any discrepancy between the incoming class and the Write Schema.</td>
</tr>
<tr>
<td>Status Code</td>
<td>Role</td>
<td>Category</td>
<td>Resolution</td>
</tr>
<tr>
<td>-------------</td>
<td>-----------</td>
<td>--------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>AEFA-000053</td>
<td>warnRole</td>
<td>Metadata</td>
<td>Change the input class to make sure that it only includes supported types.</td>
</tr>
<tr>
<td>AEFA-000054</td>
<td>warnRole</td>
<td>Adapter</td>
<td>Remove the file from the Input Directory. Determine why the file was still there so the problem will not repeat.</td>
</tr>
<tr>
<td>AEFA-000055</td>
<td>warnRole</td>
<td>Adapter</td>
<td>If the Subscription Service was not supposed to have timed out then you will need to check your publisher to find out why the publications were delayed.</td>
</tr>
<tr>
<td>AEFA-000056</td>
<td>warnRole</td>
<td>System</td>
<td>Check the Post Processing Script File to make sure that it is valid and correctly returned an error.</td>
</tr>
<tr>
<td>AEFA-000057</td>
<td>warnRole</td>
<td>Adapter</td>
<td>Check the content of your input and output data, check your logs for unsent data, and check if only one publisher is sending data.</td>
</tr>
<tr>
<td>AEFA-000058</td>
<td>infoRole</td>
<td>Configuration</td>
<td>Indicates normal adapter information. No action necessary.</td>
</tr>
<tr>
<td>AEFA-000059</td>
<td>infoRole</td>
<td>Adapter</td>
<td>Indicates normal adapter information. No action necessary.</td>
</tr>
<tr>
<td>Status Code</td>
<td>Role</td>
<td>Category</td>
<td>Resolution</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------------------</td>
<td>-----------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>AEFA-000060</td>
<td>Processing input file</td>
<td>Adapter</td>
<td>Indicates normal adapter information. No action necessary.</td>
</tr>
<tr>
<td></td>
<td>&lt;filename&gt; in Input</td>
<td></td>
<td>Directory &lt;directory name&gt;</td>
</tr>
<tr>
<td>AEFA-000061</td>
<td>Pre Processing script</td>
<td>Adapter</td>
<td>Indicates normal adapter information. No action necessary.</td>
</tr>
<tr>
<td></td>
<td>&lt;command string&gt;</td>
<td></td>
<td>succeeded</td>
</tr>
<tr>
<td>AEFA-000062</td>
<td>Execute Post Processing script</td>
<td>Adapter</td>
<td>Indicates normal adapter information. No action necessary.</td>
</tr>
<tr>
<td></td>
<td>&lt;command string&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AEFA-000063</td>
<td>Publication of file</td>
<td>Adapter</td>
<td>Indicates normal adapter information. No action necessary.</td>
</tr>
<tr>
<td></td>
<td>&lt;filename&gt; is finished.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AEFA-000064</td>
<td>File &lt;filename&gt; has</td>
<td>Adapter</td>
<td>Indicates normal adapter information. No action necessary.</td>
</tr>
<tr>
<td></td>
<td>been processed, all</td>
<td></td>
<td>lines were interpreted</td>
</tr>
<tr>
<td></td>
<td>lines were interpreted</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AEFA-000065</td>
<td>Message containing</td>
<td>Adapter</td>
<td>Indicates normal adapter information. No action necessary.</td>
</tr>
<tr>
<td></td>
<td>class &lt;class name&gt;</td>
<td></td>
<td>published on subject &lt;subject name&gt; (message is from file &lt;filename&gt;)</td>
</tr>
<tr>
<td></td>
<td>published on subject</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>&lt;subject name&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(message is from</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>file &lt;filename&gt;)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AEFA-000067</td>
<td>Message containing</td>
<td>Adapter</td>
<td>Indicates normal adapter information. No action necessary.</td>
</tr>
<tr>
<td></td>
<td>class &lt;class name&gt;</td>
<td></td>
<td>received on subject &lt;subject name&gt;</td>
</tr>
<tr>
<td></td>
<td>received on subject</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>&lt;subject name&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AEFA-000068</td>
<td>Message containing</td>
<td>Adapter</td>
<td>Indicates normal adapter information. No action necessary.</td>
</tr>
<tr>
<td></td>
<td>class &lt;class name&gt;</td>
<td></td>
<td>written to working file &lt;filename&gt; in Working Directory &lt;directory name&gt;</td>
</tr>
<tr>
<td></td>
<td>written to working</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>file &lt;filename&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>in Working Directory</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>&lt;directory name&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Status Code</td>
<td>Role</td>
<td>Category</td>
<td>Resolution</td>
</tr>
<tr>
<td>--------------</td>
<td>---------------</td>
<td>----------</td>
<td>---------------------------------------------------------------------------</td>
</tr>
<tr>
<td>AEFA-000069</td>
<td>&lt;number of messages&gt; messages have been published and received from file &lt;filename&gt;</td>
<td>infoRole</td>
<td>Adapter Indicates normal adapter information. No action necessary.</td>
</tr>
<tr>
<td>AEFA-000070</td>
<td>File &lt;filename&gt; is moved to the Output Directory &lt;directory name&gt;</td>
<td>infoRole</td>
<td>Adapter Indicates normal adapter information. No action necessary.</td>
</tr>
<tr>
<td>AEFA-000071</td>
<td>Post Processing Script &lt;command string&gt; succeeded</td>
<td>infoRole</td>
<td>Adapter Indicates normal adapter information. No action necessary.</td>
</tr>
<tr>
<td>AEFA-000072</td>
<td>Multiple publisher mode used</td>
<td>warnRole</td>
<td>Configuration Please contact TIBCO support if you experience any problem due to migration.</td>
</tr>
<tr>
<td>AEFA-000073</td>
<td>Subscription service &lt;service name&gt; received an empty MBusinessDocument in which the DataSection attribute was not set.</td>
<td>infoRole</td>
<td>Adapter Indicates normal adapter information. No action necessary.</td>
