

# **TIBCO ActiveMatrix<sup>®</sup> Adapter for Tuxedo**

## **Configuration and Deployment**

*Software Release 6.0  
March 2010*

## 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 LICENSE.PDF) 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.

TIB, TIBCO, TIBCO Adapter, Predictive Business, Information Bus, The Power of Now, TIBCO ActiveMatrix BusinessWorks, TIBCO Rendezvous, TIBCO Administrator, TIBCO Designer, TIBCO Runtime Agent, TIBCO Hawk, TIBCO Enterprise Message Service, TIBCO Designer Add-in for TIBCO Business Studio, TIBCO ActiveMatrix Service Grid, TIBCO ActiveMatrix Service Bus, TIBCO ActiveMatrix BusinessWorks Service Engine, TIBCO ActiveMatrix Administrator, and TIBCO Business Studio are either registered trademarks or trademarks of TIBCO Software Inc. in the United States and/or other countries.

EJB, Java EE, J2EE, and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. 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.TXT 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 © 1998-2010 TIBCO Software Inc. ALL RIGHTS RESERVED.

TIBCO Software Inc. Confidential Information

# Contents

<b>Figures</b> .....	<b>vii</b>
<b>Tables</b> .....	<b>ix</b>
<b>Preface</b> .....	<b>xi</b>
Related Documentation .....	xii
TIBCO ActiveMatrix Adapter for Tuxedo Documentation .....	xii
Other TIBCO Product Documentation .....	xii
Third-Party Documentation .....	xiii
Typographical Conventions .....	xiv
How to Contact TIBCO Support .....	xvii
 <b>Chapter 1 Preparing Tuxedo</b> .....	 <b>1</b>
Preparing Tuxedo .....	2
 <b>Chapter 2 Getting Started</b> .....	 <b>11</b>
Overview .....	12
Before Starting .....	12
Preparing Tuxedo .....	13
Configuring the Adapter Components .....	14
Deploying the Adapter .....	18
 <b>Chapter 3 Adapter Instance Options</b> .....	 <b>21</b>
Overview .....	22
Configuration Tasks .....	22
Multithreading .....	23
Connection Management .....	23
Internationalization .....	24
Saving the Project .....	24
Testing the Adapter .....	25
Adapter Instance Tabs .....	26
Configuration Tab .....	26
Run-time Connection Tab .....	28
Adapter Services Tab .....	30

General Tab . . . . .	31
MultiThreading Tab . . . . .	32
Logging Tab . . . . .	33
Startup Tab . . . . .	36
Monitoring Tab . . . . .	36
<b>Chapter 4 Adapter Service Options . . . . .</b>	<b>39</b>
Overview . . . . .	40
Message Transports . . . . .	40
Publication Service Tabs . . . . .	44
Configuration Tab . . . . .	44
Publisher Options Tab . . . . .	45
Fetch Schema Tab . . . . .	46
Publisher Schema View Tab . . . . .	47
Schema Tab . . . . .	48
Transport Tab . . . . .	48
Subscription Service Tabs . . . . .	51
Configuration Tab . . . . .	51
Subscriber Options Tab . . . . .	53
Fetch Schema Tab . . . . .	54
Subscriber Schema View Tab . . . . .	54
Reply Schema View Tab . . . . .	55
Schema Tab . . . . .	55
Transport Tab . . . . .	55
Request-Response Service Tabs . . . . .	58
Configuration Tab . . . . .	58
RPC Server Options Tab . . . . .	60
Fetch Schema Tab . . . . .	61
Request Schema View Tab . . . . .	61
Reply Schema View Tab . . . . .	62
Schema Tab . . . . .	62
Transport Tab . . . . .	62
Request-Response Invocation Service Tabs . . . . .	65
Configuration Tab . . . . .	65
RPC Client Options Tab . . . . .	67
Fetch Schema Tab . . . . .	68
Request Schema View Tab . . . . .	69
Reply Schema View Tab . . . . .	69
Schema Tab . . . . .	70
Transport Tab . . . . .	70
<b>Chapter 5 Deploying and Starting an Adapter Using TIBCO Administrator . . . . .</b>	<b>73</b>
Creating an EAR File in TIBCO Designer . . . . .	74

Deploying the Project . . . . .	75
Starting or Stopping the Adapter . . . . .	76
Monitoring the Adapter . . . . .	77
<b>Chapter 6 Configuring Advanced Features . . . . .</b>	<b>79</b>
Defining a TIBCO Hawk Session . . . . .	80
Using Global Variables . . . . .	81
Changing Global Variable Values at Runtime . . . . .	82
Predefined Global Variables . . . . .	83
Configuring a Remote Adapter . . . . .	86
Configuring Multiple Adapter Instances to use Distributed Queues to Process Inbound Messages . . . . .	87
Configuring Multiple Adapter Instances to use Distributed Queues to Process Outbound Messages . . . . .	89
<b>Chapter 7 Monitoring the Adapter Using TIBCO Hawk . . . . .</b>	<b>93</b>
Overview . . . . .	94
Starting TIBCO Hawk Software . . . . .	95
The Auto-Discovery Process . . . . .	96
Invoking Microagent Methods . . . . .	97
Available Microagents . . . . .	99
_onUnsolicitedMsg() . . . . .	103
activateTraceRole() . . . . .	104
deactivateTraceRole() . . . . .	105
getActivityStatistics() . . . . .	106
getActivityStatisticsByOperation() . . . . .	107
getActivityStatisticsByService() . . . . .	109
getActivityStatisticsBySchema() . . . . .	111
getAdapterServiceInformation() . . . . .	112
getComponents() . . . . .	113
getConfig() . . . . .	114
getConfigProperties() . . . . .	115
getConnectionStatistics() . . . . .	116
getHostInformation() . . . . .	117
getPerfMonSetting() . . . . .	118
getPollingInterval() . . . . .	119
getQueueStatistics() . . . . .	120
getRvConfig() . . . . .	121
getStatus() . . . . .	122
getThreadStatistics() . . . . .	123
getTraceSinks() . . . . .	124
getVersion() . . . . .	125
preRegisterListener() . . . . .	126

- resetActivityStatistics() . . . . . 127
  - resetConnectionStatistics() . . . . . 128
  - resetThreadStatistics() . . . . . 129
  - reviewLedger() . . . . . 130
  - setPollingInterval() . . . . . 131
  - setTraceSinks() . . . . . 132
  - stopApplicationInstance() . . . . . 133
  - unRegisterListener() . . . . . 134
- Appendix A Adapter Properties File . . . . . 135**
  - Overview . . . . . 136
  - Properties File Format . . . . . 137
  - Predefined Properties . . . . . 138
  - Obfuscating or Encrypting a Password in a Properties File . . . . . 141
- Appendix B Trace Messages . . . . . 143**
  - Overview . . . . . 144
  - Trace Message Fields . . . . . 145
  - Status Messages . . . . . 147
- Appendix C Message Formats . . . . . 171**
  - Subscription Service Additional Fields . . . . . 172
  - Request-Response Service Additional Fields . . . . . 174
- Appendix D Adapter Agent API References . . . . . 175**
  - TIBCO Rendezvous Adapter Agent Methods . . . . . 176
  - TIBCO EMS Adapter Agent Methods . . . . . 182
- Appendix E FAQs and Troubleshooting . . . . . 185**
  - Frequently Asked Questions . . . . . 186
  - Troubleshooting . . . . . 187
- Index . . . . . 191**

# Figures

Figure 1	Selecting a Schema . . . . .	16
Figure 2	Creating Log Sinks . . . . .	35
Figure 3	Defining a TIBCO Hawk Session . . . . .	80
Figure 4	Defining an RVCMQ Session . . . . .	88
Figure 5	Defining an RVCMQ Session . . . . .	90
Figure 6	Defining PubAgentRvCmqSubscriber . . . . .	91
Figure 7	TIBCO Hawk Enterprise Monitor . . . . .	96
Figure 8	Microagents, Methods and Arguments Dialog . . . . .	97
Figure 9	Invocation Result Dialog . . . . .	98





# Tables

Table 1	General Typographical Conventions . . . . .	xiv
Table 2	Syntax Typographical Conventions . . . . .	xv
Table 3	Tuxedo Server Environment Variables. . . . .	2
Table 4	Tuxedo Client Environment Variables . . . . .	4
Table 5	Adapter Instance Configuration Tab . . . . .	26
Table 6	Adapter Instance Run-time Connection Tab . . . . .	28
Table 7	Initialization Flags Values. . . . .	30
Table 8	Adapter Instance Adapter Services Tab . . . . .	30
Table 9	Adapter Instance General Tab . . . . .	31
Table 10	Adapter Instance MultiThreading Tab . . . . .	32
Table 11	Adapter Instance Logging Tab . . . . .	33
Table 12	Adapter Instance Startup Tab . . . . .	36
Table 13	Adapter Instance Monitoring Tab . . . . .	36
Table 14	Publication Service: Configuration Tab . . . . .	44
Table 15	Publication Service: Publisher Options Tab . . . . .	45
Table 16	Publication Service: Fetch Schema Tab . . . . .	47
Table 17	Publication Service: Publisher Schema View Tab . . . . .	47
Table 18	Publication Service: Schema Tab . . . . .	48
Table 19	Publication Service: Transport Tab . . . . .	48
Table 20	Subscription Service: Configuration Tab . . . . .	51
Table 21	Subscription Service: Subscriber Options Tab . . . . .	53
Table 22	Subscription Service: Fetch Schema Tab . . . . .	54
Table 23	Subscription Service: Subscriber Schema View Tab . . . . .	54
Table 24	Subscription Service: Schema Tab . . . . .	55
Table 25	Subscription Service: Transport Tab . . . . .	55
Table 26	Request-Response Service: Configuration Tab . . . . .	58
Table 27	Request-Response Service: RPC Server Options Tab . . . . .	60
Table 28	Request-Response Service: Fetch Schema Tab . . . . .	61

Table 29 Request-Response Service: Request Schema View Tab ..... 62

Table 30 Request-Response Service: Schema Tab ..... 62

Table 31 Request-Response Service: Transport Tab ..... 63

Table 32 Request-Response Invocation Service: Configuration Tab..... 65

Table 33 Request-Response Invocation Service: RPC Client Options Tab..... 67

Table 34 Request-Response Invocation Service: Fetch Schema Tab..... 68

Table 35 Request-Response Service: Request Schema View Tab ..... 69

Table 36 Request-Response Invocation Service: Schema Tab..... 70

Table 37 Request-Response Invocation Service: Transport Tab..... 70

Table 38 Predefined Global Variables ..... 83

Table 39 Standard Microagent Methods ..... 100

Table 40 Class Microagent Methods ..... 101

Table 41 Predefined Properties ..... 138

Table 42 Tracing Fields ..... 145

Table 43 Subscription Service Additional Fields for the Request Schema..... 172

Table 44 Subscription Service Additional Fields for the Reply Schema..... 173

Table 45 Request-Response Service Additional Fields for the Request Schema ..... 174

Table 46 Request-Response Service Additional Fields for the Reply Schema ..... 174

# Preface

This document describes how to create, configure, and deploy projects using TIBCO ActiveMatrix Adapter for Tuxedo.

## Topics

---

- [Related Documentation, page xii](#)
- [Typographical Conventions, page xiv](#)
- [How to Contact TIBCO Support, page xvii](#)

## Related Documentation

---

This section lists documentation resources you may find useful.

### TIBCO ActiveMatrix Adapter for Tuxedo Documentation

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

- *TIBCO ActiveMatrix Adapter for Tuxedo Concepts* Read this manual to familiarize yourself with the concepts used by this product.
- *TIBCO ActiveMatrix Adapter for Tuxedo Installation* Read this manual to learn how to install TIBCO ActiveMatrix Adapter for Tuxedo.
- *TIBCO ActiveMatrix Adapter for Tuxedo Configuration and Deployment* Read this manual for instructions on creating, configuring , and deploying adapter projects.
- *TIBCO ActiveMatrix Adapter for Tuxedo Examples* Read this manual to work through the examples provided with the adapter.
- *TIBCO ActiveMatrix Adapter for Tuxedo Release Notes* Read the release notes for a list of new and changed features. This document also contains lists of known issues and closed issues for this release.

### Other TIBCO Product Documentation

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

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

## Third-Party Documentation

You may also find it useful to read the Oracle Tuxedo User documentation.

# Typographical Conventions

The following typographical conventions are used in this manual.

Table 1 General Typographical Conventions

Convention	Use
<i>TIBCO_HOME</i> <i>ENV_HOME</i>	<p>Many TIBCO products must be installed within the same home directory. This directory is referenced in documentation as <i>TIBCO_HOME</i>. The value of <i>TIBCO_HOME</i> depends on the operating system. For example, on Windows systems, the default value is C:\tibco.</p> <p>Other TIBCO products are installed into an installation environment. Incompatible products and multiple instances of the same product are installed into different installation environments. The directory into which such products are installed is referenced in documentation as <i>ENV_HOME</i>. The value of <i>ENV_HOME</i> depends on the operating system. For example, on Windows systems the default value is C:\tibco.</p>
code font	<p>Code font identifies commands, code examples, filenames, pathnames, and output displayed in a command window. For example:</p> <p>Use MyCommand to start the foo process.</p>
<b>bold code font</b>	<p>Bold code font is used in the following ways:</p> <ul style="list-style-type: none"><li>• In procedures, to indicate what a user types. For example: Type <b>admin</b>.</li><li>• In large code samples, to indicate the parts of the sample that are of particular interest.</li><li>• In command syntax, to indicate the default parameter for a command. For example, if no parameter is specified, MyCommand is enabled: MyCommand [<b>enable</b>   disable]</li></ul>
<i>italic font</i>	<p>Italic font is used in the following ways:</p> <ul style="list-style-type: none"><li>• To indicate a document title. For example: See <i>TIBCO ActiveMatrix BusinessWorks Concepts</i>.</li><li>• To introduce new terms For example: A portal page may contain several portlets. <i>Portlets</i> are mini-applications that run in a portal.</li><li>• To indicate a variable in a command or code syntax that you must replace. For example: MyCommand <i>PathName</i></li></ul>

Table 1 General Typographical Conventions (Cont?)




Convention	Use
Key combinations	<p>Key name separated by a plus sign indicate keys pressed simultaneously. For example: Ctrl+C.</p> <p>Key names separated by a comma and space indicate keys pressed one after the other. For example: Esc, Ctrl+Q.</p>
	The note icon indicates information that is of special interest or importance, for example, an additional action required only in certain circumstances.
	The tip icon indicates an idea that could be useful, for example, a way to apply the information provided in the current section to achieve a specific result.
	The warning icon indicates the potential for a damaging situation, for example, data loss or corruption if certain steps are taken or not taken.

Table 2 Syntax Typographical Conventions

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

Table 2 Syntax Typographical Conventions

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



## How to Contact TIBCO Support

---

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

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

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

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

<https://support.tibco.com>

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



## Chapter 1 **Preparing Tuxedo**

This chapter explains how to prepare Tuxedo services to interact with the adapter services. It lists the environment variables that need to be set and the steps required to create a Tuxedo configuration file and boot Tuxedo services.

### Topics

---

- [Preparing Tuxedo, page 2](#)

# Preparing Tuxedo

Before configuring the adapter, you must complete the tasks in this section to allow the adapter to access Tuxedo services. Set the required environment variables, boot the Tuxedo services and ensure that they are running successfully. This section explains how to prepare Tuxedo services to interact with the adapter. You need to perform the following tasks to prepare Tuxedo:

- [Task A, Set Environment Variables](#)
- [Task B, Edit the ubbconfig File](#)
- [Task C, Create Tuxedo Configuration Files](#)
- [Task D, Modify the Server Program to Invoke the Agent Method](#)
- [Task E, Create Executable Files](#)
- [Task F, Boot Tuxedo Services](#)



Task D is necessary only when you are using a request-response invocation or agent based publication service.

## Task A Set Environment Variables

Set the following environment variables:

- Tuxedo Server Settings
- Set the following environment variables on the machine where Tuxedo is installed and where the services are to be booted.

Table 3 Tuxedo Server Environment Variables

Name	Description
TUXDIR	Tuxedo installation directory. For example, TUXDIR=C:\bea\tuxedo8.1
WSNADDR	The IP address of the WSL (Workstation Listener) that the client will contact, along with any free port number. This value must be reflected in the ubbconfig file. This variable need not be set for the adapter in the native client mode. For example, WSNADDR=//192.168.213.63:4020 (//IP address:Port number)
TIBCO_RV_HOME	TIBCO Rendezvous installation directory. For example, TIBCO_RV_HOME=TIBCO_HOME\tibrv\8.1

Table 3 Tuxedo Server Environment Variables

Name	Description
APPDIR	Directory where the Tuxedo service executable files are located. For example, APPDIR= <i>TIBCO_HOME</i> \adapter\adtuxedo\ <i>version_number</i> \examples\EventBased\Employee
PATH	The environment variable used to find executables. For example, %TUXDIR%\bin;%APPDIR%;%PATH%;%TUX_ADAPTER_HOME%\hotfix\lib;%TUX_ADAPTER_HOME%\lib;%TIBCO_RV_HOME%\bin
TUXCONFIG	The name of the Tuxedo configuration file to be generated along with the location. The value specified cannot exceed 64 characters for versions of Tuxedo 8.0 or lower whereas it can be up to 256 characters for Tuxedo 8.1 or higher. Please ensure these limits are not exceeded depending on the version of Tuxedo that is being used. For example, TUXCONFIG: %APPDIR%\tuxconfig.
ULOGPFX	The location of Application Log File. For example, %APPDIR%\u\log
<b>When using Tuxedo's queue-based communication, set the following variables:</b>	
QMCONFIG	The queue device. The name and location where the 'que' configuration file will be generated. The 'que' file will contain queue related details like queue manager, list of queues and other parameters like maximum size. Set to %APPDIR%\QUE.
QIPKEY	The IPC key for the queue is the unique identifier associated with a particular queue. Note that the value set for the QIPKEY variable must be different from that specified in the ubbconfig file *RESOURCES IPKEY parameter. To run the adapter as a native client, the environment variable WSNADDR need not be set.
<b>If the Field ID is to be loaded dynamically at runtime, set the following variables</b>	
FLDTBLDIR32	It contains a list of directories (separated by commas) where field tables can be found.
FIELDTBLS32	It contains a list of the files in the table directories (separated by commas) that are to be used.

- Adapter machine (Tuxedo client) settings:

Set the following environment variables on the machine where the adapter is installed.

Table 4 Tuxedo Client Environment Variables

Name	Description
TUXDIR	The Tuxedo installation directory. For example, TUXDIR=C:\bea\tuxedo8.1
APPDIR	The directory where the adapter executable file is located. For example, APPDIR=TIBCO_HOME\adapter\adtuxedo\version_number\bin.
PATH	The environment variable used to find executables. For example, %TUXDIR%\bin;%APPDIR%;%PATH%.
WSNADDR	The IP address of the WSL (Workstation Listener), along with the port number. This value must correspond to that set in the server machine. For example, WSNADDR=//192.168.213.63:4020 (//IP address:Port number). WSNADDR need not be set for a Native client. TUXCONFIG needs to be set instead.
ULOGPFX	The location of the application log file. For example, %APPDIR%/ULOG.
If the Field ID is to be loaded dynamically at runtime, set the following variables	
FLDTBLDIR32	It contains a list of directories (separated by commas) where field tables can be found.
FIELDTBLS32	It contains a list of the files in the table directories (separated by commas) that are to be used.

Task B Edit the ubbconfig File

Edit the `ubbconfig` file to reflect the same values for the environment variables set on the machine with the Tuxedo client and server installation. The `ubbconfig` file contains default values as bracketed items. Change the default values as per your requirements.

Ensure that the values specified for the environment variables are reflected in the `ubbconfig` file. The settings to be modified are as follows:

- **MACHINES Section**
  - Change the machine name to the machine name on which the adapter is installed.
  - TUXDIR—Tuxedo installation directory.
  - APPDIR—Directory where the Tuxedo service executable files are located.
  - TUXCONFIG—The name of the Tuxedo configuration file to be generated along with the location. The value specified cannot exceed 64 characters, so ensure that your directory structure is such that the path and file name together are 64 characters or less.
  - ULOGPFX - The location of Application Log File.
  - MAXWSCLIENTS—The maximum number of clients that can connect to Tuxedo. The default is set to 0 (zero). You must set this to 1 or higher in order to boot the Tuxedo services successfully. The maximum number of clients that can be specified is 32767. This need not be set if it is a native client.



Ensure that MAXWSCLIENTS is set to 1 or higher.

- **Servers Section**
  - CLOPT (modify the IP address to reflect the value specified for the WSNADDR environment variable). If the client that will connect to this server is to be used as a native client, this is not required to be set.