</tr>
<tr>
<td>AEFA-000074</td>
<td>The line &quot;&lt;input record&gt;&quot; contains an invalid field</td>
<td>errorRole</td>
<td>Adapter Correct the invalid field</td>
</tr>
<tr>
<td>AEFA-000075</td>
<td>Could not create simple datetime class. Received error code &lt;error code&gt;</td>
<td>errorRole</td>
<td>Adapter Unsupported locale is the main cause for this error</td>
</tr>
<tr>
<td>AEFA-000076</td>
<td>Failed to parse the datetime string &lt;datetime string&gt; for the pattern specified &lt;datetime pattern&gt;. Received error code &lt;error code&gt;</td>
<td>errorRole</td>
<td>Adapter Invalid pattern or locale mismatch is the main cause for this error</td>
</tr>
<tr>
<td>Status Code</td>
<td>Role</td>
<td>Category</td>
<td>Resolution</td>
</tr>
<tr>
<td>-------------</td>
<td>--------</td>
<td>----------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>AEFA-000077</td>
<td>Failed to format the datetime value to the specified pattern &lt;datetime pattern&gt;. Received error code &lt;error code&gt;</td>
<td>errorRole Adapter</td>
<td>Invalid pattern or locale mismatch is the main cause for this error</td>
</tr>
<tr>
<td>AEFA-000078</td>
<td>Failed to set attribute &lt;attribute name&gt; for the class &lt;class name&gt; Received error code &lt;error description&gt;</td>
<td>warnRole Adapter</td>
<td>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.</td>
</tr>
<tr>
<td>AEFA-000079</td>
<td>File &lt;filename&gt; has been parsed. Total: &lt;number of total records&gt; lines, Error: &lt;number of error records&gt; lines</td>
<td>infoRole Adapter</td>
<td>Indicates normal adapter information. No action necessary.</td>
</tr>
<tr>
<td>AEFA-000080</td>
<td>Processing file &lt;filename&gt;...</td>
<td>infoRole Adapter</td>
<td>Indicates normal adapter information. No action necessary.</td>
</tr>
<tr>
<td>AEFA-000081</td>
<td>Pre Processing Script &lt;command string&gt; returned message &lt;status string&gt;</td>
<td>infoRole Adapter</td>
<td>Indicates normal adapter information. No action necessary.</td>
</tr>
<tr>
<td>AEFA-000082</td>
<td>Post Processing Script &lt;command string&gt; returned message &lt;status string&gt;</td>
<td>infoRole Adapter</td>
<td>Indicates normal adapter information. No action necessary.</td>
</tr>
<tr>
<td>AEFA-000083</td>
<td>Post Processing Script &lt;command string&gt; returned message &lt;status string&gt;</td>
<td>infoRole Adapter</td>
<td>Indicates normal adapter information. No action necessary.</td>
</tr>
<tr>
<td>AEFA-000084</td>
<td>Skip processing file &lt;filename&gt; at the request of the preprocessing script.</td>
<td>infoRole Adapter</td>
<td>Indicates normal adapter information. No action necessary.</td>
</tr>
<tr>
<td>Status Code</td>
<td>Role</td>
<td>Category</td>
<td>Resolution</td>
</tr>
<tr>
<td>---------------</td>
<td>-------------</td>
<td>----------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>AEFA-000094</td>
<td>warnRole</td>
<td>Adapter</td>
<td>ECM is flex, and times to retry is over. The following subscribers have not responded and will be de-activated.</td>
</tr>
<tr>
<td>AEFA-000095</td>
<td>warnRole</td>
<td>Adapter</td>
<td>Registered Subscriber name = &lt;name string&gt;</td>
</tr>
<tr>
<td>AEFA-000099</td>
<td>warnRole</td>
<td>Adapter</td>
<td>Expected sequence does not match incoming data</td>
</tr>
<tr>
<td>AEFA-000100</td>
<td>warnRole</td>
<td>Adapter</td>
<td>Checksum mismatch with the published file</td>
</tr>
<tr>
<td>AEFA-000102</td>
<td>warnRole</td>
<td>Adapter</td>
<td>IO error detected</td>
</tr>
<tr>
<td>AEFA-000105</td>
<td>warnRole</td>
<td>Adapter</td>
<td>Since there are no active subscribers, publisher is switching to non-ECM</td>
</tr>
<tr>
<td>AEFA-000106</td>
<td>warnRole</td>
<td>Adapter</td>
<td>Publisher is switching to ECM</td>
</tr>
<tr>
<td>AEFA-000107</td>
<td>warnRole</td>
<td>Adapter</td>
<td>The admin confirmation timer is activated.</td>
</tr>
<tr>
<td>AEFA-000115</td>
<td>warnRole</td>
<td>Adapter</td>
<td>The admin confirmation timer is activated.</td>
</tr>
<tr>
<td>AEFA-000116</td>
<td>infoRole</td>
<td>Adapter</td>
<td>File &lt;filename&gt; is moved to the Error Directory &lt;directory name&gt; Indicating normal adapter information. No action necessary.</td>
</tr>
<tr>
<td>AEFA-000117</td>
<td>errorRole</td>
<td>Adapter</td>
<td>Cannot process messages due to version mismatch. Product version = &lt;version number&gt;. Received message version = &lt;version number&gt;</td>
</tr>
<tr>
<td>Status Code</td>
<td>Role</td>
<td>Category</td>
<td>Resolution</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------</td>
<td>----------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>AEFA-000118</td>
<td>No access to Error Directory &lt;directory name&gt;</td>
<td>errorRole</td>
<td>Configuration Indicates normal adapter information. No action necessary.</td>
</tr>
<tr>
<td>AEFA-000119</td>
<td>File transfer failed for the subscriber &lt;name string&gt; because &lt;error description&gt;. Updated subscriber specific error log file &lt;filename&gt;</td>
<td>errorRole</td>
<td>Configuration Indicates normal adapter information. No action necessary.</td>
</tr>
<tr>
<td>AEFA-000120</td>
<td>Unable to open the subscriber specific error log file &lt;filename&gt;</td>
<td>errorRole</td>
<td>Configuration Indicates normal adapter information. No action necessary.</td>
</tr>
<tr>
<td>AEFA-000125</td>
<td>Cannot reset input directory to &lt;directory name&gt;. The directory does not exist or cannot be accessed</td>
<td>errorRole</td>
<td>Configuration Indicates normal adapter information. No action necessary.</td>
</tr>
<tr>
<td>AEFA-000126</td>
<td>Input directory is successfully changed to &lt;directory name&gt;</td>
<td>infoRole</td>
<td>Indicates normal adapter information. No action necessary.</td>
</tr>
<tr>
<td>AEFA-000127</td>
<td>filename to process is reset to &lt;filename&gt;</td>
<td>infoRole</td>
<td>Indicates normal adapter information. No action necessary.</td>
</tr>
<tr>
<td>AEFA-000128</td>
<td>File prefix to process is reset to &lt;file prefix&gt;</td>
<td>infoRole</td>
<td>Indicates normal adapter information. No action necessary.</td>
</tr>
<tr>
<td>AEFA-000129</td>
<td>File extension to process is reset to &lt;file extension&gt;</td>
<td>infoRole</td>
<td>Indicates normal adapter information. No action necessary.</td>
</tr>
<tr>
<td>AEFA-000132</td>
<td>Subscriber is started Simple File Transfer mode</td>
<td>infoRole</td>
<td>Indicates normal adapter information. No action necessary.</td>
</tr>
<tr>
<td>Status Code</td>
<td>Role</td>
<td>Category</td>
<td>Resolution</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------------</td>
<td>-----------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>AEFA-000133</td>
<td>Publisher</td>
<td>Configuration</td>
<td>Indicates normal adapter information. No action necessary.</td>
</tr>
<tr>
<td>AEFA-000134</td>
<td>The progress file &lt;filename&gt; is corrupt</td>
<td>Configuration</td>
<td>Remove the progress file and restart the adapter.</td>
</tr>
<tr>
<td>AEFA-000135</td>
<td>Received advisory message. Role: &lt;role name&gt;, Subject: &lt;subject name&gt;, Message: &lt;message string&gt;</td>
<td>Adapter</td>
<td>Check the advisory message for cause.</td>
</tr>
<tr>
<td>AEFA-000136</td>
<td>Cannot add sequence number to file &lt;filename&gt; in directory &lt;directory name&gt;</td>
<td>System</td>
<td>Check the file system state to determine why the error occurred.</td>
</tr>
<tr>
<td>AEFA-000137</td>
<td>Polling subscriber endpoint is not defined</td>
<td>Adapter</td>
<td>The file palette generates the polling subscriber endpoint. Name of the endpoint is FAPollingSubscriberServiceName</td>
</tr>
<tr>
<td>AEFA-000138</td>
<td>Subscriber &lt;subscription service name&gt; 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</td>
<td>Adapter</td>
<td>If the IO error is temporary the subscriber will automatically come back up, if permanent then stop the subscriber and resolve the IO error and restart the subscriber</td>
</tr>
<tr>
<td>AEFA-000139</td>
<td>Progress filename is empty. Verify configuration and set a valid progress filename</td>
<td>Adapter</td>
<td>Set a valid progress filename using the designer and restart the publisher</td>
</tr>
<tr>
<td>Status Code</td>
<td>Role</td>
<td>Category</td>
<td>Resolution</td>
</tr>
<tr>
<td>---------------------</td>
<td>------------</td>
<td>----------</td>
<td>------------</td>
</tr>
<tr>
<td>AEFA-000140</td>
<td>IOError</td>
<td>Adapter</td>
<td>Set a valid progress filename using the designer and restart the publisher. Only LATIN_1 encoded filenames are valid.</td>
</tr>
<tr>
<td>AEFA-000143</td>
<td>Cannot open the data file &lt;filename&gt; specified in the progress file on publisher restart</td>
<td>Adapter</td>
<td>Check the existence of the data file, the data file rights, and the content of the progress file.</td>
</tr>
<tr>
<td>AEADFILES_910006</td>
<td>Exit subscriber is not defined. The Repository URL is &lt;repourl&gt; and the Configuration URL is &lt;configurl&gt;</td>
<td>Startup</td>
<td></td>
</tr>
<tr>
<td>AEADFILES_920001</td>
<td>Subscription error. &lt;service name, subject name, repourl, configurl&gt; received an invalid event.</td>
<td>Subscription</td>
<td>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 the TIBCO ActiveMatrix Adapter for Files User’s Guide for details on the configuration of the subscription service.</td>
</tr>
<tr>
<td>AEADFILES_920002</td>
<td>Subscription error. &lt;service name, subject name, repourl, configurl&gt; failed to deserialize the received event. SDK exception thrown is &lt;error description&gt;</td>
<td>Subscription</td>
<td>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 the TIBCO ActiveMatrix Adapter for Files User’s Guide for details on the configuration of the subscription service.</td>
</tr>
<tr>
<td>Status Code</td>
<td>Role</td>
<td>Category</td>
<td>Resolution</td>
</tr>
<tr>
<td>-------------------</td>
<td>-----------------------</td>
<td>------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>AEADFILES_920</td>
<td>Subscription error.</td>
<td>received inbound event</td>
<td>Subscription service &lt;service name, subject name, repourl, configurl&gt; received inbound event with null data. 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 the TIBCO ActiveMatrix Adapter for Files User’s Guide for details on the configuration of the subscription service.</td>
</tr>
<tr>
<td>003</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AEADFILES_920</td>
<td>Subscription service</td>
<td>received an MBusinessDocument in which the DataSection attribute was not set.</td>
<td></td>
</tr>
<tr>
<td>004</td>
<td></td>
<td></td>
<td>infoRole Subscription Indicates normal adapter information. No action necessary.</td>
</tr>
<tr>
<td>AEADFILES_920</td>
<td>Subscription error.</td>
<td>could not deserialize</td>
<td>Subscription error. &lt;error message&gt; could not deserialize the inbound event to MBusinessDocument &lt;name of the business document&gt; 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 the TIBCO ActiveMatrix Adapter for Files User’s Guide for details on the configuration of the subscription service.</td>
</tr>
<tr>
<td>005</td>
<td></td>
<td>the inbound event</td>
<td></td>
</tr>
<tr>
<td>AEADFILES_920</td>
<td>Subscription error.</td>
<td>could not get the class</td>
<td>Subscription error. Subscription service &lt;service name&gt; listening on subject &lt;subject name&gt; could not get the class description of &lt;class name&gt;. &lt;repoUrl and configUrl parameters&gt;. Check the repository configuration for this service. See the TIBCO ActiveMatrix Adapter for Files User’s Guide for details on how to configure, run and test the subscription service.</td>
</tr>
<tr>
<td>007</td>
<td></td>
<td>description of</td>
<td></td>
</tr>
</tbody>
</table>