For a Publication service and Subscription service configuration, you also need to set the following:

  - SRVGRP (modify the directory to point to APPDIR)
  - For Event-based communication supported by the adapter Publication and Subscription services, the location of the EventBroker Repository has to be mentioned along with the -f option in CLOPT.



Refer to Tuxedo documentation for details on editing the ubbconfig file.

## Task C Create Tuxedo Configuration Files

Convert the ubbconfig file into a Tuxedo configuration file. Once the Tuxedo configuration file has been generated, it will be placed in the path specified for the TUXCONFIG environment variable.

1. Type either of the following at a command prompt:

```
tmloadcf -y ubbconfig file name
```

or

```
tmloadcf ubbconfig file name
```

(Answer "y" if it prompts you on whether it should proceed)

2. At the command prompt, change the directory to APPDIR and run the following:

```
TIBCO_HOME\adapter\adtuxedo\veriosn_number\examples\EventBased\Employee>tmadmin
```

A ">" prompt is displayed as shown in the sample below:

```
TIBCO_HOME\adapter\adtuxedo\veriosn_number\examples\EventBased\Employee>tmadmin
tmadmin - Copyright (c) 1996-1999 BEA Systems, Inc
Portions * Copyright 1986-1997 RSA Data Security,
All Rights Reserved.
Distributed under license by BEA Systems, Inc.
>
Tuxedo is a registered trademark.
```

3. Create a transaction log (TLOG device), by going to the > prompt and typing echo, then type crdl -b 200 -z  
TIBCO\_HOME\adapter\adtuxedo\veriosn\_number\examples\EventBased\Employee\TLOG and press Enter as shown in the sample below:

```
>echo
Echo now on.

> crdl -b 200 -z
TIBCO_HOME\adapter\adtuxedo\veriosn_number\examples\EventBased\Employee\TLOG
device created on
TIBCO_HOME\adapter\adtuxedo\veriosn_number\examples\EventBased\Employee\TLOG
```



**step 2** and **step 3** need to be followed if a database is being used. They can be ignored for the examples provided with the adapter.

If the adapter has to establish multiple connections with the Tuxedo server, you need to modify the default server settings.

The following settings control access to your applications:

- MAXACCESSORS



- MAXSERVERS
- MAXWSCLIENTS
- The WSL configuration settings

You need to adjust these settings to ensure that the number of allowable client connections is properly configured.



The WSL settings are:

```
WSL SRVGRP=Group SRVID=ID
```

```
CLOPT="-A -- -n // IP address: Port number
```

```
-m <minimum WSHs that will be booted >
```

```
-M <maximum WSHs that will be booted>
```

```
-x < maximum number of clients connected at any one time>"
```

The `-m` and `-M` determine how many WSH processes will be started. If you want to allow more connections, you should increase the `-M` value. Depending on the number of connections to be increased, you have to accordingly increase the `MAXWSCLIENTS` value. The `-x` value is the multiplexing factor. This determines how many connections can be handed by one WSH process.

### Task D Modify the Server Program to Invoke the Agent Method

Complete this task only if you are configuring an adapter with a Request-Response Invocation service, or a Publication service that uses the adapter agent.

1. Modify the Tuxedo service to invoke the method of the Adapter Agent. See [Adapter Agent API References on page 175](#) for details.
2. Make a copy of the `adapteragent.h` file present in the `TIBCO_HOME\adapter\adtuxedo\version_number\include` directory.
3. Paste this file into the directory where the Tuxedo service `.c` file is present.
4. Add the following statement in the Tuxedo service `.c` file,
 

```
#include adapteragent.h, if the service is based on TIBCO Rendezvous.
#include adapteragentJms.h, if the service is based on TIBCO EMS.
```



When the buffer type is FML32, the Adapter Agent wraps the FML32 buffer in a CArray buffer and sends it to the adapter. You should allocate sufficient memory for the reply FML32 buffer in the calling Tuxedo service before calling any of the adapter agent methods that are mentioned above.



Refer to the agent based publication service and request-response invocation service Examples from *TIBCO ActiveMatrix Adapter for Tuxedo Examples*.

### Task E Create Executable Files

To create executables for Tuxedo client and server programs using the `buildclient` and `buildserver` utilities:

1. Type the following at the command prompt in order to generate the binary file for the client program
2. Type the following at the command prompt to create the executable in order to generate the binary file for the server program

— Subscription services, Request-Response services and Event-based Publication services

```
buildserver -o server executable file name -f server.c file name -s service name
```

— Request-Response invocation services and Agent-based Publication services. The service should be built with the `agent.lib`, `libagent.so` or `libagent.sl` file depending on the platform.

```
buildserver -o server executable file name -f server.c file name -s service name  
-l the home directory of TIBCO ActiveMatrix Adapter for Tuxedo\lib\agent library file name
```

(where the Rendezvous Adapter Agent library file name is `agent.lib` on Microsoft Windows, `libagent.so` on Solaris, AIX, HP-UX IA, or `libagent.sl` on HP-UX PA\_RISC)

### Task F Boot Tuxedo Services

Use the following steps to start Tuxedo services and verify that they are active.

1. Type the following command to boot the server:

```
tmboot -y
```

If the above steps are applied to the `EMPLOYEE` example, the following message displays on the screen indicating that the Tuxedo services are being successfully booted:

```
Booting all admin and server processes in  
TIBCO_HOME\adapter\adtuxedo\version_number\examples\EventBased\Em  
ployee\tuxc onfig  
INFO: BEA Tuxedo, Version 8.0  
INFO: Serial #: 650522264137-764259295626, Expiration  
2003-06-03, Maxusers 1000  
INFO: Licensed to: BEA Evaluation Customer  
Booting admin processes ...
```

```

exec BBL -A :
    process id=545 ... Started.
Booting server processes ...
exec TMUSREVT -e tmusrevt.out -o tmusrevt.out -A -- -f
TIBCO_HOME\adapter\adtuxedo\veriosn_number\examples\EventBased\Em
ployee\tmus    revt.dat :
    process id=769 ... Started.
exec TLR -A :
    process id=434 ... Started.
5 processes started.

```

2. Type the following command to verify that all server and services are active:

```
tmadmin
```

A '>' prompt displays if the configuration is active as shown in the sample below.

```

TIBCO_HOME\adapter\adtuxedo\veriosn_number\examples\EventBased\Em
ployee>tmad    min
tmadmin - Copyright (c) 1996-1999 BEA Systems, Inc
Portions * Copyright 1986-1997 RSA Data Security,
All Rights Reserved.
Distributed under license by BEA Systems, Inc.
>
Tuxedo is a registered trademark.

```

If the configuration is not active, the following message displays:

```
No bulletin board exists Entering boot mode
```

```
>
```



## Chapter 2      **Getting Started**

### Topics

---

- [Overview, page 12](#)
- [Preparing Tuxedo, page 13](#)
- [Configuring the Adapter Components, page 14](#)
- [Deploying the Adapter, page 18](#)

## Overview

---

This chapter provides a short exercise on how to configure TIBCO ActiveMatrix Adapter for Tuxedo with a publication service and start and stop the adapter.

### Before Starting

Before starting the exercise, ensure that all required software has been installed and is operating correctly. For a list of required software, see *TIBCO ActiveMatrix Adapter for Tuxedo Installation*.

See [Chapter 1, Preparing Tuxedo](#) for more information on preparing Tuxedo to work with the adapter.

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

This exercise uses the files provided in the agent-based publication example packaged with the adapter. The examples are located in the *TIBCO\_HOME\adapter\adtuxedo\version\_number\examples\AgentBased\PurchaseOrder* folder. Refer to *TIBCO ActiveMatrix Adapter for Tuxedo Examples* for more information about the files used in this exercise.

## Preparing Tuxedo

---

This section describes how to prepare Tuxedo for this example.

Follow these steps to prepare your Tuxedo environment for this example:

1. Go to the  
`TIBCO_HOME\adapter\adtuxedo\veriosn_number\examples\AgentBased\PurchaseOrder` folder.
2. Open the `make.bat` file to modify the following variables. For example, set the following values to the variables:

```
set TUXDIR=c:\bea\tuxedo8.1
set WSNADDR=//192.168.66.44:8100
set TUX_ADAPTER_HOME=TIBCO_HOME\adapter\adtuxedo\veriosn_number
set TIBCO_RV_HOME=TIBCO_HOME\tibrv\veriosn_number
```

3. Open the `ubbagent` file to modify the following variables. For example, set the following values to the variables:

```
*MACHINES
MYMACHINE LMID=SITE3
TUXDIR="c:\bea\tuxedo8.1"
APPDIR="TIBCO_HOME\adapter\adtuxedo\veriosn_number\examples\AgentBased\PurchaseOrder"
TUXCONFIG="TIBCO_HOME\adapter\adtuxedo\veriosn_number\examples\AgentBased\PurchaseOrder\tuxconfig"
ULOGPFX="TIBCO_HOME\adapter\adtuxedo\veriosn_number\examples\AgentBased\PurchaseOrder\ulog"
MAXWSCLIENTS=10
WSL SRVGRP=GROUP1 SRVID=3
CLOPT="-A -- -n //192.168.66.44:8100 -m 1 -M 10 -x 1"
```

4. Navigate to the `APPDIR` directory from a command line and run the `make.bat` file as follows. This sets the environment variables, creates the Tuxedo configuration files, creates the executables, and boots the Tuxedo services.

```
TIBCO_HOME\adapter\adtuxedo\veriosn_number\examples\AgentBased\PurchaseOrder make
```

## Configuring the Adapter Components

---

This section leads you through a quick configuration exercise. This exercise covers the following tasks:

- [Task A, Create a TIBCO Designer Project](#)
- [Task B, Create and Configure an Adapter Instance](#)
- [Task C, Create and Configure a Publication Service](#)
- [Task D, Create and Configure a TIBCO ActiveMatrix BusinessWorks Process](#)

### 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 open an existing project.

1. Start TIBCO Designer and select **New Empty Project**.
2. In the Save Project dialog, click the **Browse** button to select the location of the project and click **OK**. A project is created.

### Task B Create and Configure an Adapter Instance


An adapter instance can contain one or more adapter services. Options for logging, startup and monitoring are set on the adapter instance. In this exercise, default values are used for these options.

1. Drag and drop the **TuxedoAdapterConfiguration** icon from the palettes panel to the design panel. An adapter instance is created and the default instance name is `TuxedoAdapterConfiguration`. You can change the instance name.
2. Identify the file log options in the Logging tab if you want to log trace messages. You can configure the levels of trace messages you want log, and where trace messages are sent.
3. Click the **Apply** button to save the changes.


### Task C Create and Configure a Publication Service


1. In the project panel, expand the **TuxedoAdapterConfiguration** folder, then select the **Adapter Services** folder.
2. Drag and drop the **PublicationService** icon from the palette panel to the design panel.
3. In the Configuration tab, select **Rendezvous** in the Transport Type drop-down list. Leave the other default values unchanged.



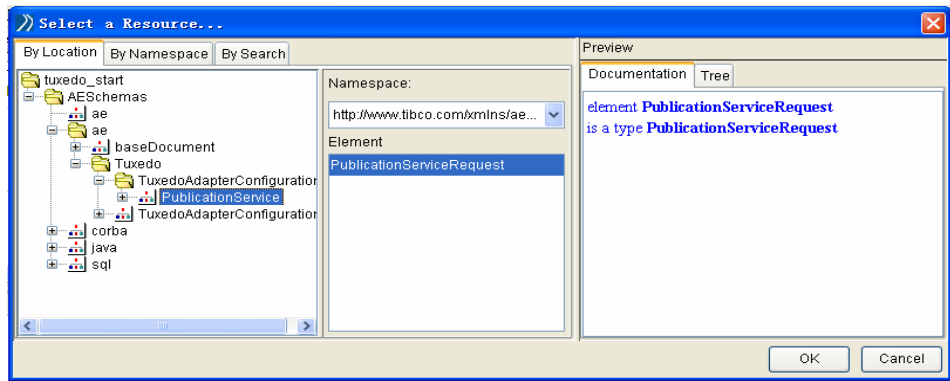
4. In the Publisher Options tab, select **FML32** in the Buffer Type drop-down list. Leave the other default values unchanged. See [Publisher Options Tab](#) for more information about buffer type.
5. In the Fetch Schema tab, click  to specify the header file in the Request Header Reference field. The header file contains the information about the Tuxedo fields. The fields will be included in the schema used for publishing.  
  
In this example, navigate to  
`TIBCO_HOME\adapter\adtuxedo\version_number\examples\AgentBased\PurchaseOrder` and select the `PODetails.h` header file.
6. Click **Apply**.
7. In the Transport Type, enter the subject name. For example, enter `tuxedo_demo` in the Subject Name field.
8. Save the project.

#### Task D Create and Configure a TIBCO ActiveMatrix BusinessWorks Process

1. Select the project created in [Task A](#) in the Project panel.
2. Drag and drop the **Process Definition** icon from the Palette panel to the Design panel.
3. In the Configuration tab, enter the process name. For example, enter `pub` in the Name field.
4. Double-click the `pub` process, drag and drop the **Adapter Subscriber**, **Render XML**, and **Write File** icons from the Palettes panel into the Design panel and create transition between them.
5. Configure the Adapter Subscriber activity.
  - a. Click the **Adapter Subscriber** icon.
  - b. In the Configuration tab, Click  against the Adapter Service field, choose `TuxedoAdapterConfiguration>Adapter Services>PublicationService`, and click **OK**.
  - c. In Transport tab, click the **Refresh Adapter Service** button.

6. Configure the Render XML activity.
  - a. Click the **Render XML** icon.
  - b. In the Configuration tab, select **text** in the Output Style drop-down list.
  - c. In the Input Editor tab, click the **Add Child** icon.
  - d. In the Content drop-down list, select **XML Element Reference**.
  - e. Click  against the Schema field, choose AESchemas>ae>Tuxedo>TuxedoAdapterConfiguration>PublicationService and click **OK** as shown in

*Figure 1 Selecting a Schema*



- f. Click **Apply**.
  - g. In the Input tab, create a connection between PublicationServiceRequest (\$Adapter-Subscriber>ProcessStarterOutput>body>PublicationServiceRequest) in the Process Data pane and PublicationServiceRequest in the Activity Input pane.
7. Configure the Write File activity.
  - a. Click the **Write File** icon.
  - b. In the Input tab, create a connection between xmlString (\$Render-XML>xmlString) in the Process Data pane and textContent (WriteActivityInputTextClass>textContent) in the Activity Input pane.
  - c. In the fileName field under WriteActivityInputTextClass, type `TIBCO_HOME\adapter\adtuxedo\version_number\examples\AgentBased\PurchaseOrder\BusinessWorks\Get_output.xml`
  - d. Click **Apply**.

8. Save the project.

## Deploying the Adapter

---

This section describes how to deploy the runtime adapter from the command line. The adapter instance configured in the last section will be deployed.

To deploy the adapter from the command line, follow these steps:

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, `testpub`.
  - d. Click OK. The `testpub.dat` file is 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. Before deploying the adapter, you need to create a TIBCO Runtime Agent file or edit the default TIBCO Runtime Agent file.

The installation program generates the `adtuxedo_native.tra` and `adtuxedo_wrkstn.tra` files for the TIBCO ActiveMatrix Adapter for Tuxedo. The files are located in the `TIBCO_HOME\adapter\adtuxedo\version_number\bin` directory. You need to use the `adtuxedo_wrkstn.tra` file if the adapter works as a Tuxedo Workstation client and use the `adtuxedo_native.tra` file if the adapter works as a Tuxedo Native client.

In this example, the adapter works as a workstation client. You need to make a copy of the `adtuxedo_wrkstn.tra` file. You can rename it, for example `test.tra`, and update the related variables. At a minimum, the following variables must be updated:

`tibco.repourl`—pathname of the TIBCO Designer project DAT file

`tibco.configurl`—name of the adapter configuration

`application.args`—properties file to pass to the application

`tibco.env.WSNADDR`—The IP address of the WSL (Workstation Listener) that the client will contact, along with a free port number. This value must be reflected in the `ubbconfig` file.

If the Field ID is to be loaded dynamically at runtime, set the following environment variables:

FLDTBLDIR32 — contains a list of directories (separated by commas) where field tables can be found

FIELDTBLS32 — contains a list of the files in the table directories (separated by commas) that are to be used

For this example, update the variables as follows:

```
tibco.repourl E:/temp/testpub.dat
tibco.configurl TuxedoAdapterConfiguration
tibco.env.WSNADDR //192.168.66.44:8100
application.args -system:propFile
                %TIB_ADTUXEDO_HOME%/bin/test.tra
```

### 3. Deploy the adapter instance from the command line.

You can run the adapter as a console application using a custom properties file. The commands given next start the adapter service that is identified in the `test.tra` properties file, which is located in the same directory as the executable. The absolute pathname to the properties file must be given if it is located in a different directory than the executable.

Before deploying the adapter, you can start the TIBCO ActiveMatrix BusinessWorks process configured in [Task D](#). The process starts to work after receiving the messages from the publication service of the specified adapter.

To deploy the adapter, follow these steps:

- a. In another command window, navigate to the `APPDIR` directory. In this example, the `APPDIR` directory is  
`TIBCO_HOME\adapter\adtuxedo\version_number\examples\AgentBased\PurchaseOrder.`
- b. Run the `make.bat` file.
- c. Navigate to the `TIBCO_HOME\tibco\adapter\version_number\bin\` directory in a command line window.
- d. Type `adtuxedo_wrkstn -system:propFile test.tra` to start the adapter instance.
- e. Run the client file. For example, run the client by typing `c1t 1000` in the command line.

`c1t` is a request sent from the Tuxedo client. `c1t` invokes the Tuxedo service that sends a request to the adapter through the adapter agent. The adapter publishes the data to the TIBCO environment.

### 4. Stop the adapter.

From a command prompt you can stop the adapter by publishing a message on the `adtuxedo_termination` subject or type **Ctrl-C** in the window where the adapter was started.

## Chapter 3 **Adapter Instance Options**

This chapter explains how to create and configure an adapter instance. All the tasks are performed using TIBCO Designer.

### Topics

---

- [Overview, page 22](#)
- [Adapter Instance Tabs, page 26](#)

## Overview

---

Please read the following sections before starting to configure an adapter.

- [Configuration Tasks](#)
- [Multithreading](#)
- [Connection Management](#)
- [Internationalization](#)
- [Saving the Project](#)
- [Testing the Adapter](#)

### Configuration Tasks

Use the following sequence to create and configure an adapter.

1. Prepare the Tuxedo environment. See [Preparing Tuxedo](#) for details.
2. Start TIBCO Designer and open a multi-file project. See the TIBCO Designer documentation for details.
3. Drag and drop the **TuxedoAdapterConfiguration** icon from the palette panel to the design panel. This creates an adapter named, by default, `TuxedoAdapterConfiguration`.
4. Configure the adapter instance. See [Adapter Instance Tabs](#) for more information.
5. 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.
6. Configure the adapter services for the adapter instance. See [Chapter 4, Adapter Service Options](#) for more information.
7. Set the combination of options required for your service at the service level.
8. Save the project and exit TIBCO Designer.

After configuring the adapter, create the runtime adapter property file and add the project name and the adapter instance name.



## Multithreading

Multithreading allows the adapter to process multiple messages concurrently to ensure better performance. TIBCO ActiveMatrix Adapter for Tuxedo supports multithreading for the publication, subscription, request-response invocation, and synchronous request-response services. Multithreading is not supported for asynchronous request-response services.

You can configure the number of threads for each session in the Multithreading tab at instance level. By default, only one thread is available for the adapter instance. There are no dispatchers for the individual sessions. For high volume messaging environments, you can improve the performance by spawning the dispatchers (threads) for individual sessions. The number of dispatchers depends on how Tuxedo responds to the dispatchers and the machine on which Tuxedo runs. While configuring the adapter services, ensure that both inbound (subscription or request-response) and outbound (publication and request-response invocation) services are not configured under the same session. See [MultiThreading Tab](#) for more information.

TIBCO ActiveMatrix Adapter for Tuxedo enables multiple connections to interface with the Tuxedo application. This feature is supported only by the Tuxedo application version 7.1 or higher. Every thread creates its own connection with the Tuxedo application, allowing one to one association between threads and connections. The thread maintains this connection throughout the lifetime of the adapter till the connection is lost with the Tuxedo application.

If the adapter has to establish multiple connections with the Tuxedo server, you need to modify the default server settings. For details on modifying the server settings, see [Create Tuxedo Configuration Files](#).

When configuring multithreading for the publication and request-response invocation services, you should note the following situations:

- For the publication service, when the Event-based, Queue-based, or Rendezvous Adapter Agent-based communication paradigms is used, only the number of threads configured in the `DefaultRVSession` session is valid. When the communication is based on JMS adapter agent, the number of the threads configured in the `DefaultJmsTopicSession` or `DefaultJmsQueueSession` session is valid.
- For the request-response invocation service, when the communication is based on the Rendezvous adapter agent, only the number of threads configured in the `DefaultRVSession` session is valid.

## Connection Management

An application or connection error may cause a loss of the connection between TIBCO ActiveMatrix Adapter for Tuxedo and Tuxedo application.

- If the loss is caused by an application error, the adapter terminates.
- If the loss is caused by a connection error, the adapter will attempt to reconnect to the Tuxedo application. The maximum number of reconnect attempts can be specified.

You can specify the parameters for connection management in the [Run-time Connection Tab](#) at the instance level.

## Internationalization

See *TIBCO ActiveMatrix Adapter for Tuxedo Concepts* for an introduction to internationalization and how adapters handle it. See [General Tab](#) for setting encoding options for an adapter instance.

Before running the TIBCO ActiveMatrix Adapter for Tuxedo in Unicode mode, you must perform the following steps:

1. Make sure that the appropriate locales exist and are valid on your operating system. To verify, do the following.

For Microsoft Windows, verify the registry entries under

`\\HKEY_LOCAL_MACHINE\\SYSTEM\\ControlSet001\\Control\\Nls\\Codepage`

2. Copy the locale files for the relevant code page to the appropriate directory. For Microsoft Windows, the directory is `WINDOWS\\system32`.
3. When configuring an adapter, specify the locale in the Adapter Encoding drop-down list in the General tab for the adapter instance. If the locale is not specified correctly, the adapter will not process the character data correctly.
  - For Microsoft Windows, an input locale must exist on the operating system and the correct code page file must exist in the `\\WINDOWS\\system32` directory. To view the set of possible input locales, refer to the Regional Setting Properties available from the Control Panel.
  - For Solaris 8, verify locale files in the `/usr/lib/locale` directory.

## Saving the Project

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

For detailed steps and more information about exporting or importing projects to different formats (such as `.dat`), see the *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 Tabs

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

- [Configuration Tab](#)
- [Run-time Connection Tab](#)
- [Adapter Services Tab](#)
- [General Tab](#)
- [MultiThreading Tab](#)
- [Logging Tab](#)
- [Startup Tab](#)
- [Monitoring Tab](#)

### Configuration Tab

After you drag and drop a **TuxedoAdapterConfiguration** icon from the palette panel to the design panel, the Configuration tab is selected by default.

Click **Apply** to apply the changes before leaving this tab.

Table 5 Adapter Instance Configuration Tab

Field	Description
Instance Name	The name of the Adapter Service Engine instance. Use the default name or replace it with a name of your choice. See <a href="#">Guidelines for Choosing an Instance Name</a> for more information.
Description	A short description of the adapter configuration.

Table 5 Adapter Instance Configuration Tab

Field	Description
Version	<p>The version string indicates the TIBCO ActiveEnterprise (AE) configuration format in which the adapter instance is saved. An adapter instance can be saved in AE 4.0, AE 5.0 or AE fro1 format.</p> <p>When a new adapter instance is created in TIBCO Designer 5.x, the version string is set to AE Version 5.1. When a 4.x adapter instance is opened in Designer 5.x, the Version field is set to AE Version 4.0.</p> <p>If a 4.x adapter instance is to be run against a 4.x run-time adapter, the instance must be saved with the Version field set to AE Version 4.0.</p> <p>If you are using TIBCO Designer 5.x to modify 4.x adapter instances, only change features supported by the 4.x. runtime adapter and use the validation utility to verify the instance before deploying the project. Invoke the utility from the Project&gt;Validate Project for Deployment menu command in Designer.</p> <p>To change versions, click the <b>Change Version</b> button.</p>
Message Filter	<p>Specify a message filter, if you have configured a message filter 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 the TIBCO Adapter SDK. See <i>TIBCO Adapter SDK Programmer's Guide</i> for information about writing a message filter.</p>
Show All Tabs	<p>Check this checkbox to display additional tabs for configuring advanced options.</p>
Choose Tuxedo Client Type Installed	<p>Select <b>adtuxedo_wrkstn</b> if the adapter works as a Tuxedo Workstation client.</p> <p>Select <b>adtuxedo_native</b> if the adapter works as a Tuxedo Native client.</p>

### Guidelines for Choosing an 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.
- 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 the next time you rename the adapter instance.

## Run-time Connection Tab

Many of the following fields can use global variables. Click the **Global Variables** tab in the project panel to add or modify a global variable. See [Using Global Variables](#) for more information

The parameters that need to be specified in the Run-time Connection tab depend on the authorization level set on the Tuxedo server (ubbbconfig file). The authorization levels and the fields to be specified for each are as follows:

- TPNOAUTH: User Name
- TPSYSAUTH: User Name and Application Password
- TPAPPAUTH: User Name, Application Password, User Password, Client Name, Group Name (should be null if the adapter is to be used as a workstation client) and Initialization flags.

The `adtuxedo.authinfo` property in the adapter properties file indicates the authorization level set on the Tuxedo server.

There are three possible values:

- TPNOAUTH: No authorization check. This is the default value.
- TPSYSAUTH: System authorization.
- TPAPPAUTH: Application authorization.

If the parameter is not specified, the adapter will attempt to connect to each of these levels sequentially.

Click **Apply** to apply the changes before leaving this tab.

Table 6 Adapter Instance Run-time Connection Tab

Field	Description
User Name	The name of a valid client user to log into the Tuxedo application. The user name should less than 30 characters.

Table 6 Adapter Instance Run-time Connection Tab

Field	Description
User Password	<p>The password associated with the Tuxedo user specified in the User Name field. This field cannot be edited. If a value has to be specified, use the Global Variables. This is the variable length data that is forwarded to the application defined authentication service.</p> <p>You can toggle between global variables and plain text values. This value is visible in non-readable form in the project. Plain text values are stored in the project in encrypted form.</p>
Group Name	<p>When the adapter is used as a workstation client, the value in this field should be set to null, which is the default. When the adapter is used as a native client, specify a valid Tuxedo group name in this field.</p>
Application Password	<p>The password associated with a particular application domain. All users in a domain will use this password to access the application, depending on the level of authorization set on Tuxedo. The password should be less than 30 characters. This field cannot be edited. If a value has to be specified, use Global Variables.</p> <p>You can toggle between global variables and plain text values. This value is visible in non-readable form in the project. Plain text values are stored in the project in encrypted form.</p>
Initialization Flags	<p>The flags that indicate the client-specific notification mechanism and mode of system access. See <a href="#">Initialization Flags Values</a> for more information.</p>
Maximum Number of Reconnect Attempts	<p>The total number of reconnection attempts to make after the service has been suspended. When this number is reached, the runtime adapter or adapter service will be stopped.</p> <p>A value of -1 means reconnection attempts will continue indefinitely.</p>
Number of Reconnect Attempts Before Suspending Impacted Service(s)	<p>Specify the number of reconnection attempts to make before suspending the service.</p> <p>Note that reconnection with the application does not depend on this parameter. Reconnection depends on the value specified in the Maximum Number of Reconnect Attempts field.</p>
Interval between Reconnect Attempts (milliseconds)	<p>Specify the time interval (in milliseconds) to elapse between each reconnection attempt.</p>
Adapter Termination Criteria (after max number of reconnect attempts)	<p>The adapter and all of its services are stopped if any one of its services has been unable to re-establish connection after the Maximum Number of Reconnect Attempts has been made. This option cannot be changed.</p>

### Initialization Flags Values

The valid values in the Initialization Flags field are given in [Table 7](#). These values are used while initiating connections to Tuxedo. The values are set in the TPINIT structure. Refer to the Oracle Tuxedo documentation for more information.

Table 7 Initialization Flags Values

Flag	Value
TPU_SIG	1
TPU_DIP (default)	2
TPU_IGN	4
TPSA_FASTPATH	8
TPSA_PROTECTED	10
TPMULTICONTEXTS	20
TPU_THREAD	40
TPU_MASK	47

### Adapter Services Tab

These settings affect publication and request-response services defined for the adapter. Click **Apply** to apply the changes before leaving this tab.

Table 8 Adapter Instance Adapter Services Tab

Field	Description
All Publication Services	
Polling Interval (milliseconds)	This parameter determines how often the adapter publication service polls the Tuxedo EventBroker. The default value is 3000 milliseconds.
All Request-Response Services	
Polling Interval for Replies (milliseconds)	This parameter determines how often the request-response service polls for a reply from Tuxedo in an asynchronous mode. The default value is 1000 milliseconds. This field can be set as a global variable or it can be set in the TRA properties file.



Table 8 Adapter Instance Adapter Services Tab

Field	Description
Polling Interval for Outstanding Invocations (milliseconds)	<p>This parameter determines how often the Request-Response service polls Tuxedo in an asynchronous mode for the outstanding invocations. The default setting is 2000 milliseconds.</p> <p>Example: If you have not changed default settings, the Request-Response service in asynchronous mode polls for replies every 1000 milliseconds till the maximum number of pending invocations configured is reached after which polling is done every 2000 milliseconds. This field can be set as a global variable or it can be set in the TRA properties file.</p>
Maximum Number of Pending Invocations	<p>Specify the maximum number of pending invocations. Once maximum number of pending invocations is reached, the value specified in the polling interval for outstanding invocations field is used to poll for replies. The default setting is 50. Tuxedo allows a maximum of 50 pending invocations and any subsequent additional invocation will yield a <code>TPELIMIT</code> error. This field can be set as a global variable or it can be set in the TRA properties file.</p>

## General Tab

This tab allows you to set a termination subject or topic and specify the encoding type. Note that the adapter should communicate only with other applications that support the same code pages or Unicode.

Click **Apply** to apply the changes before leaving this tab.

Table 9 Adapter Instance General Tab

Field	Description
Termination Subject or Topic	<p>A message sent on this 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.</p> <p>See the TIBCO Rendezvous documentation for information about specifying subject names. See the TIBCO Enterprise Message Service documentation for information about publishing on a topic.</p>

Table 9 Adapter Instance General Tab

Field	Description
Adapter Encoding	<p>Select the encoding from the drop-down list. The default encoding is ASCII. If you do not set a value in this field, the default value (ASCII) is used.</p> <p>Here are the encodings available in the drop-down list:</p> <ul style="list-style-type: none"><li>• <b>ASCII</b> 7-bit ASCII</li><li>• <b>UTF8</b> Unicode Transformation Format-8</li><li>• <b>Shift_JIS (CP943)</b> Japanese Shift-JIS, CP943</li><li>• <b>Shift_JIS (TIBCO)</b> Variant of IBM-943, flavoring some MS-932 conversions</li><li>• <b>Big5</b> Traditional Chinese Big5 (with Euro Sign)</li><li>• <b>ISO8859-1</b></li></ul> <p>The Adapter Encoding field is editable. See <i>TIBCO ActiveMatrix Adapter for Tuxedo Concepts</i> for a list of additional encodings that can be typed into this field.</p> <p>The palette does not validate encoding values that you type into the field. The runtime adapter will throw an error if the encoding value you type is not supported.</p>

MultiThreading Tab

Every adapter instance has one or more sessions linked to it. Sessions encapsulate stateful connections to a messaging source, such as a TIBCO Rendezvous daemon. When you configure an adapter instance, sessions and the associated endpoints are created automatically. Click **Advanced** in the project panel, then click **Sessions** to view the sessions created. Click the required endpoint to view the parameters for the endpoint.

Click **Apply** to apply the changes before leaving this tab.

Table 10 Adapter Instance MultiThreading Tab

Field	Description
Session Name	<p>The session name for which multithreading is being set. The <code>DefaultRVSession</code> and <code>HawkSession</code> sessions are set by default.</p>

Table 10 Adapter Instance MultiThreading Tab

Field	Description
No. of Threads	<p>The number of threads to be used for the session.</p> <p>If using Oracle Tuxedo 6.5, the number of threads should be set to zero. If using Oracle Tuxedo 7.1 or higher, the adapter supports multiple connections for inbound services. The number of connections made to the Tuxedo application will be equal to one more than the number of threads configured. For example: If you set 3 for the number of threads, the number of connections made is 4. The minimum value for number of threads in the Hawk session is one and for other sessions, it is zero.</p>

## Logging Tab

Use these settings to configure a log file or log sinks, including which types of trace messages you want logged and where they are sent.

Click **Apply** to apply the changes before leaving this tab.

Table 11 Adapter Instance Logging Tab

Field	Description
Use Advanced Logging	<p>When unchecked, the standard log file is used. This is the default. Fill out the remaining fields on this tab. You do not need to read the rest of this field description.</p> <p>When checked, you can set two standard output destinations (sinks) for trace messages and set the tracing level for the roles selected. The following sink types are available:</p> <ul style="list-style-type: none"> <li>• File</li> <li>• STDIO</li> <li>• Hawk</li> <li>• Network</li> </ul> <p>See <a href="#">Creating Log Sinks</a> for more information.</p>
Log to Standard I/O	<p>When checked, trace messages are displayed in the command prompt window where the adapter is started. This is the same as creating a STDIO sink. When unchecked, trace messages do not display in the window.</p>
Log File	<p>Specify the name of the log file to which trace messages are written. This is the same as creating a file sink. Global variables can be used to specify the location of the log file. See <a href="#">Using Global Variables</a> for more information.</p> <p>Type the name and file system path, or click <b>Browse</b> and select an existing log file. If no file name is specified, trace information is not written to a file.</p>

Table 11 Adapter Instance Logging Tab

Field	Description
Log Info/ Debug/ Warning/ Error Messages	Select the types of trace messages you want logged.

By default all error, warning, debug and information messages are printed in the console window in which the adapter was started. Alternatively, you can specify a log file and path to redirect trace output to a log file located anywhere in your file system. The default log file name is `%%DirTrace%%/%%Deployment%%.%%InstanceId%%.log`, and is saved in the same directory where your project (repository instance) is stored.

Most errors received by the adapter are logged. The only errors that might not be logged are any TIBCO Rendezvous or TIBCO Adapter SDK errors that appear at startup time before tracing can be initialized.

Logging trace messages is helpful for troubleshooting. There are four levels of trace messages that you can log: *Information*, *Warning*, *Debug*, and *Error*. Trace messages are described and listed in [Appendix B, Trace Messages](#)

Logging affects system performance. It is recommended that you use logging only as needed.

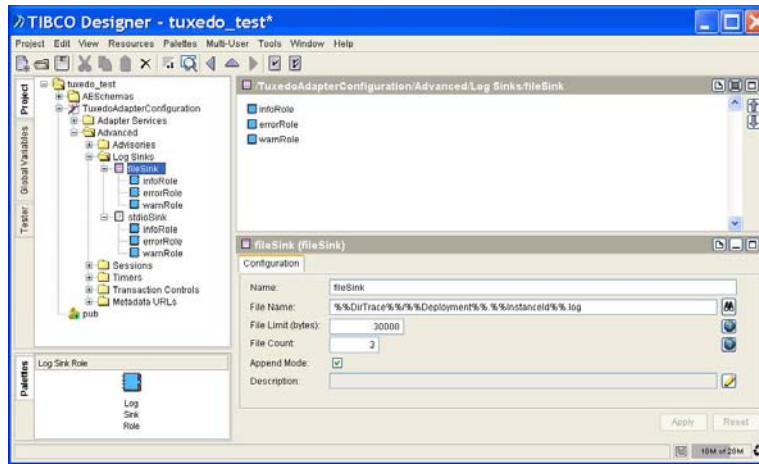
Debug messages should not be logged unless requested by the TIBCO Product Support Group. This option writes a lot of information to the log file and significantly reduces the speed of the adapter.

Creating Log Sinks

When you check the **Use Advanced Logging** checkbox, you configure log sinks using icons in the TIBCO Designer project panel. This gives you complete control on selecting the destinations and associating desired roles with each of the destinations.

1. Check the **Use Advanced Logging** box.
2. Click **Apply**.
3. In the TIBCO Designer project panel, select the **Log Sinks** folder under the Advanced folder.

Figure 2 Creating Log Sinks



4. Select an existing log sink or create a new one:
  - Selecting the File or STDIO log sink icon.
  - Creating a new log sink by dragging and dropping the **Generic log sink** icon from the palette panel into the design panel, then assigning a type to it from the drop down menu in the configuration panel. Click **Apply**.
5. With the desired log sink icon selected in the design panel, fill in the fields in the configuration panel. You can also change the name and enter a description for each sink by right-clicking on the sink icon in the project panel.
  - File sink logs the trace messages to a file. Specify the file limit, file count, and the option to append or overwrite. By default, the file limit is 30000 bytes, the file count is 3, and the mode is append.
  - STDIO sink sends trace messages to stdout or stderr. By default, stdout is selected.
  - Hawk sink sends each trace message to TIBCO Hawk Monitor or TIBCO Hawk Display using the Hawk session, which is created by the adapter for monitoring purposes. Specify the `MicroAgent` Name. (For details on Hawk sessions, see [Using Global Variables](#).)
  - Network sink publishes each trace message on TIBCO Rendezvous. Specify the session and the subject on which the trace messages need to be published.

## Startup Tab

This tab displays the startup behavior.

Table 12 Adapter Instance Startup Tab

Field	Description
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. Check this checkbox to display the startup banner.
Metadata Search URL	This field is predefined and cannot be changed. It specifies the location where the adapter searches for base schemas. All schemas that have been defined and saved at this location are loaded at startup.

## Monitoring Tab

These settings do not need to be configured unless TIBCO Hawk is installed.

Many of the following fields can use global variables. Click the **Global Variables** tab in the project panel to add or modify a global variable.

See [Chapter 7, Monitoring the Adapter Using TIBCO Hawk](#) for a list of supported microagents.

Click **Apply** to apply the changes before leaving this tab.

Table 13 Adapter Instance Monitoring Tab

Field	Description
Enable Standard Microagent	Allows you to turn on or off the standard TIBCO Hawk Microagent. Clicking the globe icon changes the checkbox to a text field, allowing you to specify a global variable. When this is a text field, turn the microagent on and off by entering true or false.
Standard Microagent Name	<p>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, COM.TIBCO.ADAPTER.adtuxedo.%%Deployment%%.%%InstanceId%%.</p> <p>The value for the %%deployment%% global variable can be set or modified by selecting the session icon and then clicking the <b>Global Variables</b> tab in the project panel.</p> <p>The %%InstanceId%% variable does not need to be set because it is automatically set at runtime by the runtime adapter.</p>

Table 13 Adapter Instance Monitoring Tab

Field	Description
Standard Microagent Timeout(ms)	Specifies the amount of time the Hawk Agent should wait for the HMA method invocations to complete before timing them out. The default is 10000 milliseconds. Normally there is no need to change this value, however, on machines under extreme stress where method invocations are timing out, this new option allows the timeout value to be increased.
Enable Class Microagent	Allows you to turn on or off the instance-specific or class-specific standard TIBCO Hawk Microagent. You can configure how the class microagent is turned on and off in this field: clicking the globe icon changes the checkbox to a true/false text field.
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, COM.TIBCO.adtuxedo.%%Deployment%%.%%InstanceId%%.
Class Microagent Timeout(ms)	Specifies the amount of time the Hawk Agent should wait for HMA method invocations to complete before timing them out. The default is 10000 milliseconds. Normally there is no need to change this value, however, on machines under extreme stress where method invocations are timing out, this new option allows the timeout value to be increased.
Default Microagent Session	<p>This is the name of the TIBCO Rendezvous session that is predefined and cannot be changed. The session is automatically generated by TIBCO Designer and will be used by the standard, class, and custom microagents.</p> <p>However, you can create a new session if required. Navigate to the Sessions folder under the Advanced folder to create a new session. See <a href="#">Defining a TIBCO Hawk Session</a> for details.</p> <p>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.</p>





## Chapter 4      **Adapter Service Options**

### Topics

---

- [Overview, page 40](#)
- [Publication Service Tabs, page 44](#)
- [Subscription Service Tabs, page 51](#)
- [Request-Response Service Tabs, page 58](#)
- [Request-Response Invocation Service Tabs, page 65](#)

## Overview

---

TIBCO ActiveMatrix Adapter for Tuxedo supports publication, subscription, request-response, and request-response invocation services.

After configuring an adapter instance, select the adapter instance, then click **Adapter Services**. Add one or more services to the adapter instance by dragging and dropping a **Publication Service**, **Subscription Service**, **Request-Response Service** or **Request-Response Invocation Service** icon from the palette panel to the design palette.

## Message Transports

The transport type (Rendezvous or JMS) you select for the runtime adapter determines which quality of service, delivery mode, connection factory type, and wire format the service can use.

### Quality of Service

This is the level of service that determines how messages are sent.

Possible values are:

- Reliable

(TIBCO Rendezvous transport type only) 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. Messages are received without explicit confirmation.