TIBCO ActiveMatrix Adapter for Files for Unix/Win Configuration and Deployment
<table>
<thead>
<tr>
<th>Status Code</th>
<th>Role</th>
<th>Category</th>
<th>Resolution</th>
</tr>
</thead>
<tbody>
<tr>
<td>AEADFILES_920</td>
<td>Subscription error.</td>
<td></td>
<td>&lt;service name, subject name, repourl, configurl&gt; received event with invalid value &lt;attribute value&gt; for property &lt;attribute name&gt; in class &lt;class name&gt;.</td>
</tr>
<tr>
<td>009</td>
<td>AEADFILES_920</td>
<td>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 the TIBCO ActiveMatrix Adapter for Files User’s Guide for details on the configuration of the subscription service.</td>
<td></td>
</tr>
<tr>
<td>AEADFILES_920</td>
<td>Subscription error.</td>
<td></td>
<td>&lt;service name, subject name, repourl, configurl&gt; received event with missing attribute &lt;attribute name&gt; in class &lt;class name&gt;.</td>
</tr>
<tr>
<td>010</td>
<td>AEADFILES_920</td>
<td>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 the TIBCO ActiveMatrix Adapter for Files User’s Guide for details on the configuration of the subscription service.</td>
<td></td>
</tr>
<tr>
<td>AEADFILES_920</td>
<td>Subscription error.</td>
<td></td>
<td>&lt;service name, subject name, repourl, configurl&gt; received event with missing association &lt;association name&gt; for class &lt;class name&gt;.</td>
</tr>
<tr>
<td>011</td>
<td>AEADFILES_920</td>
<td>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 the TIBCO ActiveMatrix Adapter for Files User’s Guide for details on the configuration of the subscription service.</td>
<td></td>
</tr>
<tr>
<td>Status Code</td>
<td>Role</td>
<td>Category</td>
<td>Resolution</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------------</td>
<td>----------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>AEADFILES_920 012</td>
<td>Subscription error. Subscription service &lt;service name&gt;, subject &lt;subject name&gt;, repourl, configurl&gt; received MBusinessDocument &lt;business document name&gt; with NULL value for attribute &lt;attribute name&gt;.</td>
<td>errorRole Subscription</td>
<td>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 the TIBCO ActiveMatrix Adapter for Files User’s Guide for details on the configuration of the subscription service.</td>
</tr>
<tr>
<td>AEADFILES_920 015</td>
<td>Subscription error. Subscription service &lt;service name&gt; listening on subject &lt;subject name&gt; failed due to target application invocation error &lt;error description&gt;. Target application is FILES. The target application specific commands and parameters are &lt;command name&gt;</td>
<td>errorRole Subscription</td>
<td>Make sure the directory or file exists and the permission is set properly.</td>
</tr>
<tr>
<td>AEADFILES_920 020</td>
<td>Subscription error. &lt;service name&gt;, subject &lt;subject name&gt;, repourl, configurl&gt; received an event from the wire but encountered error &lt;error description&gt; in pre-processing user exit invocation. The User exit is &lt;userexit client name&gt;. The event details are &lt;class name&gt;.</td>
<td>errorRole Subscription</td>
<td>Make sure the parameters passed to the UserExit are valid and the User Exit can be invoked by the adapter.</td>
</tr>
<tr>
<td>AEADFILES_930 002</td>
<td>Publication error. Publication service &lt;service name&gt; with publication subject &lt;subject name&gt; encountered error &lt;error description&gt;.</td>
<td>errorRole Publication</td>
<td>Make sure the directory or file exists and the permission is set correctly.</td>
</tr>
<tr>
<td>Status Code</td>
<td>Role</td>
<td>Category</td>
<td>Resolution</td>
</tr>
<tr>
<td>----------------</td>
<td>--------------</td>
<td>----------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>AEADFILES_930</td>
<td>Publication</td>
<td></td>
<td>Verify the configuration of the publication service and check that the schema/class definitions are present in the repository. See the TIBCO ActiveMatrix Adapter for Files User’s Guide for details on how to configure a Publication service.</td>
</tr>
<tr>
<td>003</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Appendix A  Trace Messages