- Certified

(TIBCO Rendezvous transport type only) Certified message delivery. Offers stronger assurances of message receipt, along with tighter control, greater flexibility and fine-grained reporting. 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.

If certified message delivery is used, data is stored in a ledger file. The size of the ledger depends on several factors, the most important of which is the retention rate of stored data. That is, the ledger grows fastest in response to

the cumulative length of undeliverable messages. You must ensure that sufficient disk space is available for the expected size of the ledger.

- **Distributed Queue**

(TIBCO Rendezvous transport type only) 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 `DistributedQueue` 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*.

- **Transactional**

This option is no longer supported. The option is only included for backward compatibility.

## Delivery Mode

(JMS transport type only) The method of delivery for a JMS message. 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.

- For publication and request-response invocation service, the delivery modes are:

- **Persistent**

In general, a message marked persistent will be available to a JMS client even if the TIBCO Enterprise Message Service server goes down. Persistent messages are held in secondary storage in the server and have guaranteed delivery when sent to a topic that has durable subscribers. (If a topic has no durable subscribers, there are no subscribers that need messages resent in the event of a server failure and therefore messages do not need to be saved.) Performance is improved because disk I/O is not required.

- **Non-Persistent**

(JMS transport type only) A message marked non persistent will not be available to a JMS client if the TIBCO Enterprise Message Service server goes down. These messages are never written to persistent storage.

- For subscription and request-response service, the delivery modes are:

- **Durable**

Indicates that the service is registered with the EMS server. Messages sent to a durable Subscription Service are held by the EMS 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

Indicates that the service is not registered with the EMS server. Messages sent to a non-durable Subscription Service are not held by the EMS server. If the service is down, it will not receive the messages that arrived at the EMS server while the service was down.

## Connection Factory Type

(JMS transport type only) The JMS mode of transport supports the following messaging protocols (TIBCO Enterprise Message Service must be installed to use the JMS transport):

- 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 type of message protocol is also known as broadcast messaging because messages are sent over the network and received by all interested subscribers. This messaging model is known as publish-subscribe.

- Queue

A message sent to a queue is consumed by one receiver. Each message has only one receiver though multiple receivers may connect to the queue. The first receiver to access the queue receives the message and the other receivers do not. A queue retains all messages sent until a certain time the messages are consumed or expired. This message protocol is known as point-to-point.

## Wire Formats

Services must use the same wire format to exchange data.

- Rendezvous Message (TIBCO Rendezvous transport type only)

A self-describing wire format used by TIBCO Rendezvous applications. Control information for validation is *not* sent in the message. If you use this format, the adapter is compatible with adapters not developed with TIBCO Adapter SDK.

- XML Message (TIBCO Rendezvous and JMS transport types)

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

- ActiveEnterprise Message (TIBCO Rendezvous transport type only)

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

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

# Publication Service Tabs

The publication service sends messages from a Tuxedo application to the applications configured for the TIBCO environment.

You can configure the parameters for the Publication service using the following tabs:

- [Configuration Tab](#)
- [Publisher Options Tab](#)
- [Fetch Schema Tab](#)
- [Publisher Schema View Tab](#)
- [Schema Tab](#)
- [Transport Tab](#)

## Configuration Tab

You can specify a name and select the transport type for a publication service in this tab. Click **Apply** to apply the changes before leaving this tab.

Table 14 Publication Service: Configuration Tab

Field	Description
Name	<p>You can use the default name or replace it with a name of your choice.</p> <ul style="list-style-type: none"><li>• A service name must use alphanumeric characters. An underscore ( <code>_</code> ) character can be used. The entire instance name must be less than 80 characters. The space character cannot be used in an instance name.</li><li>• A service name cannot use global variables.</li></ul>
Description	<p>Provide information about the publication service. This field is optional.</p>
Transport Type	<p>Select the transport type (JMS or TIBCO Rendezvous) to be used by the runtime adapter. This selection determines which options appear in the <a href="#">Transport Tab</a>.</p> <p>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.</p> <p>To enable and configure SSL, in the Project panel, expand the <b>Advanced</b> folder, then expand the <code>Sessions</code> folder. Select the TIBCO Rendezvous session or JMS topic and click <b>Use SSL?</b>. The SSL configuration options are explained in the online help associated with the session dialog. Click the question mark to display the online help.</p>

# Publisher Options Tab

After you are done with the configuration, click **Apply** to apply the changes before leaving this tab.

Table 15 Publication Service: Publisher Options Tab

Field	Description
Buffer Type	<p>Three options are available:</p> <ul style="list-style-type: none"> <li> <b>FML32</b> <p>If the buffer type is FML32, the FML32 header files need to be selected in the <a href="#">Fetch Schema Tab</a>. The schema to be used is created based on the header file.</p> </li> <li> <b>String</b> <p>A class named <i>service name</i> Request is created under Schemas/classes/Tuxedo/<i>adapter instance name</i>/<i>service name</i>. The class has a String data type attribute named <i>in</i>. See <a href="#">Appendix C, Message Formats</a> for more information.</p> </li> <li> <b>CArray</b> <p>A class named <i>service name</i> Request is created with a binary data type attribute named <i>in</i>.</p> <p>Note that the automatically created classes for CArray and String buffer types should not be manually edited.</p> </li> </ul>
Event Name	<p>This field appears only if you select <b>FML32</b> in the Buffer Type drop-down list. It is the target name of the Tuxedo event for which the adapter is listening, in order to trigger a publish.</p> <p>This field is mandatory for the Event-based publication service. It is not required for Rendezvous Adapter Agent-based publication or JMS Adapter Agent-based publication services.</p> <p>Event-broker publication service only supports the FML32 buffer type. The Adapter Agent supports all buffer types (FML32, String, CArray).</p> <p>For an Event-based publisher, the EVENT_NAME field is a reserved FML32 field for the name of the event that is being posted on the Event Broker, set by the Tuxedo service or client.</p>
Queue Space	<p>It is the name of a queue space. The Oracle Tuxedo administrator creates the queue space, which contains a collection of queues. Each queue space is a resource manager(RM) instance.</p>
Queue Name	<p>It is the name of a queue. The queue name is a NULL-terminated string of up to 15 characters.</p>

Table 15 Publication Service: Publisher Options Tab

Field	Description
Agent Transport Type	<div><p>Choose one of the following:</p><ul style="list-style-type: none"><li>Rendezvous</li></ul><p>Rendezvous indicates that TIBCO Rendezvous will be used to transport messages.</p><li>JMS QUEUE</li><li>JMS TOPIC</li><li>Distributed Queue</li><p>Distributed Queue indicates that load balancing will be used for Rendezvous Adapter Agent-based publication services. Multiple publication services will run in the RVCMQ mode to process messages passed by the Rendezvous Adapter Agent.</p><p>After selecting <b>Distributed Queue</b>, a <code>DefaultRVCMQSession</code> session is created in the Project. You can modify the parameters of this default session if needed.</p><p>A <code>PubAgentRvCmqSubscriber</code> endpoint is automatically created upon the creation of the <code>DefaultRVCMQSession</code> session for the publication service. You can configure the subject name for this communication by clicking the <code>PubAgentRvCmqSubscriber endpoint</code> under <code>Advanced &gt; Session &gt; DefaultRVCMQSession</code> in the Project panel. Do not manually modify any other attribute of this Endpoint.</p><p>See <i>TIBCO Rendezvous Concepts</i> for details about RVCMQ.</p></div>
Load Field Id Dynamically	<div><p>This checkbox appears only if you select FML32 in the Buffer Type drop-down list. It specifies whether the Field IDs for the FML32 fields should be loaded dynamically at runtime. If unchecked, the Field ID values loaded from the header files are used.</p></div>

Fetch Schema Tab

The Fetch Schema tab is available only after you select **FML32** in the Buffer Type drop-down list in the Publisher Options tab and click **Apply**.



Click **Apply** to apply the changes before leaving this tab.

Table 16 Publication Service: Fetch Schema Tab

Field	Description
Request Header Reference	<p>Browse and select the header file containing information about the fields to be included in the schema. The schema is created based on the FML32 header files that you have selected and is used for publishing.</p> <p>If you made any changes to the header files, click the <b>Refresh</b> button to load the new schema. The sequence names that you have specified in the <a href="#">Publisher Schema View Tab</a> remain unchanged.</p> <p>The schema fields are created with names suffixed with <i>_field ID</i>. Irrespective of the type of the FML32 fields, the schema is always the string type. The classes that are created are stored in the <code>tibco/public/class/ae/Tuxedo</code> directory.</p>

**Publisher Schema View Tab**

The Publisher Schema View tab is available only after you select **FML32** in the Buffer Type drop-down list in the [Publisher Options Tab](#), specify the request header reference in the [Fetch Schema Tab](#), and click **Apply**.

You can configure the schema for the fields that occur multiple times. Different adapter service uses different schema. The schema is dependent on the Tuxedo communication paradigm(s) that are used

Click **Apply** to apply the changes before leaving this tab.

Table 17 Publication Service: Publisher Schema View Tab

Field	Description
Name	This column lists the names of the field or record. The values in this column come from the header file selected in the <a href="#">Fetch Schema Tab</a> .
Sequence Name	Enter the sequence name. This name groups fields that occur the same number of times. The sequence names can use ASCII alphanumeric characters. Additionally, the <code>_</code> , <code>%</code> , <code>@</code> , <code>'</code> , <code> </code> , <code>/</code> , <code>~</code> , <code>{</code> , <code>}</code> , <code>#</code> , <code>-</code> , <code>\$</code> non-alphanumeric characters are the acceptable.

## Schema Tab

The Schema tab is read only. Click the **Go to referenced resources..** icon to view the schema in the AESchemas folder.

Table 18 Publication Service: Schema Tab

Field	Description
Request Schema Reference	The request schema reference is updated automatically.

## Transport Tab

Message Transport options can be set for a publication service depending on the transport type selected in the [Configuration Tab](#).

Click **Apply** to apply the changes before leaving this tab.

Table 19 Publication Service: Transport Tab

Field	Description
When TIBCO Rendezvous is selected as the transport type, the following options are available.	
Message Subject	<p>Publisher subject. 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 that is different from the default in this field.</p> <p>See <i>TIBCO Rendezvous Concepts</i> for information about specifying subject names.</p>
Reply Message Subject	This field is disabled for Publication service.
Quality of Service	<p>Select the level of service that determines how messages are sent. See <a href="#">Quality of Service</a> for a description of these options.</p> <ul style="list-style-type: none"><li>Reliable</li><li>Certified</li></ul>
Wire Format	<p>The wire format in which data will be sent. See <a href="#">Wire Formats</a> for a description of these formats.</p> <ul style="list-style-type: none"><li>ActiveEnterprise Message</li><li>XML Message</li></ul>

Table 19 Publication Service: Transport Tab

Field	Description
Session Reference	When you create a service, TIBCO Designer creates a corresponding session resource in the Advanced > Sessions folder. The session resource displays in this field. If you have explicitly created a custom session of the same type, you can click the <b>Browse</b> icon to replace the autogenerated session. Changing the session for a service is not recommended.
Endpoint Reference	<p>It is an endpoint reference for the service. You do not need to edit this field. The endpoint points to the corresponding endpoint resource in the Advanced &gt; Sessions folder. The endpoint resource is automatically created by TIBCO Designer.</p> <p>Note that the adapter agent internally uses TIBCO Rendezvous with the Reliable quality of service to communicate with the adapter in the Adapter Agent. An Endpoint called <code>PubAgentSubscriber</code> is automatically created upon the creation of a Publication service. The subject name for this communication is configurable. No other attribute of this Endpoint should be manually edited.</p> <p>For a service managed endpoint, do not change the endpoint type.</p>
<b>When JMS is selected as the transport type, the following options are available.</b>	
Destination	<p>The publisher destination. A service uses a default destination generated using the <code>Domain</code> and <code>Deployment</code> global variables, the adapter acronym, the adapter instance name and the service name. If you use this default destination, make sure the values for <code>Domain</code> and <code>Deployment</code> are not empty.</p> <p>Alternatively, you can manually enter a destination in this field. The destination does not have to be predefined in the TIBCO Enterprise Message Service server. The destination can be static or dynamic.</p> <p>See <i>TIBCO Enterprise Message Service User's Guide</i> for information about destinations.</p>
Reply Destination	This field is disabled for Publication service.
Wire Format	<p>The wire format in which messages are to be published. See <a href="#">Wire Formats</a> for a description of the format.</p> <ul style="list-style-type: none"> <li>XML Message</li> </ul>
Connection Factory Type	<p>Select one of the following:</p> <ul style="list-style-type: none"> <li>Topic</li> <li>Queue</li> </ul> <p>See <a href="#">Connection Factory Type</a> for a description of connection factory type.</p>

Table 19 Publication Service: Transport Tab

Field	Description
Delivery Mode	<p>The delivery mode for each message sending operation. See <a href="#">Delivery Mode</a> for a description of each mode.</p> <ul style="list-style-type: none"><li>• Persistent</li><li>• Non-Persistent</li></ul>
Session Reference	<p>When you create a service, TIBCO Designer creates a corresponding session resource in the Advanced &gt; Sessions folder. The session resource displays in this field. If you have explicitly created a custom session of the same type, you can click the <b>Browse</b> icon to replace the autogenerated session. Changing the session for a service is not recommended.</p>
Endpoint Reference	<p>It is an endpoint reference for the service. You do not need to edit this field. The endpoint points to the corresponding endpoint resource in the Advanced &gt; Sessions folder. The endpoint resource is automatically created by TIBCO Designer.</p> <p>Note that the adapter agent internally uses TIBCO Rendezvous with the Reliable quality of service to communicate with the adapter in the Adapter Agent. An Endpoint called <code>PubAgentSubscriber</code> is automatically created upon the creation of a Publication service. The subject name for this communication is configurable. No other attribute of this Endpoint should be manually edited.</p> <p>For a service managed endpoint, do not change the endpoint type.</p>

# Subscription Service Tabs

The subscription service receives messages from the applications configured for the TIBCO environment and sends the messages to Tuxedo applications. If the Tuxedo communication paradigm is set to either Queue-based or Conversational, a reply is published on the reply subject.

You can configure the parameters for the subscription service using the following tabs:

- [Configuration Tab](#)
- [Subscriber Options Tab](#)
- [Fetch Schema Tab](#)
- [Subscriber Schema View Tab](#)
- [Reply Schema View Tab](#)
- [Schema Tab](#)
- [Transport Tab](#)

## Configuration Tab

You can specify a name and select the transport type for a subscription service in this tab. Click **Apply** to apply the changes before leaving this tab.

Table 20 Subscription Service: Configuration Tab

Field	Description
Name	<div>You can use the default name or replace it with a name of your choice.</div> <ul style="list-style-type: none"><li>• A service name must use alphanumeric characters. An underscore ( _ ) character can be used. The entire instance name must be less than 80 characters. The space character cannot be used in an instance name.</li><li>• A service name cannot use global variables.</li></ul>
Description	<div>Provide information about the publication service. This field is optional.</div>

Table 20 Subscription Service: Configuration Tab

Field	Description
Transport Type	<p>Select the transport type (JMS or TIBCO Rendezvous) to be used by the runtime adapter. This selection determines which options appear in the <a href="#">Transport Tab</a>.</p> <p>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.</p> <p>To enable and configure SSL, in the Project panel, expand the <b>Advanced</b> folder, then expand the <b>Sessions</b> folder. Select the TIBCO Rendezvous session or JMS topic and click <b>Use SSL?</b>. The SSL configuration options are explained in the online help associated with the session dialog. Click the question mark to display the online help.</p>

## Subscriber Options Tab

After you are done with the configuration, click **Apply** to apply the changes before leaving this tab.

Table 21 Subscription Service: Subscriber Options Tab

Field	Description
Buffer Type	<p>Three options are available:</p> <ul style="list-style-type: none"> <li> <b>FML32</b> <p>If the buffer type is FML32, the FML32 header files need to be selected in the <a href="#">Fetch Schema Tab</a>. The schema to be used is created based on the header file.</p> </li> <li> <b>String</b> <p>A class named <i>service name</i> Request is created under Schemas/classes/Tuxedo/adaptor instance name/service name. The class has a String data type attribute named in. See <a href="#">Appendix C, Message Formats</a> for more information.</p> <p>a class named <i>service name</i> Reply is created under Schemas/classes/Tuxedo/adaptor instance name/service name. The class has a String data type attribute named out, which is provided by the Subscription, Request-Response or Request-Response Invocation services.</p> </li> <li> <b>CArray</b> <p>A class named <i>service name</i> Request is created with a binary data type attribute named in. This attribute has an extended property named length. Refer to <a href="#">Appendix C, Message Formats</a> for more information.</p> <p>A class named <i>service name</i> Reply is created with a binary data type attribute named out, which is provided by the Subscription, Request-Response and Request-Response Invocation services (in addition to the default field ErrorCode which appears for Subscription and Request-Response). This attribute has an extended property named length with a value of 10.</p> <p>Note that the automatically created classes for CArray and String buffer types should not be manually edited.</p> </li> </ul>
Load Field Id Dynamically	<p>This checkbox appears only if you select <b>FML32</b> in the Buffer Type drop-down list. It specifies whether the Field IDs for the FML32 fields should be loaded dynamically at runtime. If unchecked, the Field ID values loaded from the header files are used.</p>
CArray Request Length	<p>This field appears only if you select <b>CArray</b> in the Buffer Type drop-down list. It is the length of the request CArray buffer type.</p>
CArray Reply Length	<p>This field appears only if you select <b>CArray</b> in the Buffer Type drop-down list. It is the length of the reply CArray buffer type.</p>

## Fetch Schema Tab

The Fetch Schema tab is available only if you select **FML32** in the Buffer Type drop-down list in the [Subscriber Options Tab](#). Click **Apply** to apply the changes before leaving this tab.

Table 22 Subscription Service: Fetch Schema Tab

Field	Description
Request Header Reference	<p>Browse and select the header file containing information about the fields to be included in the schema. The schema is created based on the FML32 header files that you have selected and is used for subscribing.</p> <p>The schema fields are created with names suffixed with <code>_field ID</code>. Irrespective of the type of the FML32 fields, the schema is always the string type. The classes that are created are stored in the <code>tibco/public/class/ae/Tuxedo</code> directory.</p>
Reply Header Reference	<p>Browse and select the header file containing information about the fields to be included in the reply schema. The schema is created based on the FML32 header files that you have selected and is used for subscribing.</p> <p>The schema fields are created with names suffixed with <code>_field ID</code>. Irrespective of the type of the FML32 fields, the schema that is created is always of the string type. The classes that are created are stored in the <code>tibco/public/class/ae/Tuxedo</code> directory.</p> <p>If you made any changes to the header files, click the <b>Refresh</b> button to load the new schema. The sequence names that you have specified in the <a href="#">Subscriber Schema View Tab</a> remain unchanged.</p>

## Subscriber Schema View Tab

The Subscriber Schema View tab is available only after you select **FML32** in the Buffer Type drop-down list in the [Subscriber Options Tab](#), specify the request header reference in the [Fetch Schema Tab](#), and click **Apply**.

You can configure the schema for the fields that occur multiple times. Different adapter service uses different schema. The schema is dependent on the Tuxedo communication paradigm(s) that are used

Note that sequences cannot be used with the Conversational communication paradigm.

Click **Apply** to apply the changes before leaving this tab.

Table 23 Subscription Service: Subscriber Schema View Tab

Field	Description
Name	<p>This column lists the names of the field or record. The values in this column come from the header file selected in the <a href="#">Fetch Schema Tab</a>.</p>



Table 23 Subscription Service: Subscriber Schema View Tab

Field	Description
Sequence Name	Enter the sequence name. This name groups fields that occur the same number of times. The sequence names can use the ASCII alphanumeric characters. Additionally, the <code>_</code> , <code>%</code> , <code>@</code> , <code>'</code> , <code> </code> , <code>/</code> , <code>~</code> , <code>{</code> , <code>}</code> , <code>#</code> , <code>-</code> , <code>\$</code> non-alphanumeric characters are the acceptable.

## Reply Schema View Tab

The Reply Schema View tab is available only after you select **FML32** in the Buffer Type drop-down list in the [Subscriber Options Tab](#).

This tab is similar to the [Subscriber Schema View Tab](#). You can set the sequence name for the fields that may occur more than once in a reply schema.





Sequences cannot be used with the Conversational communication paradigm.

## Schema Tab

The Schema tab is read only. The Request Schema Reference and Reply Schema Reference fields are updated automatically.

Table 24 Subscription Service: Schema Tab

Field	Description
Request Schema Reference	This is a reference to the subscription service request schema that can be found in the AESchemas folder. Click  to view the schema resource in the AESchemas folder.
Reply Schema Reference	This is a reference to the subscription service reply schema that can be found in the AESchemas folder. Click the  to view the schema resource in the AESchemas folder.

## Transport Tab

Message Transport options can be set for a subscription service depending on the transport type selected in the [Configuration Tab](#).

Click **Apply** to apply the changes before leaving this tab.

Table 25 Subscription Service: Transport Tab

Field	Description
When TIBCO Rendezvous is selected as the transport type, the following options are available.	

Table 25 Subscription Service: Transport Tab

Field	Description
Message Subject	<p>Subscriber subject. 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 that is different from the default in this field.</p> <p>See <i>TIBCO Rendezvous Concepts</i> for information about specifying subject names.</p>
Quality of Service	<p>Select the level of service that determines how messages are sent. See <a href="#">Quality of Service</a> for a description of these options.</p> <ul style="list-style-type: none"><li>Reliable</li><li>Certified</li><li>Distributed Queue</li></ul> <p>After selecting Distributed Queue, a session called <code>DefaultRVCMQSession</code> is created in the Project. You can modify the parameters of this default session if needed.</p> <p>An Endpoint called <code>SubscriptionServiceEndPoint</code> is automatically created upon the creation of the <code>DefaultRVCMQSession</code> session for the subscription service. No attributes of this endpoint need to be manually edited.</p> <p>Note that the Transactional quality of service is not supported</p>
Wire Format	<p>The wire format in which data will be sent. See <a href="#">Wire Formats</a> for a description of these formats.</p> <ul style="list-style-type: none"><li>ActiveEnterprise Message</li><li>XML Message</li></ul>
Session Reference	<p>When you create a service, TIBCO Designer creates a corresponding session resource in the Advanced &gt; Sessions folder. The session resource displays in this field. If you have explicitly created a custom session of the same type, you can click <b>Browse</b> to replace the autocreated session. Changing the session for a service is not recommended.</p>
Endpoint Reference	<p>It is an endpoint reference for the service. You do not need to edit this field. The endpoint points to the corresponding endpoint resource in the Advanced &gt; Sessions folder. The endpoint resource is automatically created by TIBCO Designer.</p> <p>Note that for a service managed endpoint, do not change the endpoint type.</p>
When JMS is selected as the transport type, the following options are available.	

Table 25 Subscription Service: Transport Tab

Field	Description
Destination	<p>The subscriber destination. A service uses a default destination generated using the Domain and Deployment global variables, the adapter acronym, the adapter instance name and the service name. If you use this default destination, make sure the values for Domain and Deployment are not empty.</p> <p>Alternatively, you can manually enter a destination in this field. The destination does not have to be predefined in the TIBCO Enterprise Message Service server. The destination can be static or dynamic.</p> <p>See <i>TIBCO Enterprise Message Service User's Guide</i> for information about destinations.</p>
Wire Format	<p>The wire format in which messages are to be subscribed. See <a href="#">Wire Formats</a> for a description of the format.</p> <ul style="list-style-type: none"> <li>XML Message</li> </ul>
Connection Factory Type	<p>Select one of the following:</p> <ul style="list-style-type: none"> <li>Topic</li> <li>Queue</li> </ul> <p>See <a href="#">Connection Factory Type</a> for a description of connection factory type.</p>
Delivery Mode	<p>This drop-down list is available only if you select <b>Topic</b> in the Connection Factory Type drop-down list.</p> <p>It is the delivery mode for each message sending operation. See <a href="#">Delivery Mode</a> for a description of each mode.</p> <ul style="list-style-type: none"> <li>Durable</li> <li>Non-Durable</li> </ul>
Session Reference	<p>When you create a service, TIBCO Designer creates a corresponding session resource in the Advanced &gt; Sessions folder. The session resource displays in this field. If you have explicitly created a custom session of the same type, you can click the <b>Browse</b> icon to replace the autogenerated session. Changing the session for a service is not recommended.</p>
Endpoint Reference	<p>It is an endpoint reference for the service. You do not need to edit this field. The endpoint points to the corresponding endpoint resource in the Advanced &gt; Sessions folder. The endpoint resource is automatically created by TIBCO Designer.</p> <p>For a service managed endpoint, do not change the endpoint type.</p>

# Request-Response Service Tabs

This service is also called a Request Reply Server or RPC (Remote Procedural Call) Server. When running as a request-response service, the adapter acts as a Tuxedo client and accesses Tuxedo services via ATMI. The adapter offers request-response functionality for tasks requiring a response by invoking the Tuxedo service.

You can configure parameters in the following tabs:

- [Configuration Tab](#)
- [RPC Server Options Tab](#)
- [Fetch Schema Tab](#)
- [Request Schema View Tab](#)
- [Reply Schema View Tab](#)
- [Schema Tab](#)
- [Transport Tab](#)

## Configuration Tab

You can specify a name and select the transport type for a request-response service in this tab. Click **Apply** to apply the changes before leaving this tab.

Table 26 Request-Response Service: Configuration Tab

Field	Description
Name	<div>You can use the default name or replace it with a name of your choice.</div> <ul style="list-style-type: none"><li>• A service name must use alphanumeric characters. An underscore ( <code>_</code> ) character can be used. The entire instance name must be less than 80 characters. The space character cannot be used in an instance name.</li><li>• A service name cannot use global variables.</li></ul>
Description	<div>Provide information about the publication service. This field is optional.</div>

*Table 26 Request-Response Service: Configuration Tab*

Field	Description
Transport Type	<p>Select the transport type (JMS or TIBCO Rendezvous) to be used by the runtime adapter. This selection determines which options appear in the <a href="#">Transport Tab</a>.</p> <p>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.</p> <p>To enable and configure SSL, in the Project panel, expand the <b>Advanced</b> folder, then expand the <b>Sessions</b> folder. Select the TIBCO Rendezvous session or JMS topic and click <b>Use SSL?</b>. The SSL configuration options are explained in the online help associated with the session dialog. Click the question mark to display the online help.</p>

## RPC Server Options Tab

After you are done with the configuration, click **Apply** to apply the changes before leaving this tab.

Table 27 Request-Response Service: RPC Server Options Tab

Field	Description
Buffer Type	<p>Three options are available:</p> <ul style="list-style-type: none"><li>• <b>FML32</b>  If the buffer type is FML32, the FML32 header files need to be selected in the <a href="#">Fetch Schema Tab</a>. The schema to be used is created based on the header file.</li><li>• <b>String</b>  A class named <i>service name</i> Request is created under Schemas/classes/Tuxedo/adapter instance name/service name. The class has a String data type attribute named in. See <a href="#">Appendix C, Message Formats</a> for more information.  a class named <i>service name</i> Reply is created under Schemas/classes/Tuxedo/adapter instance name/service name. The class has a String data type attribute named out, which is provided by the Subscription, Request-Response or Request-Response Invocation services.</li><li>• <b>CArray</b>  A class named <i>service name</i> Request is created with a binary data type attribute named in. This attribute has an extended property named length. Refer to <a href="#">Appendix C, Message Formats</a> for more information.  A class named <i>service name</i> Reply is created with a binary data type attribute named out, which is provided by the Subscription, Request-Response and Request-Response Invocation services (in addition to the default field ErrorCode which appears for Subscription and Request-Response). This attribute has an extended property named length with a value of 10.</li></ul> <p>Note that the automatically created classes for CArray and String buffer types should not be manually edited.</p>
Use Asynchronous Invocation	<p>Check this checkbox to use the asynchronous communication paradigm. The adapter accesses the Tuxedo service asynchronously when this checkbox is checked.</p> <p>Note that when using the asynchronous communication paradigm with the request-response service, the adapter does not support multithreading.</p>
Load Field Id Dynamically	<p>This checkbox appears only if you select <b>FML32</b> in the Buffer Type drop-down list. It specifies whether the Field IDs for the FML32 fields should be loaded dynamically at runtime. If unchecked, the Field ID values loaded from the header files are used.</p>

Table 27 Request-Response Service: RPC Server Options Tab

Field	Description
CArray Request Length	This field appears only if you select <b>CArray</b> in the Buffer Type drop-down list. It is the length of the request CArray buffer type.
CArray Reply Length	This field appears only if you select <b>CArray</b> in the Buffer Type drop-down list. It is the length of the reply CArray buffer type.

## Fetch Schema Tab

The Fetch Schema tab is available only if you select **FML32** in the Buffer Type drop-down list in the [RPC Server Options Tab](#). Click **Apply** to apply the changes before leaving this tab.

Table 28 Request-Response Service: Fetch Schema Tab

Field	Description
Request Header Reference	<p>Browse and select the header file containing information about the fields to be included in the schema. The schema is created based on the FML32 header files that you have selected and is used for the request-response service.</p> <p>The schema fields are created with names suffixed with <i>_field ID</i>. Irrespective of the type of the FML32 fields, the schema is always the string type. The classes that are created are stored in the <code>tibco/public/class/ae/Tuxedo</code> directory.</p>
Reply Header Reference	<p>Browse and select the header file containing information about the fields to be included in the reply schema. The schema is created based on the FML32 header files that you have selected and is used for the request-response service.</p> <p>The schema fields are created with names suffixed with <i>_field ID</i>. Irrespective of the type of the FML32 fields, the schema that is created is always the string type. The classes created are stored in the <code>tibco/public/class/ae/Tuxedo</code> directory.</p> <p>If you made any changes to the header files, click the <b>Refresh</b> button to load the new schema. The sequence names that you have specified in the <a href="#">Request Schema View Tab</a> and <a href="#">Reply Schema View Tab</a> remain unchanged.</p>

## Request Schema View Tab

This tab is available only after you select **FML32** in the Buffer Type drop-down list in the [RPC Server Options Tab](#), specify the request and reply header references in the [Fetch Schema Tab](#), and click **Apply**.

You can configure the schema for the fields that occur multiple times. Different adapter service uses different schema. The schema is dependent on the Tuxedo communication paradigm(s) that are used

Click **Apply** to apply the changes before leaving this tab.

Table 29 Request-Response Service: Request Schema View Tab

Field	Description
Name	This column lists the names of the field or record. The values in this column come from the header file selected in the <a href="#">Fetch Schema Tab</a> .
Sequence Name	Enter the sequence name. This name groups fields that occur the same number of times. The sequence names can use ASCII alphanumeric characters. Additionally, the <code>_</code> , <code>,</code> , <code>%</code> , <code>@</code> , <code>'</code> , <code>'</code> , <code> </code> , <code>/</code> , <code>~</code> , <code>{</code> , <code>'</code> , <code>}</code> , <code>#</code> , <code>-</code> , <code>\$</code> non-alphanumeric characters are the acceptable.

### Reply Schema View Tab



The Reply Schema View tab is available only after you select **FML32** in the Buffer Type drop-down list in the [RPC Server Options Tab](#), specify the request and reply header references in the [Fetch Schema Tab](#), and click **Apply**.

This tab is similar to the [Request Schema View Tab](#). You can set the sequence name for the fields that may occur more than once in a reply schema.

### Schema Tab

The Schema tab is read only. The Request Schema Reference and Reply Schema Reference fields are updated automatically.

Table 30 Request-Response Service: Schema Tab

Field	Description
Request Schema Reference	This is a reference to the request-response service request schema that can be found in the AESchemas folder. Click  to view the schema resource in the AESchemas folder.
Reply Schema Reference	This is a reference to the request-response service reply schema that can be found in the AESchemas folder. Click the  to view the schema resource in the AESchemas folder. Note that the RPC class that is automatically created when a Request-Response service is created should not be manually edited.

### Transport Tab

Message Transport options can be set for a request-response service depending on the transport type selected in the [Configuration Tab](#).

Click **Apply** to apply the changes before leaving this tab.



Table 31 Request-Response Service: Transport Tab

Field	Description
<b>When TIBCO Rendezvous is selected as the transport type, the following options are available.</b>	
Message Subject	<p>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 that is different from the default in this field.</p> <p>See <i>TIBCO Rendezvous Concepts</i> for information about specifying subject names.</p>
Quality of Service	<p>Select the level of service that determines how messages are sent. See <a href="#">Quality of Service</a> for a description of these options.</p> <ul style="list-style-type: none"> <li>• Reliable</li> <li>• Certified</li> <li>• Distributed Queue</li> </ul> <p>After selecting Distributed Queue, a session called <code>DefaultRVCMQSession</code> is created in the Project. You can modify the parameters of the default session if needed.</p> <p>An Endpoint called <code>RequestResponseServiceEndPoint</code> is automatically created upon the creation of the <code>DefaultRVCMQSession</code> session for the Request Response service. Do not manually modify any other attributes of this Endpoint.</p> <p>See <i>TIBCO Rendezvous Concepts</i> for details about RVCMQ.</p>
Wire Format	<p>The wire format in which data will be sent. See <a href="#">Wire Formats</a> for a description of these formats.</p> <ul style="list-style-type: none"> <li>• ActiveEnterprise Message</li> </ul>
Session Reference	<p>When you create a service, TIBCO Designer creates a corresponding session resource in the Advanced &gt; Sessions folder. The session resource displays in this field. If you have explicitly created a custom session of the same type, you can click <b>Browse</b> to replace the autocreated session. Changing the session for a service is not recommended.</p>
Endpoint Reference	<p>It is an endpoint reference for the service. You do not need to edit this field. The endpoint points to the corresponding endpoint resource in the Advanced &gt; Sessions folder. The endpoint resource is automatically created by TIBCO Designer.</p> <p>Note that for a service managed endpoint, do not change the endpoint type.</p>
<b>When JMS is selected as the transport type, the following options are available.</b>	

**Table 31 Request-Response Service: Transport Tab**

Field	Description
Destination	<p>A service uses a default destination generated using the Domain and Deployment global variables, the adapter acronym, the adapter instance name and the service name. If you use this default destination, make sure the values for Domain and Deployment are not empty. Alternatively, you can manually enter a destination in this field. The destination does not have to be predefined in the TIBCO Enterprise Message Service server. The destination can be static or dynamic.</p> <p>See <i>TIBCO Enterprise Message Service User's Guide</i> for information about destinations.</p>
Wire Format	<p>The wire format in which messages are to be subscribed. See <a href="#">Wire Formats</a> for a description of the format.</p> <ul style="list-style-type: none"> <li>XML Message</li> </ul>
Connection Factory Type	<p>Select one of the following:</p> <ul style="list-style-type: none"> <li>Topic</li> <li>Queue</li> </ul> <p>See <a href="#">Connection Factory Type</a> for a description of connection factory type.</p>
Delivery Mode	<p>This drop-down list is available only if you select <b>Topic</b> in the Connection Factory Type drop-down list.</p> <p>It is the delivery mode for each message sending operation. See <a href="#">Delivery Mode</a> for a description of each mode.</p> <ul style="list-style-type: none"> <li>Durable</li> <li>Non-Durable</li> </ul>
Session Reference	<p>When you create a service, TIBCO Designer creates a corresponding session resource in the Advanced &gt; Sessions folder. The session resource displays in this field. If you have explicitly created a custom session of the same type, you can click the <b>Browse</b> icon to replace the autogenerated session. Changing the session for a service is not recommended.</p>
Endpoint Reference	<p>It is an endpoint reference for the service. You do not need to edit this field. The endpoint points to the corresponding endpoint resource in the Advanced &gt; Sessions folder. The endpoint resource is automatically created by TIBCO Designer.</p> <p>For a service managed endpoint, do not change the endpoint type.</p>

## Request-Response Invocation Service Tabs

This service is also called RPC (Remote Procedural Call) Client. The adapter receives a request from the Tuxedo application and sends the request through the TIBCO environment. When a response is returned from the TIBCO environment, the adapter sends the response back to the Tuxedo application.

You can configure parameters in the following tabs:

- [Configuration Tab](#)
- [RPC Client Options Tab](#)
- [Fetch Schema Tab](#)
- [Request Schema View Tab](#)
- [Reply Schema View Tab](#)
- [Schema Tab](#)
- [Transport Tab](#)

### Configuration Tab

You can specify a name and select the transport type for a request-response service in this tab. Click **Apply** to apply the changes before leaving this tab.

*Table 32 Request-Response Invocation Service: Configuration Tab*

Field	Description
Name	<p>You can use the default name or replace it with a name of your choice.</p> <ul style="list-style-type: none"> <li>• A service name must use alphanumeric characters. An underscore (_) character can be used. The entire instance name must be less than 80 characters. The space character cannot be used in an instance name.</li> <li>• A service name cannot use global variables.</li> </ul>
Description	Provide information about the publication service. This field is optional.

Table 32 Request-Response Invocation Service: Configuration Tab

Field	Description
Transport Type	<p>Select the transport type (JMS or TIBCO Rendezvous) to be used by the runtime adapter. This selection determines which options appear in the <a href="#">Transport Tab</a>.</p> <p>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.</p> <p>To enable and configure SSL, in the Project panel, expand the <b>Advanced</b> folder, then expand the <b>Sessions</b> folder. Select the TIBCO Rendezvous session or JMS topic and click <b>Use SSL?</b>. The SSL configuration options are explained in the online help associated with the session dialog. Click the question mark to display the online help.</p>

## RPC Client Options Tab

After you are done with the configuration, click **Apply** to apply the changes before leaving this tab.

**Table 33** *Request-Response Invocation Service: RPC Client Options Tab*

Field	Description
Buffer Type	<p>Three options are available:</p> <ul style="list-style-type: none"> <li> <b>FML32</b> <p>If the buffer type is FML32, the FML32 header files need to be selected in the <a href="#">Fetch Schema Tab</a>. The schema to be used is created based on the header file.</p> </li> <li> <b>String</b> <p>A class named <i>service name</i> Request is created under Schemas/classes/Tuxedo/<i>adapter instance name</i>/<i>service name</i>. The class has a String data type attribute named in. See <a href="#">Appendix C, Message Formats</a> for more information.</p> <p>a class named <i>service name</i> Reply is created under Schemas/classes/Tuxedo/<i>adapter instance name</i>/<i>service name</i>. The class has a String data type attribute named out, which is provided by the Subscription, Request-Response or Request-Response Invocation services.</p> </li> <li> <b>CArray</b> <p>A class named <i>service name</i> Request is created with a binary data type attribute named in. This attribute has an extended property named length. Refer to <a href="#">Appendix C, Message Formats</a> for more information.</p> <p>A class named <i>service name</i> Reply is created with a binary data type attribute named out, which is provided by the Subscription, Request-Response and Request-Response Invocation services (in addition to the default field ErrorCode which appears for Subscription and Request-Response). This attribute has an extended property named length with a value of 10.</p> <p>Note that the automatically created classes for CArray and String buffer types should not be manually edited.</p> </li> </ul>

Table 33 Request-Response Invocation Service: RPC Client Options Tab

Field	Description
Agent Transport Type	<p>Select one of the following:</p> <ul style="list-style-type: none"><li>• Rendezvous</li></ul> <p>TIBCO Rendezvous will be used to transport messages.</p> <ul style="list-style-type: none"><li>• Distributed Queue</li></ul> <p>load balancing will be used. Multiple request-response invocation services will run in RVCMQ mode to process messages passed by the Rendezvous Adapter Agent.</p> <p>After selecting Distributed Queue, a session called <code>DefaultRVCMQSession</code> is created. You may need to modify the parameters of this default session if needed.</p> <p>An Endpoint called <code>ClientAgentRvCmqSubscriber</code> is automatically created upon the creation of the <code>DefaultRVCMQSession</code> session for this request-response invocation service. You can configure the subject name for this communication by clicking the <code>ClientAgentRvCmqSubscriber</code> Endpoint under Advanced &gt; Session &gt; <code>DefaultRVCMQSession</code> in the Project panel. Do not manually modify any other attributes of this Endpoint.</p> <p>See <i>TIBCO Rendezvous Concepts</i> for details about RVCMQ</p>
Load Field Id Dynamically	<p>This checkbox appears only if you select <b>FML32</b> in the Buffer Type drop-down list. It specifies whether the Field IDs for the FML32 fields should be loaded dynamically at runtime. If unchecked, the Field ID values loaded from the header files are used.</p>

Fetch Schema Tab

The Fetch Schema tab is available only if you select **FML32** in the Buffer Type drop-down list in the [RPC Client Options Tab](#). Click **Apply** to apply the changes before leaving this tab.