<table>
<thead>
<tr>
<th>Status Code</th>
<th>Role</th>
<th>Category</th>
<th>Resolution</th>
</tr>
</thead>
<tbody>
<tr>
<td>AEDFILES_930 006</td>
<td>Publication error. &lt;service name, subject name, repourl, configurl&gt;</td>
<td>It failed while converting event to MInstance. Attribute &lt;attribute name&gt; of class &lt;class name&gt; is missing.</td>
<td>Verify the configuration of the publication service and check that the schema definitions are present in the repository. See the TIBCO ActiveMatrix Adapter for Files User’s Guide for details on how to configure a Publication service.</td>
</tr>
<tr>
<td>AEDFILES_930 007</td>
<td>Publication error. &lt;service name, subject name, repourl, configurl&gt;</td>
<td>received event from target application but could not create the business document &lt;business document name&gt;. The target application details are &lt;class name&gt;.</td>
<td>Verify the configuration of the publication service and check that the schema definitions for the MBusinessDocument maps properly to the event received from the target application. See the TIBCO ActiveMatrix Adapter for Files User’s Guide for details on how to configure a Publication service.</td>
</tr>
<tr>
<td>AEDFILES_930 008</td>
<td>Publication error. &lt;service name, subject name, repourl, configurl&gt;</td>
<td>received SDK Exception &lt;error description&gt; while converting the event received from target application to BusinessDocument. The exception occurred while setting the attribute &lt;attribute name&gt;. The target application details are &lt;class name&gt;.</td>
<td>Verify the configuration of the publication service and check that the schema definitions for the MBusinessDocument maps properly to the event received from the target application. See the TIBCO ActiveMatrix Adapter for Files User’s Guide for details on how to configure a Publication service.</td>
</tr>
<tr>
<td>Status Code</td>
<td>Role</td>
<td>Category</td>
<td>Resolution</td>
</tr>
<tr>
<td>---------------------</td>
<td>---------------</td>
<td>----------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>AEADFILES_930</td>
<td>Publication</td>
<td>AEADFILES_930 009</td>
<td>Publication error. &lt;service name, subject name, repourl, configurl&gt; received an event from the target application but encountered error &lt;error description&gt; in pre-processing user exit invocation. The User exit is &lt;userexit client name&gt;. The target application details are &lt;class name&gt;. Make sure the parameters passed to the UserExit are valid and the User Exit can be invoked by the adapter.</td>
</tr>
<tr>
<td>AEADFILES_930</td>
<td>Publication</td>
<td>AEADFILES_930 014</td>
<td>Publication error. &lt;service name, subject name, repourl, configurl&gt; received error while sending event over the wire. The Publish endpoint details are &lt;error description&gt; Check the repository settings for a valid configuration of the publish endpoint for this service. See the TIBCO ActiveMatrix Adapter for Files User’s Guide for details on setting up a Publication service and a publish endpoint.</td>
</tr>
<tr>
<td>AEADFILES_930</td>
<td>Publication</td>
<td>AEADFILES_930 015</td>
<td>Publication error. &lt;service name, subject name, repourl, configurl&gt; Field count or record length does not match the schema. Correct the record according to the schema.</td>
</tr>
<tr>
<td>AEADFILES_930</td>
<td>Publication</td>
<td>AEADFILES_930 016</td>
<td>Publication error. &lt;service name, subject name, repourl, configurl&gt; Publication service cannot create RegEx matcher for the configured pattern. Check the configured pattern for validity.</td>
</tr>
<tr>
<td>AEADFILES_930</td>
<td>Publication</td>
<td>AEADFILES_930 017</td>
<td>Publication service notification. Indicates normal adapter information. No action necessary.</td>
</tr>
<tr>
<td>AEADFILES_980</td>
<td>System</td>
<td>AEADFILES_980 001</td>
<td>&lt;function name&gt; Running out of memory when trying to create a new object. Shutting down... Check the system memory usage.</td>
</tr>
<tr>
<td>AEADFILES_990</td>
<td>Shutdown</td>
<td>AEADFILES_990 002</td>
<td>Shutdown error. SDK cleanup exception = &lt;error description&gt; Check the system resource.</td>
</tr>
<tr>
<td>Status Code</td>
<td>Role</td>
<td>Category</td>
<td>Resolution</td>
</tr>
<tr>
<td>--------------</td>
<td>--------------</td>
<td>------------</td>
<td>---------------------------------</td>
</tr>
<tr>
<td>AEADFILES_990003</td>
<td>Shutdown error. Failed to cleanup Hawk microagent. SDK exception = &lt;error description&gt;</td>
<td>errorRole Shutdown</td>
<td>Check the system resource.</td>
</tr>
</tbody>
</table>
Appendix B  Adapter Properties File