Table 34 Request-Response Invocation Service: Fetch Schema Tab

Field	Description
Request Header Reference	<p>Browse and select the header file containing information about the fields to be included in the schema. The schema is created based on the FML32 header files that you have selected and is used for the request-response invocation service.</p> <p>The schema fields are created with names suffixed with <code>_field ID</code>. Irrespective of the type of the FML32 fields, the schema is always the string type. The classes that are created are stored in the <code>tibco/public/class/ae/Tuxedo</code> directory.</p>

Table 34 Request-Response Invocation Service: Fetch Schema Tab

Field	Description
Reply Header Reference	<p>Browse and select the header file containing information about the fields to be included in the reply schema. The schema is created based on the FML32 header files that you have selected and is used for the request-response service.</p> <p>The schema fields are created with names suffixed with <code>_field ID</code>. Irrespective of the type of the FML32 fields, the schema that is created is always of the string type. The classes that are created are stored in the <code>tibco/public/class/ae/Tuxedo</code> directory.</p> <p>If you made any changes to the header files, click the <b>Refresh</b> button to load the new schema. The sequence names that you have specified in the <a href="#">Request Schema View Tab</a> and <a href="#">Reply Schema View Tab</a> remain unchanged.</p>

## Request Schema View Tab

This tab is available only after you select **FML32** in the Buffer Type drop-down list in the [RPC Client Options Tab](#), specify the request and reply header references in the [Fetch Schema Tab](#), and click **Apply**.

You can configure the schema for the fields that occur multiple times. Different adapter service uses different schema. The schema is dependent on the Tuxedo communication paradigm(s) that are used.

Click **Apply** to apply the changes before leaving this tab.

Table 35 Request-Response Service: Request Schema View Tab

Field	Description
Name	This column lists the names of the field or record. The values in this column come from the header file selected in the <a href="#">Fetch Schema Tab</a> .
Sequence Name	Enter the sequence name. This name groups fields that occur the same number of times. The sequence names can use ASCII alphanumeric characters. Additionally, the <code>_, %, @, ' ',  , /, ~, {, }, #, -, \$</code> non-alphanumeric characters are the acceptable.

## Reply Schema View Tab



This tab is available only after you select **FML32** in the Buffer Type drop-down list in the [RPC Client Options Tab](#), specify the request and reply header references in the [Fetch Schema Tab](#), and click **Apply**.

This tab is similar to the [Request Schema View Tab](#). You can set the sequence name for the fields that may occur more than once in a reply schema.

## Schema Tab

The Schema tab is read only. The Request Schema Reference and Reply Schema Reference fields are updated automatically.

Table 36 Request-Response Invocation Service: Schema Tab

Field	Description
Request Schema Reference	This is a reference to the request-response invocation service request schema that can be found in the AESchemas folder. Click  to view the schema resource in the AESchemas folder.
Reply Schema Reference	<p>This is a reference to the request-response invocation service reply schema that can be found in the AESchemas folder. Click the  to view the schema resource in the AESchemas folder.</p> <p>Note that the RPC class and the RPCClient_Closure class are automatically created when a request-response invocation service is created. Do not manually edit the classes.</p>

## Transport Tab

Message Transport options can be set for a request-response invocation service depending on the transport type selected in the [Configuration Tab](#).

Click **Apply** to apply the changes before leaving this tab.

Table 37 Request-Response Invocation Service: Transport Tab

Field	Description
When TIBCO Rendezvous is selected as the transport type, the following options are available.	
Message Subject	<p>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 that is different from the default in this field.</p> <p>See <i>TIBCO Rendezvous Concepts</i> for information about specifying subject names.</p>
Quality of Service	<p>Select the level of service that determines how messages are sent. See <a href="#">Quality of Service</a> for a description of these options.</p> <ul style="list-style-type: none"><li>Reliable</li><li>Certified</li></ul>
Wire Format	<p>The wire format in which data will be sent. See <a href="#">Wire Formats</a> for a description of these formats.</p> <ul style="list-style-type: none"><li>ActiveEnterprise Message</li></ul>



Table 37 Request-Response Invocation Service: Transport Tab

Field	Description
Session Reference	When you create a service, TIBCO Designer creates a corresponding session resource in the Advanced > Sessions folder. The session resource displays in this field. If you have explicitly created a custom session of the same type, you can click <b>Browse</b> to replace the autogenerated session. Changing the session for a service is not recommended.
Endpoint Reference	<p>It is an endpoint reference for the service. You do not need to edit this field. The endpoint points to the corresponding endpoint resource in the Advanced &gt; Sessions folder. The endpoint resource is automatically created by TIBCO Designer.</p> <p>The adapter agent internally uses TIBCO Rendezvous with Reliable Quality of Service to communicate with the adapter in the Adapter Agent. An Endpoint called <code>ClientAgentSubscriber</code> is automatically created upon the creation of a request-response invocation service. You can specify the subject name for the communication. No other attributes of this Endpoint should be manually edited.</p> <p>Note that for a service managed endpoint, do not change the endpoint type.</p>
<b>When JMS is selected as the transport type, the following options are available.</b>	
Destination	<p>A service uses a default destination generated using the <code>Domain</code> and <code>Deployment</code> global variables, the adapter acronym, the adapter instance name and the service name. If you use this default destination, make sure the values for <code>Domain</code> and <code>Deployment</code> are not empty.</p> <p>Alternatively, you can manually enter a destination in this field. The destination does not have to be predefined in the TIBCO Enterprise Message Service server. The destination can be static or dynamic.</p> <p>See <i>TIBCO Enterprise Message Service User's Guide</i> for information about destinations.</p>
Wire Format	<p>The wire format in which messages are to be subscribed. See <a href="#">Wire Formats</a> for a description of the format.</p> <ul style="list-style-type: none"> <li>XML Message</li> </ul>
Connection Factory Type	<p>Select one of the following:</p> <ul style="list-style-type: none"> <li>Topic</li> <li>Queue</li> </ul> <p>See <a href="#">Connection Factory Type</a> for a description of connection factory type.</p>
Delivery Mode	<p>It is the delivery mode for each message sending operation. See <a href="#">Delivery Mode</a> for a description of each mode.</p> <ul style="list-style-type: none"> <li>Persistent</li> <li>Non-Persistent</li> </ul>

Table 37 Request-Response Invocation Service: Transport Tab

Field	Description
Session Reference	<p>When you create a service, TIBCO Designer creates a corresponding session resource in the Advanced &gt; Sessions folder. The session resource displays in this field. If you have explicitly created a custom session of the same type, you can click the <b>Browse</b> icon to replace the autogenerated session. Changing the session for a service is not recommended.</p>
Endpoint Reference	<p>It is an endpoint reference for the service. You do not need to edit this field. The endpoint points to the corresponding endpoint resource in the Advanced &gt; Sessions folder. The endpoint resource is automatically created by TIBCO Designer.</p> <p>The adapter agent internally uses TIBCO Rendezvous with Reliable Quality of Service to communicate with the adapter in the Adapter Agent. An Endpoint called <code>ClientAgentSubscriber</code> is automatically created upon the creation of a request-response invocation service. You can specify the subject name for the communication. No other attributes of this Endpoint should be manually edited.</p> <p>Note that for a service managed endpoint, do not change the endpoint type.</p>

## Chapter 5

# Deploying and Starting an Adapter Using TIBCO Administrator

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

## Topics

---

- [\*Creating an EAR File in TIBCO Designer, page 74\*](#)
- [\*Deploying the Project, page 75\*](#)
- [\*Starting or Stopping the Adapter, page 76\*](#)
- [\*Monitoring the Adapter, page 77\*](#)

## Creating an EAR File in TIBCO Designer

---

An Enterprise Archive file (EAR) contains the information about the adapter that you want to deploy. This could be one or more adapters, or 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 for an adapter instance.
2. Drag and drop the **Enterprise Archive** icon from the palette panel to the design panel.

If there are any configured adapter services in your project, an Adapter Archive resource becomes available in the palette panel.

3. Select the Enterprise Archive created in [step 2](#) and drag and drop the **Adapter Archive** icon from the Palettes panel to the Design Panel.
4. Configure the adapter archive in the Configuration pane and then click **Apply**.
5. Select the Enterprise Archive created in [step 2](#) and click **Build Archive** to create the archive file.

### See Also

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

## 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) can be added to the administration domain. The TIBCO software installed on the machine is then visible and accessible via the TIBCO Administrator GUI.

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

To deploy a project:

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

### See Also

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

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

## Starting or Stopping the Adapter

---

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

### Starting the Adapter

To start the adapter from the module:

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

The status changes from `Stopped` to `Starting` up to `Started`.

### Stopping the Adapter

To stop the adapter service, click the **Stop Selected** button.

### See Also

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





## Topics

---

- [Defining a TIBCO Hawk Session, page 80](#)
- [Using Global Variables, page 81](#)
- [Configuring a Remote Adapter, page 86](#)
- [Configuring Multiple Adapter Instances to use Distributed Queues to Process Inbound Messages, page 87](#)
- [Configuring Multiple Adapter Instances to use Distributed Queues to Process Outbound Messages, page 89](#)

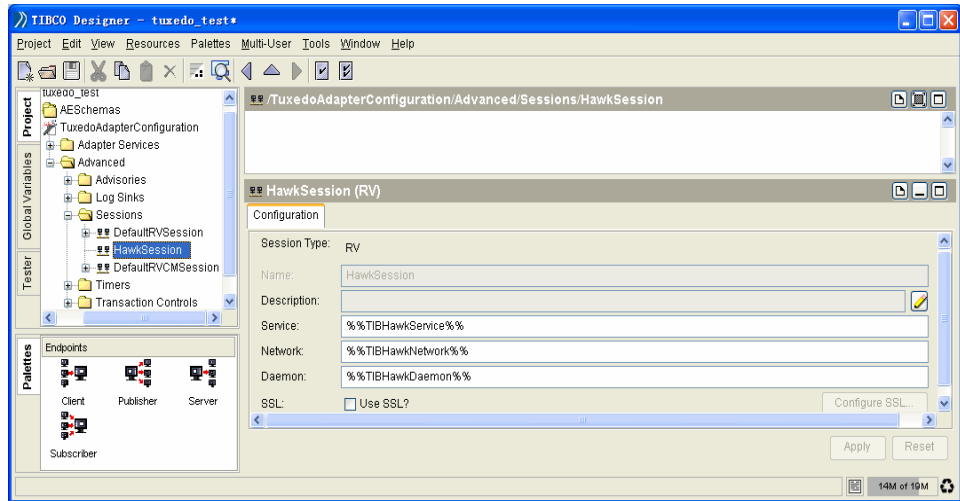
## Defining a TIBCO Hawk Session

A default TIBCO Rendezvous session, `HawkSession`, is defined in the project whenever a new instance of an adapter is configured. You can use this session to monitor the adapter using TIBCO Hawk. Use the following steps to modify the parameters of this default session.

1. In the project tree panel, click the **Tuxedo Adapter Configuration** icon defined for your adapter instance.
2. Select the **Advanced>Sessions>HawkSession**.

**Figure 3** shows the configuration information for the predefined Hawk session. You can modify the default values.

*Figure 3 Defining a TIBCO Hawk Session*



## 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](#).

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.

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

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

The project panel is updated to display all currently defined global variables. Click Open Advanced Editor (pencil icon at the top left corner). 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 the leftmost icon. Specify the name of the group, then press Enter. With the group icon selected, you can click the abc icon to add variables to the group.
- To add a global variable, click the abc icon. A new global variable item is added to the bottom of the list. Supply the variable name and, optionally, the value. Press Enter when you're done.

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

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

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

A number of global variables are predefined. See [Predefined Global Variables](#) for information. You may add definitions of any variables you need to the predefined variables.

## Changing Global Variable Values at Runtime

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

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

```
-tibco.clientVar.variablePathAndName value
```

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

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

## Predefined Global Variables

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

*Table 38 Predefined Global Variables*

Variable	Description
Deployment	Defaults to the TIBCO Designer project name. This global variable is used by the system to partially define the subject name defined for a service.
DirLedger	Specifies the path name of the TIBCO Rendezvous certified messaging ledger file. The default is the root installation directory.
DirTrace	Specifies the path name for log file used by the adapter. The default is the root installation directory.
Domain	The default value for file-based local projects is <code>MyDomain</code> . The value for server-based projects is the domain to which the project was saved.
HawkEnabled	Indicates whether TIBCO Hawk is used to monitor the adapter. True indicates that a TIBCO Hawk microagent is defined for the adapter. False indicates the microagent is not to be used.
JmsProviderUrl	Tells applications where the JMS daemon is located. Setting this value mostly makes sense in early stages of a project, when only one JMS daemon is used.
JmsSslProviderUrl	Specifies where the JMS SSL daemon is located.
RemoteRvDaemon	TIBCO Rendezvous routing daemon ( <code>rvrd</code> ) to be used. See <i>TIBCO Administrator Server Configuration Guide</i> for details about setting up a domain using <code>rvrd</code> .
RvDaemon	TIBCO Rendezvous daemon. Sessions use this daemon to establish communication. The default value is 7500.
RvNetwork	TIBCO Rendezvous network. This variable need only be set on computers with more than one network interface. If specified, the TIBCO Rendezvous daemon uses that network for all outbound messages.  In most cases, you can leave the default.

Table 38 Predefined Global Variables

Variable	Description
RvService	<p>TIBCO Rendezvous service. The Rendezvous daemon divides the network into logical partitions. Each transport communicates on a single service. A transport can communicate only on the same service with other transports.</p> <p>Unless you are using a non-default TIBCO Rendezvous configuration, you should leave the default (7500).</p>
RvaHost	<p>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 <i>TIBCO Rendezvous Administration</i> for details about specifying the rva parameters.</p>
RvaPort	<p>TCP port where the TIBCO Rendezvous agent (rva) listens for client connection requests. See <i>TIBCO Rendezvous Administration</i> for details about specifying the rva parameters. Defaults to 7501.</p>
TIBHawkDaemon	<p>TIBCO Rendezvous daemon used in the TIBCO Hawk session. See the <i>TIBCO Hawk Installation and Configuration</i> manual for details about this parameter.</p>
TIBHawkNetwork	<p>TIBCO Rendezvous network used by the TIBCO Hawk session. See the <i>TIBCO Hawk Installation and Configuration</i> manual for details about this parameter.</p>
TIBHawkService	<p>TIBCO Rendezvous service used by the TIBCO Hawk session. See the <i>TIBCO Hawk Installation and Configuration</i> manual for details about this parameter.</p>
adtuxedo.connection.clientName	<p>Used by the system to identify the Tuxedo application the adapter needs to connect to.</p>
adtuxedo.connection.password	<p>Password to be used by the system to log into the Tuxedo application.</p>
adtuxedo.connection.flags	<p>The flags that indicate client-specific notification mechanism and the mode of system access. The valid values are 1, 2, 4, 8, 10, 20, 40 and 47.</p>
adtuxedo.connection.groupName	<p>When the adapter is used as a workstation client, the value for this variable should be set to null, which is the default. When the adapter is used as a native client, specify a valid Tuxedo group name for this variable.</p>

*Table 38 Predefined Global Variables*

Variable	Description
<code>adtuxedo.connection.userName</code>	User name or User ID to be used by the adapter to log into the Tuxedo application.
<code>adtuxedo.connection.userPassword</code>	Variable length data that is forwarded to the application defined authentication service.
<code>numOfMaxPendingInvns</code>	This parameter is used to specify the Maximum number of Pending Invocations. The default value is 50.
<code>pollIntervalForOutstandingInvn</code>	This parameter determines how often the Request-Response service polls Tuxedo in an asynchronous mode for the outstanding invocations. The default setting is 2000 milliseconds.
<code>pollIntervalForReplies</code>	This parameter determines how often the Request-Response service polls for a reply from Tuxedo in an asynchronous mode. The default value is 1000 milliseconds.
<code>pollingInterval</code>	This parameter determines how often the Event-based Publication service polls for a connection between the adapter and the Tuxedo server. The default value is 3000 milliseconds.

## Configuring a Remote Adapter

---

If only TIBCO Designer is installed on your computer but the adapter is installed on a remote computer, you can still configure the adapter from your computer, as long as the remote computer has the required Tuxedo header file(s) and the palette files.

To do this:

1. Install only the design-time component of the adapter installer.
2. Copy all required Tuxedo header (.h) files to the system where TIBCO Designer is installed.
3. Open the project in TIBCO Designer. In the project tree panel, select the **Tuxedo Adapter Configuration** icon and configure the adapter. This icon represents the remote adapter configuration.



## Configuring Multiple Adapter Instances to use Distributed Queues to Process Inbound Messages

---

The distributed queue mode allows multiple adapters to distribute the load among the distributed queue members. The subscription or request response services provide process level load balancing for inbound messages.

You can configure multiple adapter instances to use distributed queues, then create a TIBCO ActiveMatrix BusinessWorks process to test them and use TIBCO Administrator to deploy multiple instances of the adapter. The instances can be deployed on the same machine or multiple machines.

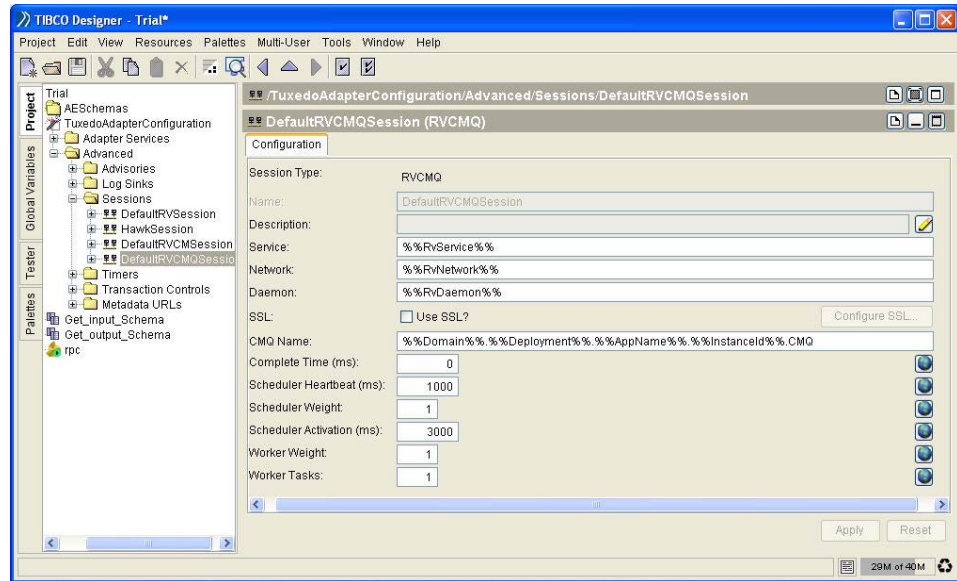
### Example Scenario

In this scenario, several adapter instances with a subscriber service are configured to use a distributed queue. To do this, follow these steps:

1. Start TIBCO Designer and create a new project.
2. Drag and drop the **TuxedoAdapterConfiguration** icon to the Design panel and configure the adapter instance.
3. Drag and drop the **SubscriptionService** icon from the Palette panel to the Design panel.
4. Under the Transport tab, in the Quality of Service drop-down list, select **Distributed Queue**. See [Transport Tab](#) for detail.
5. Under the Fetch Schema tab, browse and select the header file. See [Fetch Schema Tab](#) for detail.
6. In the project tree panel, click the Tuxedo Adapter Configuration icon defined for your adapter instance.
7. Select **Advanced>Sessions>DefaultRVCMQSession**.
8. You may need to change **CMQ Name**, **Scheduler Weight**, and **Worker Weight**.
  - **CMQ Name** The multiple adapter instances that you are using must have the same CMQ Name.
  - **Scheduler Weight** Acceptable values range from 1 to 65545. Default is 1.
  - **Worker Weight** The queue member with the greatest Worker Weight will process the heaviest message.

See *TIBCO Designer Palette Reference* for detail.

Figure 4 Defining an RVCMQ Session



9. Repeat step 2 through step 8 to create several adapter instances.
10. Create a TIBCO ActiveMatrix BusinessWorks process, configure it, and save the project. See *TIBCO ActiveMatrix BusinessWorks Process Design Guide* for detail about how to design a process.
11. Prepare Tuxedo, set the environment variables, and start the Tuxedo services.
12. Start the adapter instances and use the TIBCO ActiveMatrix BusinessWorks to send request messages to the Adapter.

After testing the process and creating the EAR file, you are ready to import it into TIBCO Administrator Enterprise Edition and then deploy it. See *TIBCO Administrator User's Guide* for detail.

## Configuring Multiple Adapter Instances to use Distributed Queues to Process Outbound Messages

---

The distributed queue mode allows multiple adapters to distribute the load among the distributed queue members. The agent-based publication and request-response invocation services provide process level load balancing for outbound messages.

You can configure multiple adapter instances to use distributed queues, then create a process to test them and use TIBCO Administrator to deploy multiple instances of the adapter. The instances can be deployed on the same machine, or multiple machines.

### Example Scenario

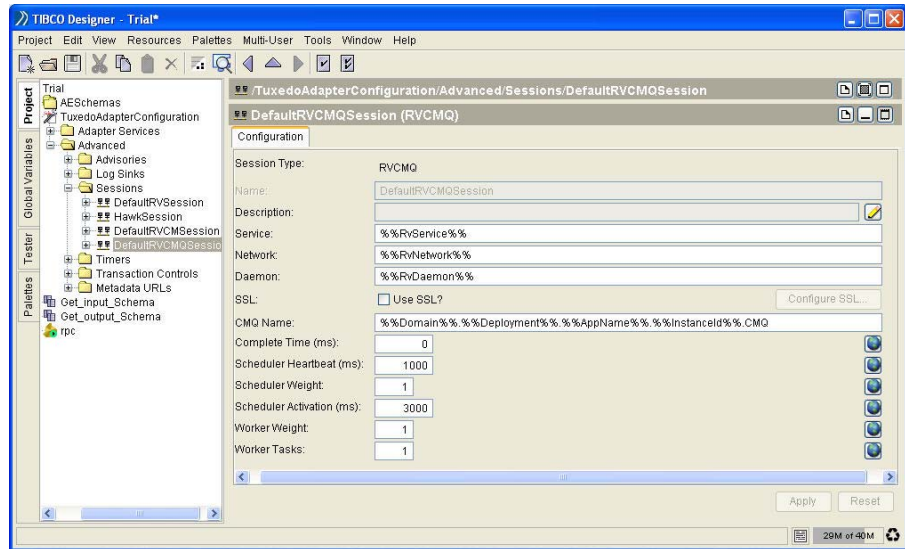
In this scenario, several adapter instances with Publication service is configured to use distributed queues. To do this, follow these steps:

1. Start TIBCO Designer and create a new project.
2. Drag and drop the **TuxedoAdapterConfiguration** icon to the Design panel and configure the adapter instance.
3. Drag and drop the **PublicationService** icon to the Design panel and make sure that the service name of the Publication service is same as the `serviceName` parameter of the TIBCO Rendezvous Adapter Agent Method that you are using.
4. Under the Publisher Options tab, select **Distributed Queue** in the Agent Transport Type drop-down list. See [Publisher Options Tab](#) for detail.
5. Under the Fetch Schema tab, browse and select the header file. See [Fetch Schema Tab](#) for detail.
6. In the project tree panel, click the Tuxedo Adapter Configuration icon defined for your adapter instance.
7. Select **Advanced>Sessions>DefaultRVCMQSession**.

8. You may need to change **CMQ Name**, **Scheduler Weight**, and **Worker Weight**.
  - **CMQ Name** The adapter instances that you are using must have the same CMQ Name.
  - **Scheduler Weight** Acceptable values range from 1 to 65545. Default is 1.
  - **Worker Weight** The queue member with the greatest Worker Weight will process the heaviest message.

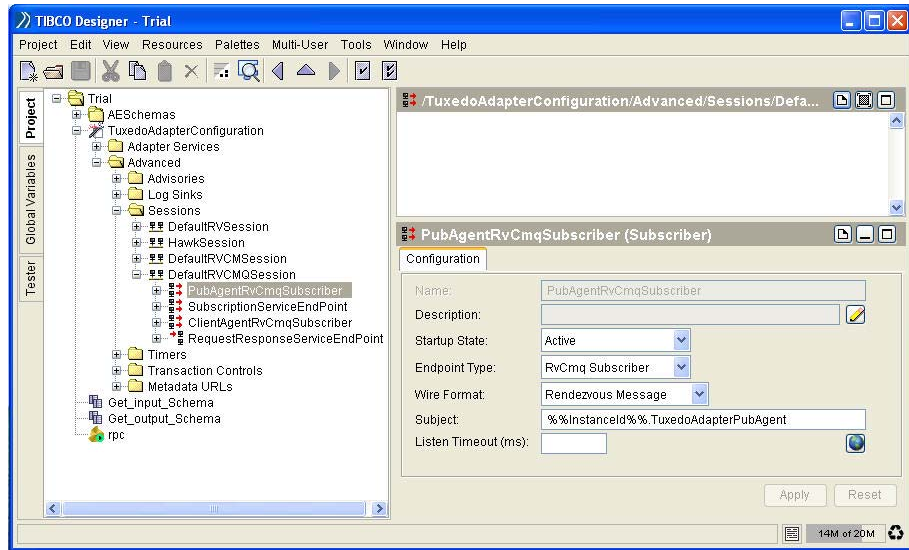
See *TIBCO Designer Palette Reference* for detail.

Figure 5 Defining an RVCMQ Session



9. Select **PubAgentRvCmqSubscriber** under **DefaultRVCMQSession**, the subject name you are using in the Subject field should be same as the subject parameter of the TIBCO Rendezvous Adapter Agent Method that you are using. All the adapter instances that you are using should have the same subject name. No other attribute of this Endpoint should be manually edited.

**Figure 6** *Defining PubAgentRvCmqSubscriber*



10. Repeat step 2 through step 9 to create several adapter instances.
11. Create a TIBCO ActiveMatrix BusinessWorks process, configure it, and save the project. See *TIBCO ActiveMatrix BusinessWorks Process Design Guide* for detail about how to design a process.
12. Prepare Tuxedo, set the environment variables, and start the Tuxedo services.
13. Start the adapter instances and use the Tuxedo services to send request messages to the Adapter through agent.

After testing the process and creating the EAR file, you are ready to import it into TIBCO Administrator and then deploy it. See *TIBCO Administrator User's Guide* for detail.



# Monitoring the Adapter Using TIBCO Hawk

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

## Topics

---

- [Overview, page 94](#)
- [Starting TIBCO Hawk Software, page 95](#)
- [The Auto-Discovery Process, page 96](#)
- [Invoking Microagent Methods, page 97](#)
- [Available Microagents, page 99](#)

## Overview

---

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

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

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

The TIBCO Hawk Enterprise Monitor consists of these components:

- **Display**—GUI front end that displays alarms and provides editors to create rule bases, create tests, view messages, and invoke microagents to request information or initiate an action.
- **Agents**—Intelligent processes that perform monitoring and take actions as defined in rules.
- **Rulebases**—Rules that are loaded by agents to determine agent behavior.
- **Application Management Interface (AMI)**—Manages network applications via TIBCO Rendezvous and supports communication between a network application and monitoring TIBCO Hawk agents, including the ability to examine application variables, invoke methods, and monitor system performance.
- **Microagents**—Feed information back to TIBCO Hawk and expose action methods to rulebases.

For more information, see the TIBCO Hawk documentation.



## Starting TIBCO Hawk Software

---

The TIBCO Hawk agent can be configured to start automatically during the system boot cycle. See the *TIBCO Hawk Installation and Configuration* guide for information about starting TIBCO Hawk.

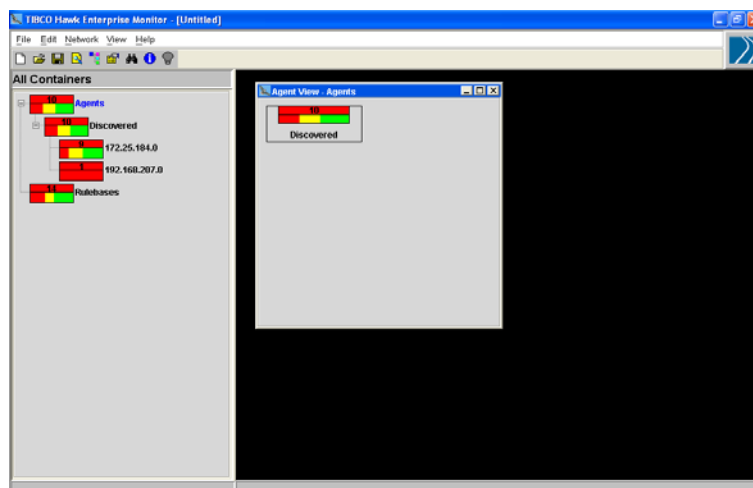
The *TIBCO Hawk Administrator's Guide* explains how to start the TIBCO Hawk Display.

## The Auto-Discovery Process

After you start an instance of TIBCO Hawk Display, it continues to search for 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.

Figure 7 TIBCO Hawk Enterprise Monitor



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

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

## Invoking Microagent Methods

A set of default microagents is loaded when a TIBCO Hawk Agent is started. When you install and start the adapter, its microagents are dynamically added to the local agent.

To invoke a microagent method:

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

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

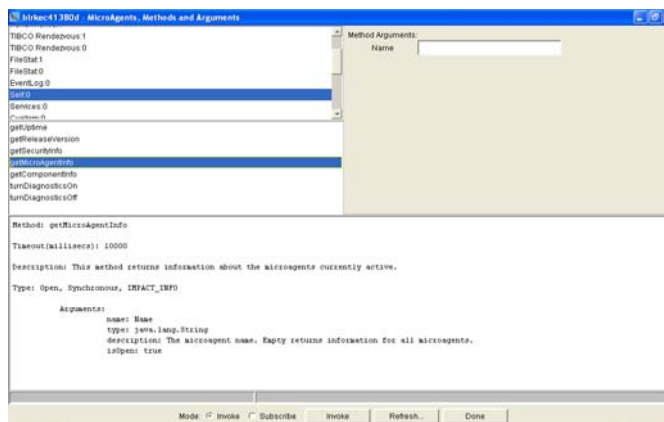
The Microagents, Methods, and Arguments dialog is displayed. The panel on the upper left lists microagents you can access on the current agent as shown in [Figure 8](#).

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

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

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

*Figure 8 Microagents, Methods and Arguments Dialog*



- 4. Specify any arguments for the method invocation.
- 5. Verify that the Invoke radio button is selected.
- 6. Click the **Invoke** button to invoke the selected method.

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

Figure 9 Invocation Result Dialog

Name	Display Name	Count	Help
COM TIBCO.hawk.hma.EventLog	EventLog	2	TIBCO Hawk Windows Event Log Microagent
COM TIBCO.hawk.hma.Process	Process	2	TIBCO Hawk Process Microagent
COM TIBCO.hawk.hma.FileStat	FileStat	2	TIBCO Hawk File Status Microagent
COM TIBCO.hawk.hma.Tib Rendezvous	TIBCO Rendezvous	2	TIBCO Hawk Rendezvous Microagent
COM TIBCO.hawk.hma.Services	Services	2	TIBCO Hawk Windows Services Microagent
COM TIBCO.hawk.microagent.Self	Self	1	TIBCO Hawk built-in Microagent
IM Engine - AE Tracing Hawk Sink for imed_	IM Engine - AE Tracing Hawk Sink for imed_	1	Provides trace messages from IM Engine
COM TIBCO.hawk.microagent.Custom	Custom	1	TIBCO Hawk built-in Microagent
IM Engine Administration for imed_debug_e	IM Engine Administration for imed_debug_e	1	HAWK agent for the shell
COM TIBCO.hawk.hma.Registry	Registry	2	TIBCO Hawk Windows Registry Microagent
COM TIBCO.hawk.hma.Performance	Performance	2	TIBCO Hawk Windows Performance Microa...
IM Engine Monitor for imed_debug_engine1	IM Engine Monitor for imed_debug_engine1	1	Various Engine monitoring functions
COM TIBCO.hawk.microagent.Logfile	Logfile	1	TIBCO Hawk built-in Microagent
COM TIBCO.hawk.microagent.RuleBaseEn	RuleBaseEngine	1	TIBCO Hawk built-in Microagent
COM TIBCO.hawk.microagent.SysInfo	SysInfo	1	TIBCO Hawk built-in Microagent

Click a cell to display its value in this area!

Done

- 7. Click **Done** to close the dialog.

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

## Available Microagents

Each adapter has three microagents, a standard TIBCO Hawk microagent named `COM.TIBCO.ADAPTER.xyz` where `xyz` is the adapter configuration name, a custom microagent and a class microagent. The microagents provide:

- Business level statistics—statistics that report the progress of the adapter as it interacts with the vendor application. These statistics indicate whether the Tuxedo service returned a success or a failure.
- Queries that return information about the state of the adapter. This can be an important tool for seeing the internals of an adapter and debugging it if something appears wrong. For example, methods can return information about threads, internal queues, or connections to the target system. Using these methods, one might be able to identify certain bottlenecks or gauge how successfully an adapter is scaling with respect to the current environment.
- Updates of the adapter runtime parameters. This includes retrieving the current runtime parameters and setting new runtime parameters without restarting the adapter. An example of this is getting and setting the polling interval. Updating a runtime parameter through the Hawk microagent only affects the setting of the instance that is running. It does not make a permanent change of the setting in either the repository or the `.tra` file.

By default, all microagents are available at runtime.



Custom microagents are deprecated in this release. All business statistics related methods which were part of the custom microagent are now available in the class microagent.

`perfMon` property value set in the adapter's property file affects the business statistics related methods. If this property is set to `on`, the adapter does all the performance related calculations. When you invoke the methods, if the `perfMon` property is set to `off`, default values are displayed and not the valid values.



The `getActivityStatisticsbyService` and `getActivityStatisticsbyOperation` methods display the following message: Method Invocation returned NO Data, if the `perfMon` property is set to `off`.

You can disallow adding custom methods to the class microagent when deploying the adapter by changing the `addCustomHawkMethodstoClassMAgent` property value in the adapter's property file. If this property is set to `on`, custom methods (`getPollingInterval` and `setPollingInterval`) are added to the class microagents. If set to `off`, these methods are not visible.

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

*Table 39 Standard Microagent Methods*

Method	Description	Page
<a href="#">_onUnsolicitedMsg()</a>	Displays alert messages sent to the current adapter.	<a href="#">103</a>
<a href="#">activateTraceRole()</a>	Activates a mapping of a role to a sink at runtime.	<a href="#">104</a>
<a href="#">deactivateTraceRole()</a>	Deactivates a mapping of a role to sinks at runtime.	<a href="#">105</a>
<a href="#">getAdapterServiceInformation()</a>	Returns information about the services implemented by this adapter.	<a href="#">112</a>
<a href="#">getComponents()</a>	Returns information about the publisher, subscriber and IODescriptor.	<a href="#">113</a>
<a href="#">getConfig()</a>	Returns basic configuration information. More specific information is accessed by the more specific methods.	<a href="#">114</a>
<a href="#">getConfigProperties()</a>	Returns all attributes and elements for the given repository object.	<a href="#">115</a>
<a href="#">getHostInformation()</a>	Returns standard and extended application information.	<a href="#">117</a>
<a href="#">getRvConfig()</a>	Returns information about all TIBCO Rendezvous sessions defined.	<a href="#">121</a>
<a href="#">getStatus()</a>	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.	<a href="#">122</a>
<a href="#">getTraceSinks()</a>	Returns information about sinks to which traces currently go.	<a href="#">124</a>
<a href="#">getVersion()</a>	Returns the configuration ID, application name, version, and date for this adapter instance.	<a href="#">125</a>
<a href="#">preRegisterListener()</a>	Preregisters an anticipated listener.	<a href="#">126</a>
<a href="#">resetActivityStatistics()</a>	Resets all the counts for the activity statistics.	<a href="#">127</a>
<a href="#">resetConnectionStatistics()</a>	Resets all the counts for the connection statistics.	<a href="#">128</a>
<a href="#">resetThreadStatistics()</a>	Resets all the counts for the thread statistics.	<a href="#">129</a>

*Table 39 Standard Microagent Methods*

Method	Description	Page
<a href="#">reviewLedger()</a>	Returns information retrieved from the ledger file of a certified messaging session for a publisher adapter.	<a href="#">130</a>
<a href="#">setTraceSinks()</a>	Adds a role or changes the file limit of a previously specified sink.	<a href="#">132</a>
<a href="#">stopApplicationInstance() )</a>	Stops the running adapter instance.	<a href="#">133</a>
<a href="#">unRegisterListener()</a>	Unregisters a currently preregistered listener.	<a href="#">134</a>

*Table 40 Class Microagent Methods*

Method	Description	Page
<a href="#">getActivityStatistics()</a>	Returns the total number of objects processed for all the services or operations.	<a href="#">106</a>
<a href="#">getActivityStatisticsByOperation()</a>	Returns the total number of objects processed for all the schemas by each service that is associated with a specified operation.	<a href="#">107</a>
<a href="#">getActivityStatisticsByService()</a>	Returns information about the services implemented by this adapter.	<a href="#">109</a>
<a href="#">getActivityStatisticsBySchema()</a>	Returns the total number of objects processed for the given schema by each service that uses the schema.	<a href="#">111</a>
<a href="#">getConnectionStatistics() )</a>	Returns the state and statistics for all the current connections used by the adapter.	<a href="#">116</a>
<a href="#">getPerfMonSetting()</a>	Returns the setting of the perfMon option.	<a href="#">118</a>
<a href="#">getPollingInterval()</a>	Returns the current polling interval setting.	<a href="#">119</a>
<a href="#">getQueueStatistics()</a>	Returns the current count of elements in any internal queue used by the adapter.	<a href="#">120</a>
<a href="#">getThreadStatistics()</a>	Returns the operation counts of the current threads	<a href="#">123</a>
<a href="#">resetActivityStatistics() )</a>	Resets all the counts for the activity statistics.	<a href="#">127</a>
<a href="#">resetConnectionStatistics() )</a>	Resets all the counts for the connection statistics.	<a href="#">128</a>

Table 40 Class Microagent Methods

Method	Description	Page
<code>resetThreadStatistics()</code>	Resets all the counts for the thread statistics.	129
<code>setPollingInterval()</code>	Sets the polling interval for the publication service.	131



## **\_onUnsolicitedMsg()**

---

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

# activateTraceRole()

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

Parameters	Type	Description
Role Name	string	Name of the role to activate.
Sink Name	string	Name of the sink for which to activate the role.

## deactivateTraceRole()

---

Deactivates a mapping of a role to sinks at runtime.

Parameters	Type	Description
Role Name	string	Name of the role to activate.
Sink Name	string	Name of the sink for which to activate the role.

# getActivityStatistics()

Returns the total number of objects processed for all the services or operations, based on the request type. Also, returns the number of success and error objects.

Parameter	Type	Description
Type	integer	Statistics categorized by service or by operation.

Returns	Type	Description
Name	string	Name of the service or operation.
Total	integer	Total number of objects processed for this service or operation.
Success	integer	The number of objects that were successfully identified and processed for this service or operation.
Failure	integer	The number of objects that were identified for this service or operation but for which processing failed.
MeasurementInterval	integer	Displays the time (in seconds) since last time the adapter was reset, or if never reset, since the adapter started.



If statistics are requested for, based on operation and no activity has been performed, or the `perfmon` property is set to off, the following is displayed, Method Invocation returned NO Data



The statistics provided by the `Failure` column take into account only application level invocation errors such as connection errors, incorrect Tuxedo service name, business logic error etc. Incorrect data supplied to the adapter (wrong communication type, unspecified Tuxedo service name) is not considered.

## getActivityStatisticsByOperation()

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. If no value is provided for the input parameter `Operation`, then statistics for all operations of this instance will be displayed.

Parameter	Type	Description
Operation	string	<p>The operation is a combination of Tuxedo Service name and the Communication Type. The Tuxedo Service is the name of the service that is invoked by adapter Subscription and Request-Response services.</p> <p>In the case of Event-based Subscription or Publication services, it is the EventName on which data is being posted or subscribed to respectively. In the case of Agent based Publication or Request-Response Invocation services, it is not applicable (NA is displayed). The Communication type is the Tuxedo communication paradigm that was used in the transaction. The possible values for Communication Type are Async, Sync, Queue, Event, Conv1, ClientAgent and PubAgent.</p>

Returns	Type	Description
Service Name	string	Name of the service that is associated with the specified operation.
Total	integer	Total number of objects processed for this operation.
Success	integer	The number of objects that were successfully identified for this operation and processed.
Failure	integer	The number of objects that were identified for this operation but for which processing failed.
LineIndex	string	Used as an index for the method. Its value is a concatenated string of the Operation and ServiceName parameters separated by a comma.
MeasurementInterval	integer	Displays the time (in seconds) since last time the adapter was reset, or if never reset, since the adapter started.



If no activity has been performed or the `perfMon` property is set to `off` the following message is displayed:

Method Invocation returned NO Data



The statistics provided by the `Failure` column take into account only application level invocation errors such as connection errors, incorrect Tuxedo service name, business logic error etc. Incorrect data supplied to the adapter (wrong communication type, unspecified Tuxedo service name) is not considered.

## getActivityStatisticsByService()

Returns the total number of objects processed for each of the services. Also, returns the number of success and error objects. If no value is provided for the input parameter `ServiceName`, then statistics for all services of this instance will be displayed. A final sum of all services or operations should be given at the end, in the display.

Parameter	Type	Description
Service Name	string	Name of the service.