This appendix introduces the adapter TRA file.

Topics

- Overview, page 200
- Properties File Format, page 201
- Predefined Properties, page 202
Overview

The runtime adapter parses a properties file at startup. The default runtime adapter properties file is named `adfilesagent.tra`.

The default properties file is located in the `bin` subdirectory of the adapter installation directory.

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-white space character and ends at the first `"=", ":",` or white space character.

Properties defined in the properties file override the same properties defined in the project.
The following restrictions apply to properties:

- The "!" character may not be used as a comment line indicator. Only the "#" character is recognized.
- The line continuation character is ignored (a value must fit on a line).
- The key may not contain any of the termination characters. The adapter does not support this syntax.
## Predefined Properties

Table 11 lists the predefined properties for TIBCO ActiveMatrix Adapter for Files.

All paths inside a properties file, including Microsoft Windows directory names, must use forward slashes.

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>adfiles.traceOldMessages</td>
<td>Either ON or OFF. The default value is OFF.</td>
</tr>
<tr>
<td></td>
<td>If ON, it enables 4.x style log messages and disables 5.x style log messages.</td>
</tr>
<tr>
<td>adfiles.EEMEnabled</td>
<td>Either ON or OFF. The default value is OFF.</td>
</tr>
<tr>
<td></td>
<td>It enables or disables Business Event Messages. See Business Event Messages on page 130 for more information on using this property.</td>
</tr>
<tr>
<td>adfiles.perfMon</td>
<td>Either ON or OFF. The default value is OFF.</td>
</tr>
<tr>
<td>adfiles.addCustomHawkMethodsToClassMAgent</td>
<td>Either ON or OFF. The default value is OFF.</td>
</tr>
<tr>
<td></td>
<td>It allows or disallows adding custom methods to the adapter’s standard microagent.</td>
</tr>
<tr>
<td>adfiles.SFTUseJMSMapMsg</td>
<td>Either ON or OFF. The default value is OFF.</td>
</tr>
<tr>
<td></td>
<td>If ON, SFT mode is enabled to exchange JMS map messages.</td>
</tr>
</tbody>
</table>

### The Following Properties Are for Publication Services Only.

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>adfiles.logSftProgressEveryNlocks</td>
<td>The default value is not to log progress status.</td>
</tr>
<tr>
<td></td>
<td>The value is integer. If you set the value to 10, the adapter will print a progress status every 10 messages.</td>
</tr>
<tr>
<td>Property</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>adfiles.sendLotEndWithDataMsg</td>
<td>Either ON or OFF. The default value is ON. If this value is ON and the Data Format is Business Document, this property instructs the publication service to send the LotEnd attribute either in an independent message or with the very last data message of a file. See <em>TIBCO ActiveMatrix Adapter for Files Concepts</em> for more information about the LotEnd attribute.</td>
</tr>
<tr>
<td>adfiles.schemaDiagnosticsFile</td>
<td>Either txt or xml. The default value is xml. When a publication service finds records that do not match the schemas, it writes them to a file. This property instructs the publication service to write the file either as plain text or in XML-like format.</td>
</tr>
<tr>
<td>adfiles.matchFieldCountAndRecordLength</td>
<td>Either ON or OFF. The default value is OFF. It enables or disables checking field count and record length when validating a record.</td>
</tr>
<tr>
<td>adfiles.quotedField</td>
<td>Either ON or OFF. The default value is OFF. If ON, the fields in double quotes will be treated and processed as one field.</td>
</tr>
<tr>
<td>adfiles.JMSCompress</td>
<td>Either ON or OFF. The default value is OFF. If this property is set to ON, the JMS messages will be compressed before being sent.</td>
</tr>
<tr>
<td>adfiles.DeltaFlushInterval</td>
<td>The default value is 3. In Delta Publishing mode, if there is no new data appended to an input file after a specified amount of polling, the data remaining in memory will be considered as complete data and published.</td>
</tr>
<tr>
<td>adfiles.DeltaTrimInterval</td>
<td>The default value is 10. The time interval in minutes to trim the WIP file in Delta Publishing mode.</td>
</tr>
</tbody>
</table>
Table 11  Predefined Properties for TIBCO ActiveMatrix Adapter for Files

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>adfiles.LBEnabled</td>
<td>Either ON or OFF. The default value is OFF.</td>
</tr>
<tr>
<td></td>
<td>If ON, Load Balancing mode is enabled to allow multiple instances of the same publication service to process separate files in parallel. See Load Balancing Mode for more information.</td>
</tr>
<tr>
<td>adfiles.NullNumberFieldAsZero</td>
<td>Either ON or OFF. The default value is OFF.</td>
</tr>
<tr>
<td></td>
<td>It enables or disables treating null fields of integer type as zero.</td>
</tr>
<tr>
<td>adfiles.PublishEmptyField</td>
<td>Either ON or OFF. The default value is OFF.</td>
</tr>
<tr>
<td></td>
<td>If ON, the empty fields will be published. If OFF, the empty fields will be skipped and will not be published.</td>
</tr>
<tr>
<td>adfiles.BDGroupingFactor</td>
<td>At runtime, this value will override the Grouping Factor value specified at design time. This property only applies to Business Document data format.</td>
</tr>
</tbody>
</table>

The Following Properties Are for Subscription Services Only.

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>adfiles.startSubscriberOnNewFileBoundary</td>
<td>Either ON or OFF. The default value is ON.</td>
</tr>
<tr>
<td></td>
<td>If ON, upon restart, a subscriber will discard partial files and start processing only from the beginning of a new file. This property only applies to the Simple File Transfer mode with non-ECM, JMS transport, and Durable Delivery Mode.</td>
</tr>
<tr>
<td>adfiles.dontAddEOLToFinalRecord</td>
<td>Either ON or OFF. The default value is OFF.</td>
</tr>
<tr>
<td></td>
<td>If ON, the EOL will not be added to the final record of a file. This property only applies to the Simple File Transfer mode with TEXT File Transfer Mode, and File Content Encoding being anything other than ASCII.</td>
</tr>
<tr>
<td>adfiles.toggleChildRecordsOrdering</td>
<td>Either ON or OFF. The default value is OFF.</td>
</tr>
<tr>
<td></td>
<td>If ON, the sequence of the child orders is reversed in the generated file. If OFF, the sequence of the child orders does not change.</td>
</tr>
<tr>
<td></td>
<td>This property will take effect only if no reordering of child records is done in TIBCO Designer.</td>
</tr>
</tbody>
</table>
Table 11  Predefined Properties for TIBCO ActiveMatrix Adapter for Files

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>adfiles.sub.discardDuplicateMessages</td>
<td>Either ON or OFF. The default value is OFF. If ON, this property prevents a subscription service from writing duplicate messages, which are caused by a publication service restart, to the output file. This property only applies to Business Document data format.</td>
</tr>
<tr>
<td>adfiles.useBDEmbeddedFileName</td>
<td>Either ON or OFF. The default value is OFF. If ON, this property instructs the subscription service to use 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 Designer. This property only applies to the Business Document data format. See <em>TIBCO ActiveMatrix Adapter for Files Concepts</em> for more information about the LotId attribute.</td>
</tr>
</tbody>
</table>
Index

A

ActiveEnterprise (AE) 33
adapter
   configuration tab 32
   configuring 32
   monitoring tab 36
   startup tab 35
adapter features 2
adapter project life cycle 12
adapter services folder 6
adapter, configuration 5
adapter, deployment 10
adapter, monitor 95
adapter, start 94
adapter, stop 94
add timestamp to file name, publication 46
adfiles.addCustomHawkMethodsToClassMAgent 20
   2
adfiles.BDGroupingFactor 204
adfiles.DeltaFlushInterval 203
adfiles.DeltaTrimInterval 203
adfiles.dontAddEOLToFinalRecord 204
adfiles.EEMEnabled 202
adfiles.JMSCompress 203
adfiles.LBEnabled 204
adfiles.logSftProgressEveryNBlocks 202
adfiles.matchFieldCountAndRecordLength 203
adfiles.NullNumberFieldAsZero 204
adfiles.perfMon 202
adfiles.PublishEmptyField 204
adfiles.quotedField 203
adfiles.schemaDiagnosticsFormat 203
adfiles.sendLotEndWithDataMsg 203
adfiles.SFTUseJMSMapMsg 202
adfiles.startSubscriberOnNewFileBoundary 204
adfiles.sub.discardDuplicateMessages 205
adfiles.toggleChildRecordsOrdering 204
adfiles.traceOldMessages 202
adfiles.useBDEmbeddedFileName 205
adfilesagent.tra 200
advanced folder 6
advanced tab, publication 56
advanced tab, subscription 72
AESchemas 6
agents 136
alerts 136
auto-discovery process 138

B

base record 111
business document name, publication 57
business event messages 130
byte order mark, subscription 70

C

child record, promoting and demoting 88
child record, promotion restrictions 88
Class Microagent Name field, adapter 36
close file after time elapsed, subscription 62
closing subscriber, subscription 63
command line arguments 153
completion directory, publication 45
CONFIG_HOME xiv
configuration properties, retrieving through TIBCO Hawk 153
configuration tab
   adapter 32
configuration tab, publication 38
configuration tab, subscription 60
configuration task, adapter 28
configuring an adapter 32
Connection Factory Type, JMS, publication 53
Connection Factory Type, JMS, subscription 67
Container Record 111
Content Encodings 128
Creation Mode, subscription 61
Custom Microagents 145
customer support xvi

d
Data Format, publication 57
Data Validation 110
date and time symbols 115
defining
  publication service 38
  subscription service 60
defining schema 3
Delimited Record
  Attributes 78
  Delimiter 76
  Delimiter Other 77
  Display Wizard 77
  Identifier Type 77
  Name 76
  Position from Header 76
  Repeating 76
  Strip Blanks 76
Delimited Record Type 76
Delivery Mode, JMS, publication 53
Delivery Mode, JMS, subscription 67
Delta Publishing, publication 39
Destination, JMS, publication 53
Destination, JMS, subscription 67
Diagnostics File 101
Document Delay, publication 57
Documentation xiii

e
EAR 92
ECM Subscriber Name, subscription 72

e
encoding tab, publication 54
Encoding Tab, subscription 68
End of Line, publication 56
End of Line, subscription 70
Endpoint Reference, publication 54
ENV_NAME xiv
error codes, list 178
Error Directory, publication 46
Error Directory, subscription 63
Error Handling 100
Explicit Confirmation Mode, subscription 71

f
File and Content Encoding 128
File and Directory Encodings 128
File Content Encoding, subscription 69, 70
File Encoding, publication 55, 56
File Extension, publication 43
File Name Selection, subscription 62
File Name, publication 42
File Name, subscription 62
File Prefix, publication 43
File Recognition 3
File Recognition Methods 108
File Records, modify the relationship 87
File Transfer Mode, subscription 72
flow control 3

G
Getting Started 15
global variables 104, 104, 104, 105
Global Variables, specify 104
Globalization 3
Grouping Factor, publication 57

Index | 209

I

ICU Regular Expression 4
Input Buffer Max Size, publication 58
Input Directory, publication 41
invoking microagent methods 139

J

JMS Selective Routing 3

L

layout file record 84
ledger files
    retrieving information through TIBCO Hawk 166
Life Cycle, publication 40
Locales Supported for Date and Time 117
Log File 35
Log File field, adapter 35
Log Info field, adapter 35
Log sink type
    File 34
    Hawk 34
    Network 34
    STDIO 34
Log to Standard field, adapter 35
log,Standard I/O 35
logging tab, adapter 34

M

Message Filter 33
Message Subject, publication 51
Message Subject, subscription 65
Message Time Out field, subscription 62
Message Time Out, subscription 62
metadata 7
Metadata Search URL 35

microagent 3
Microagent Methods 139
microagent methods supported 142
Microagent Session field, adapter 36
microagents
    invoking 139
microagents list 142
Monitoring tab, adapter 36

N

Name field
    adapter 32
Name, subscription 60

O

Operation Mode 28
Output Creation Mode, subscription 62
Output Directory, subscription 63
Output File Names 127
output Filename, Append, subscription 64
output Filename, Location, subscription 64
output Filename, Sequence Number Width, subscription 64
output Filename, subscription 64

P

Palettes 6
palettes 6
Polling Interval, publication 41
Polling Method, publication 40
Polling, Subject or Destination, publication 41
Positional Record Type 79
  Attributes 80
  Display Wizard 79
  Identifier Type 79
  Length 79
  Name 79
  Position from Header 79
  Strip Blanks 79
Post Processing Arguments, publication 46
Post Processing Arguments, subscription 64
Post Processing Script File, publication 46
Post Processing Script, subscription 63
Post Processing, publication 46
Pre Processing Arguments, publication 44
Pre Processing Script File, publication 43
Predefined Global Variables 105
Preserve Undelivered, publication 39
processing tab, publication 43
Processing Tab, subscription 63
Progress Directory, publication 46
Progress Directory, subscription 63
Progress File Name, subscription 72
  project 5
    Enterprise Archive File 7
    Local Repository 7
    Server Repository 7
ZIP Archive 7
project, deploy 93

properties
  adfiles.BDGroupingFactor 204
  adfiles.DeltaFlushInterval 203
  adfiles.DeltaTrimInterval 203
  adfiles.dontAddEOLToFinalRecord 204
  adfiles.EEMEnabled 202
  adfiles.JMSCompress 203
  adfiles.LBEenabled 204
  adfiles.logSftProgressEveryNBlocks 202
  adfiles.matchFieldCountAndRecordLength 203
  adfiles.NullNumberFieldAsZero 204
  adfiles.PublishEmptyField 204
  adfiles.quotedField 203
  adfiles.schemaDiagnosticsFileFormat 203
  adfiles.sendLotEndWithDataMsg 203
  adfiles.SFTUseJMSMapMsg 202
  adfiles.startSubscriberOnNewFileBoundary 204
  adfiles.sub.discardDuplicateMessages 205
  adfiles.toggleChildRecordsOrdering 204
  adfiles.traceOldMessages 202
  adfiles.useBDEmbeddedFileName 205

publication service
  advanced tab 56
  defining 38

Quality of Service, Certified 51
Quality of Service, Distributed Queue, subscription 66
Quality of Service, publication 51
Quality of Service, Reliable 52
Quality of Service, Reliable, subscription 65
Quality of Service, subscription 65

Read Schema 74
Recognition Method, publication 41
Record Transfer Mode 100
Record, edit 87
Index | 211

Reply Message Subject, publication 51
repository instance 7
reviewLedger, TIBCO Hawk method 166
revision control systems 98

TIBCO Administrator
   Application Management 10
   Resource Management 10
   User Management 10
TIBCO Administrator GUI 11
TIBCO Administrator Server 11
TIBCO Designer 5
   Configuration Panel 7
   Design Panel 6
   Palette Panel 6
   Project Panel 5
TIBCO Hawk 136
   background information 136
   enterprise monitor components 136
TIBCO Hawk methods
   _onUnsolicitedMsg() 163
   activateTraceRole() 146
   deactivateTraceRole() 147
   getActivityStatisticsByOperation(Operation) 148
   getActivityStatisticsBySchema(SchemaName) 149
   getActivityStatisticsByService(ServiceName) 150
   getAdapterServicesInformation() 151
   getConfig() 153
   getConfigProperties() 154
   getDocumentDelay() 155
   getHostInformation() 156
   getPollingInterval() 157
   getQueueStatistics() 158
   getRvConfig() 159
   getStatus() 160
   getTraceSinks() 161
   getVersion() 162
   preRegisterListener() 164
   resetActivityStatistics() 165
   reviewLedger() 166
   setDocumentDelay() 168
   setPollingInterval() 169
   setTraceSinks() 170
   stopApplicationInstance() 171
   unregisterListener() 172
TIBCO Rendezvous, retrieving configuration through
   TIBCO Hawk 159
TIBCO_HOME xiv
TRA file 199
Trace Message Fields 176

S

schema data 7
Schema tab
   subscription service 68
Schema Tab, subscription 65
Selective Routing 125
   Syntax 125
Session Reference, publication 54
Session Reference, subscription 68
SFT Tab, subscription 70
Simple File Transfer 47, 47, 102
Simple File Transfer, status message 131
Simple Transfer Mode 3
sinks, creating 34
Standard and Class Microagents 142
Standard Microagent 36
Standard Microagent Name field, adapter 36
Startup Banner 35
startup tab
   adapter 35
Subscriber Endpoint Reference, subscription 68
subscription service
   schema tab 68
subscription service, defining 60
substitution 104
support, contacting xvi
Synchronous Mode, publication 58
System Encoding, publication 54, 55
System Encoding, subscription 68, 69

T

technical support xvi
Termination Subject, Topic 34
TIBCO administration domain 10
Trace Messages 173
Trace Messages, example 174
trace messages, list 178
trace messages, Status Messages 178
Tracing Levels and Fields 176, 176
Transfer Mode, publication 39
Transfer Mode, subscription 61
Transport Tab, subscription 65
Transport Type, publication 38
Transport Type, subscription 60
Trigger File Extension, publication 43
Trigger Messages 122
    Syntax 122
    Usage Scenarios 123

U

Use Advanced Logging field, adapter 34
User Defined EOL, subscription 70
User Exit Endpoint Reference, publication 57
User Exit Endpoint Reference, subscription 73

V

variable substitution 104
variables 104
Version Control 8

W

Wire Format, publication 52
Wire Format, subscription 66
Working Directory, publication 45
Working Directory, subscription 63
Write Schemas 81