Returns	Type	Description
Service Name	string	Name of the service.
Schema Name	string	Name of the schema that is associated with the service.
Operation	string	<p>The operation is a combination of Tuxedo Service name and the Communication Type. The Tuxedo Service is the name of the service that is invoked by adapter Subscription and Request-Response services.</p> <p>In the case of Event-based Subscription or Publication services, it is the EventName on which data is being posted or subscribed to respectively. In the case of Agent based Publication or Request-Response Invocation services, it is not applicable (NA is displayed). The Communication type is the Tuxedo communication paradigm that was used in the transaction. The possible values for Communication Type are Async, Sync, Queue, Event, ConvL, ClientAgent and PubAgent.</p>
Total	integer	Total number of objects processed for this operation.
Success	integer	The number of objects that were successfully identified for this operation and processed.
Failure	integer	The number of objects that were identified for this operation but for which processing failed.
LineIndex	string	LineIndex string Used as an index for the method. Its value is a concatenated string of the ServiceName and Operation parameters separated by a comma.
MeasurementInterval	integer	Displays the time (in seconds) since last time the adapter was reset, or if never reset, since the adapter started.



If no activity has been performed or the `perfMon` property is set to `off` the following message is displayed:

Method Invocation returned NO Data



The statistics provided by the `Failure` column take into account only application level invocation errors such as connection errors, incorrect Tuxedo service name, business logic error etc. Incorrect data supplied to the adapter (wrong communication type, unspecified Tuxedo service name) is not considered.



## getActivityStatisticsBySchema()

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.

Parameter	Type	Description
Schema Name	string	Name of the schema.

Returns	Type	Description
Service Name	string	Name of the service.
Total	integer	Total number of objects processed for this operation.
Success	integer	The number of objects that were successfully identified for this operation and processed.
Failure	integer	The number of objects that were identified for this operation but for which processing failed.



If no activity has been performed or the `perfMon` property is set to `off`, the following message is displayed:

Method Invocation returned NO Data



The statistics provided by the `Failure` column take into account only application level invocation errors such as connection errors, incorrect Tuxedo service name, business logic error etc. Incorrect data supplied to the adapter (wrong communication type, unspecified Tuxedo service name) is not considered.

# getAdapterServiceInformation()

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

Parameter	Type	Description
Service Name	string	Name of the service from which to get information. Default is ALL.

Returns	Type	Description
Line	integer	Sequential row number.
Service Name	string	Name of the service as defined at design-time.
Endpoint Name	string	Name of the endpoint used for this service.
Type	string	Type of the endpoint, for example, publisher or subscriber.
Quality of Service	string	Quality of service for the endpoint. For example RVCM or JMS Persistent.
Subject	string	Subject defined for this endpoint.
Class	string	Class associated with the endpoint.
Number of Messages	integer	Number of messages processed for this endpoint.

## getComponents()

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

Parameters	Type	Description
Component Name	string	Name of the component. If no value is enter, all components display.
Component Type	string	Any of Publisher, Subscriber, Timer, or IODescriptor. The default value is All.

Returns	Type	Description
Instance ID	string	Name of this adapter instance as defined at design-time.
Adapter Name	string	Name of the adapter.
Component Name	string	Name of the component.
Component Type	string	The name of the TIBCO Adapter SDK class for this component, such as MPublisher, MSubscriber, or MIODescriptorSource. For more information about the class, see your TIBCO Adapter SDK documentation.
Session Name	string	Name of the session.
Description	string	Information about this component, for example, time interval, signal type, validating publisher (or subscriber) etc.

# getConfig()

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

Returns	Type	Description
Instance ID	string	Configuration ID of this adapter.
Adapter Name	string	Name of the adapter.
Repository Connection	string	URL of the repository used for adapter instance.
Configuration URL	string	Location of the adapter project; either a file name or configuration URL.
Command	string	Command line arguments used to start the adapter.

# getConfigProperties()

Returns all attributes and elements for the given repository object.

Parameter	Type	Description
Property	string	Name of the property for which elements (tags) and attributes are desired. For example, agentone/startup.  If no value is given, all properties are returned.

Returns	Type	Description
Element Name	string	Repository directory for the property.
Attribute Name	string	Name of the repository object attribute.
Attribute Value	string	Value of the repository object attribute.
Line	integer	Line number in which this property is defined in the project file.

# getConnectionStatistics()

Returns the connection-related statistics for the adapter.

Returns	Type	Description
Connection ID	string	A unique identification of a connection.
Connection Type	string	The client type, possible values are 'Workstation Client' and 'Native Client'.
State	string	Current state: Connected or Disconnected
NumRetries	integer	This value is always 0.
TotalNumOperations	integer	Total number of operations processed by this connection since the adapter started.
CurrentNumOperations	integer	Total number of operations processed by this connection since the adapter started.
MeasurementInterval	integer	Displays the time (in seconds) since last time the adapter was reset, or if never reset, since the adapter started.
NumLostConnections	integer	This value is always 0.

## getHostInformation()

---

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

Returns	Type	Description
Name	string	Name of the property.
Value	string	Value of the property.

---

# getPerfMonSetting()

Returns the setting of the perfMon option. It returns the following information.

Returns	Type	Description
Setting	string	Value of the perfMon option.



# getPollingInterval()

Returns the current polling interval setting.

Returns	Type	Description
PollingInterval	integer	Polling interval in milliseconds used by the Event-based Publisher.

## getQueueStatistics()

Returns the current count of elements in the TIBCO Rendezvous event queues automatically spawned by Rendezvous for each adapter.

Returns	Type	Description
QueueID	string	A unique identification of a particular queue.
QueueType	string	The type displayed is RV.
QueueCount	integer	Current number of elements in the queue.
MaxQueueSize	integer	Maximum number of elements in the queue.
MeasurementInte rval	integer	Displays the time (in seconds) since last time the adapter was reset, or if never reset, since the adapter started.

## getRvConfig()

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

Parameter	Type	Description
Session Name	string	Name of the TIBCO Rendezvous session for which configuration is required (default is all).

Returns	Type	Description
Instance ID	string	The configuration ID of this adapter.
Adapter Name	string	Name of the adapter.
Session Name	string	Name of the session.
Service	string	Service parameter for this session.
Daemon	string	Daemon parameter for this session.
Network	string	Network parameter for this session.
Synchronous?	boolean	Returns 1 if this is a synchronous session, 0 otherwise.
Session Type	string	Type of session; one of <code>M_RV</code> , <code>M_RVCM</code> , or <code>M_RVCMQ</code> .
Certified Name	string	Name of this certified session.
Ledger File	string	Ledger file for this certified messaging session. Returns the empty string for sessions that are not certified messaging sessions.
CM Timeout	string	Timeout for this certified messaging session. Returns the empty string for sessions that are not certified messaging sessions.

## getStatus()

Retrieves basic status information about the adapter.

This information is fairly limited; for more detail, additional methods are provided (*getConfig()* on page 75 and *getRvConfig()* on page 77).

Returns	Type	Description
Instance ID	string	Configuration ID for this adapter instance.
Adapter Name	string	Name of the adapter.
Uptime	integer	Number of seconds since startup.
Messages Received	integer	Number of TIBCO Rendezvous messages received.
Messages Sent	integer	Number of TIBCO Rendezvous messages published.
New Errors	integer	Number of errors since the last call to this method.
Total Errors	integer	Total number of errors since startup.
Process ID	integer	Process ID of the application.
Host	string	Name of host machine on which this adapter is running.

## getThreadStatistics()

Returns the operation counts of the current threads.

Returns	Type	Description
ThreadID	string	A unique identification of a particular thread.
ThreadType	string	A type or key that will match this thread to the transport (RV or JMS).
TaskType	string	Short description of the tasks this thread processes. The type displayed would be <i>Async/Sync/Event/Queue/Convl/PubAgent/ClientAgent</i>
TaskCount	integer	Number of tasks processed by this thread.
MeasurementInterval	integer	Displays the time (in seconds) since last time the adapter was reset, or if never reset, since the adapter started.

# getTraceSinks()

Returns information about sinks to which traces currently go.

Parameters	Type	Description
Sink Name	string	Name of the sink for which you need information. If no name is specified, information about all sinks is returned. Default is all.
Role Name	string	Name of the role for which you need information for the specified sink or sinks. Default is all.

Returns	Type	Description
Instance ID	string	Name of this adapter instance as a string.
Adapter Name	string	Name of the application for this sink.
Sink Name	string	Name of the sink
Sink Type	string	Type of this sink. One of fileSink, rvSink, hawkSink, stderrSink.
Roles	string	Roles this sink supports, as a string. For example “warning, error, debug”.

## getVersion()

Retrieves version information for the current application. Two lines may be returned, one for the TIBCO Adapter SDK, one for the adapter.

Returns	Description
Instance ID	The configuration ID as a string, for example SDK.
Adapter Name	Name of the adapter as a string, for example agentone.
Version	Version number as a string, for example 1.1.

## preRegisterListener()

Preregisters an anticipated listener. Some sending applications can anticipate requests for certified delivery even before the listening applications start running. In such situations, the sender can preregister listeners, so TIBCO Rendezvous software begins storing outbound messages in the sender's ledger. If the listening correspondent requires old messages, it receives the backlogged messages when it requests certified deliver.

Parameters	Type	Description
Session Name	string	Name of the session that anticipates the listener.
Publisher Name	string	Name of the component for which the listener should be preregistered.
Listener Session Name	string	Name of the listener to preregister.

Returns OK if the listener was preregistered successfully, false otherwise.



## resetActivityStatistics()

---

Resets all the counts for the activity statistics.

## **resetConnectionStatistics()**

---

Resets all the counts for the connection statistics.

## resetThreadStatistics()

---

Resets all the counts for the thread statistics.

## reviewLedger()

Returns information retrieved from the ledger file of a 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.

Parameters	Type	Description
Session Name	string	Name of the TIBCO Rendezvous session for which ledger information is desired (default is all).
Subject	string	Name of the subject for which ledger information is desired.

Returns	Type	Description
Session Name	string	Name of the TIBCO Rendezvous CM session to which this information applies.
Subject	string	Subject name for this session.
Last Sent Message	integer	Sequence number of the most recently sent message with this subject name.
Total Messages	string	Total number of pending messages with this subject name.
Total Size	integer	Total storage (in bytes) occupied by all pending messages with this subject name.  If the ledger contains ten messages with this subject name, then this field sums the storage space over all of them.
Listener Session Name	string	Within each listener submessage, the Listener Session Name field contains the name of the delivery-tracking listener session.
Last Confirmed	string	Within each listener submessage, the Last Confirmed field contains the sequence number of the last message for which this listener session confirmed delivery.
Line	integer	Row number in ledger file.
Unacknowledged Messages	integer	Number of RVCM messages pending for this listener. The value is computed by subtracting the last sent sequence number from the last acknowledged sequence number.

## setPollingInterval()

---

Sets the polling interval for the publication service.

Parameter	Type	Description
PollingInterval	integer	Polling interval in milliseconds used by the Event-based Publisher.

## setTraceSinks()

Adds a role or changes the file limit of a previously specified sink.

Parameters	Type	Description
Sink Name	string	Name of the sink for which you want to add a role or change the file limit.
Role Name	string	Name of the role you want to add to this sink (warning, error, debug, or user defined). Default is all.
File Size	integer	Maximum file size for this sink.  This parameter is ignored if the sink specified by sinkName is not a file sink.

Returns OK if successful or an error if not successful.

## stopApplicationInstance()

---

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

# unRegisterListener()

Unregister a currently preregistered listener

Parameters	Type	Description
Session Name	string	Name of the session that anticipates the listener.
Publisher Name	string	Name of the component for which the listener should be preregistered.
Listener Session Name	string	Name of the listener to unregister.

This method returns true if the listener was unregistered successfully, false if otherwise.



## Appendix A **Adapter Properties File**

### Topics

---

- [Overview, page 136](#)
- [Properties File Format, page 137](#)
- [Predefined Properties, page 138](#)
- [Obfuscating or Encrypting a Password in a Properties File, page 141](#)

## Overview

---

The runtime adapter parses a properties file at startup. The default runtime adapter properties files are named `adtuxedo_wrkstn.tra` and `adtuxedo_native.tra`. Which TRA file you should use is determined by the client type you are using.

The default properties files are 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-whitespace character and ends at the first "=", ":", or whitespace character. The value starts at the first character after the equal sign (=). For example:

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

Properties defined in the properties file override the same properties defined in the project.

The `adtuxedo.authinfo` property in the adapter properties file indicates the authorization level set on the Tuxedo server.

There are three possible values:

**TPNOAUTH:** No authorization check. This is the default value.

**TPSYSAUTH:** System authorization.

**TPAPPAUTH:** Application authorization.

If the parameter is not specified, the adapter will attempt to connect to each of these levels sequentially.

## Properties File Format

---

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.

### Tagging Values for Obfuscation

The presence of a "#" character as the first character in a value (not the key) indicates that the value has been obfuscated or is to be obfuscated. The obfuscation command-line tool prompts for values to be obfuscated when it encounters a *value* with "#" as the first character in the properties file.

When the obfuscate tool is run, it rewrites the properties file with the obfuscated value in place. See [Obfuscating or Encrypting a Password in a Properties File](#) for more information.

# Predefined Properties

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



All paths inside a properties file, including Microsoft Windows directory names, must use forward slashes.

Table 41 Predefined Properties

Property	Description
<code>tibco.repourl</code>	Identifies the absolute pathname to the Designer project where the adapter configuration is defined.
<code>tibco.configurl</code>	Specifies the location of the adapter configuration inside the project file. If a relative path is specified, the adapter service is assumed to be under the default area in the project: <code>/tibco/private/adapter/</code> If an absolute path is specified, the adapter configuration is looked up in the project as defined by the argument.
<code>tibco.username</code>	The user name and password used by the repository server to access the project.
<code>tibco.password</code>	
<code>tibco.clientVar.var name</code>	Specifies the runtime values to substitute for global variables defined in the project. This value takes precedence over the named global value set in the project. Substitution takes place only at start up. Append the global variable to <code>tibco.clientVar</code> , then give its value. For example: <code>tibco.clientVar.DirLedger=TIBCO_HOME/adapter/adtuxedo/verios n_number/ledger</code>
<code>adtuxedo.perfMon &lt;on or off&gt;</code>	Turns the performance statistics microagent on or off. Default is off.
<code>adtuxedo.addCustomHawkMethodstoClassMAgent&lt;on or off&gt;</code>	If it is set on, the <code>getPollingInterval</code> and <code>setPollingInterval</code> methods are added to the class microagent. If it is set off, the invocation of these custom methods will return default values. Default is on.
<code>adtuxedo.enableDebug&lt;on or off&gt;</code>	If it is set on, the adapter will log trace messages with a Debug role; if it is set off, the adapter will not log those messages.

Table 41 Predefined Properties

Property	Description
<code>ntservice.name</code>	<p>Name for this Windows Service.</p> <p>This property is useful if you wish to have multiple Windows Services for the same executable. That is, you may wish to have two adapters running on the same machine. By specifying different service names and display names for the adapters, you can accomplish this.</p>
<code>ntservice.displayname</code>	<p>Name to display in the Services control for this Windows Service.</p> <p>This property is useful if you wish to have multiple Windows Services for the same executable. That is, you may wish to have two adapters running on the same machine. By specifying different service names and display names for the adapters, you can accomplish this.</p>
<code>ntservice.starttype</code>	<p>Start type for this Windows Service. Either manual or automatic. For example:</p> <p><code>ntservice.starttype automatic</code></p> <p>You can use this property to initially set the start type for the service, but once the service is installed, use the Windows Services control to change the start type of services.</p>
<code>ntservice.binary.path.absolute</code>	<p>Absolute path to the executable that is run when the service is started. For example:</p> <p><code>ntservice.binary.path.absolute</code>  <code>TIBCO_HOME/adaptor/adtuxedo/veriosn_number/bin/adtuxedo_wrkstn.exe</code></p>
<code>ntservice.interactive</code>	<p>Specifies whether the Windows Service is interactive. Either true or false.</p> <p><code>ntservice.interactive=true</code></p>
<code>ntservice.account</code>	<p>Username under which to run the Windows Service.</p> <p>You can use this property to initially set the account for the service, but once the service is installed, use the Services control to change the user account of services.</p>
<code>ntservice.password</code>	<p>Password for the username in the <code>ntservice.account</code> property.</p> <p>You can use this property to initially set the password for the user account, but once the service is installed, use the Services control to change the password.</p>
<code>adtuxedo.stackSize</code>	The stack size used by dispatcher threads.
<code>adtuxedo.PollingInterval</code>	The polling interval, in milliseconds, used by the Event-based publisher.
<code>adtuxedo.traceOldMessages</code>	To enable the printing of 4.x trace messages. Default is OFF.

Table 41 Predefined Properties

Property	Description
<code>tibco.env.WSNADDR</code>	Workstation Address of the Tuxedo server to connect to. Required if the adapter is used as a Workstation Client.
<code>tibco.env.TUXCONFIG</code>	TUXCONFIG of the Tuxedo server to connect to. Required if the adapter is used as a Native Client.
<code>tibco.env.FLDTBLDIR</code> 32	This contains a list of directories (separated by commas) where field tables can be found.
<code>tibco.env.FIELDTBLS</code> 32	This contains a list of the files in the table directories (separated by commas) that are to be used.

## Obfuscating or Encrypting a Password in a Properties File

---

This section describes password handling and how to obfuscate or encrypt a password in a properties file.

### Password Handling

At runtime, the adapter uses passwords to connect to the back-end application and interoperate with it. If you create a 4.x configuration using TIBCO Designer 5.6 and use the configuration against a 4.x adapter version, some special considerations are required for security.

If you plan to run the adapter locally, define the password values to be global variables. Before starting the adapter, include the runtime password as client variable in the adapter's TRA file and obfuscate it using the obfuscate tool. For example, if the password value is defined as `%%myPassword%%`, create a global variable named `myPassword` in the global variables section with no value and include the following entry in the adapter's `.tra` file:

```
tibco.clientVar.myPassword
```

During configuration, you can toggle between global variables and plain text values. These values are visible in non-readable form in the project. Plain text values are stored in the project in encrypted form.

### Obfuscating a Password

If the runtime adapter uses a password to access the vendor application, the password is stored in readable form as a value to the `tibco.clientVar.variable=#password` property in the runtime adapter's properties file. You can use the obfuscate tool to hide the password in the properties file so the password cannot be viewed by unauthorized users.

To obfuscate the password:

1. Using TIBCO Designer, open the adapter configuration and navigate to the panel where the password used by the runtime adapter to connect to Tuxedo is defined. In the password field, provide a user defined global variable, such as `%%MyPassword%%`. The global variable must also be added to the global variable list. See [Using Global Variables](#) for more information about creating global variables.
2. In the runtime adapter properties file, verify that the `tibco.clientVar.variable=#password` property is defined in the file.

3. Run the obfuscation tool supplied with adapter software against the properties file. This tool is named `obfuscate.exe` and resides in the `TIBCO_HOME\tibco\tra\veriosn_number\bin` directory.

The command syntax on Microsoft Windows for a default installation is:

```
TIBCO_HOME\tra\5.6\bin\obfuscate tra-file-path-name
```

where *tra-file-path-name* is the absolute pathname of the adapter properties file that contains the `tibco.clientVar.variable=# password` property.

For example, on Microsoft Windows:

```
TIBCO_HOME\tra\5.6\bin>obfuscate
```

```
TIBCO_HOME\adapter\adtuxedo\veriosn_number\bin\adtuxedoagent.tra
```

The password is now obfuscated and you can start the adapter with the changed properties file.



If you do not want to obfuscate a password, remove # at the beginning of the obfuscated password and replace it with the password in plain text.

### Encrypting a Password

Encryption is only supported for version 5.1 adapters and higher. If you have a property in a properties file that needs to be encrypted, follow these steps:

1. In the property file, add the #! characters in front of the value you want to encrypt. For example:

```
Repo.serverPassword = #!mysecret
```

2. Invoke the obfuscate utility from the command line:

```
TIBCO_HOME/tibco/tra/veriosn_number/bin/obfuscate.exe  
--propertyfile=tra-file-name
```

The next time you open the property file, `mysecret` will have been replaced with a random sequence of characters.



## Appendix B **Trace Messages**

This appendix explains the trace messages that are logged to a location specified at configuration time.

### Topics

---

- [Overview, page 144](#)
- [Trace Message Fields, page 145](#)
- [Status Messages, page 147](#)

## Overview

---

Trace messages provide information about adapter activities. The messages are logged to the console where the runtime adapter was started and to a log file. Trace messages can also be redirected to the TIBCO Hawk Display application, or sent to other applications using the TIBCO Rendezvous transport.

Each trace message can include the following fields:

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

The above fields are explained in [Trace Message Fields on page 145](#). The following diagram shows an example trace message and calls out the fields.

```
2002 Jul 04 23:11:07:649 GMT +5  
TuxedoAdapter.TuxedoAdapterConfiguration Info [Adapter]  
AETUX-00043 Completed processing the request, reply sent. Tracking  
Id: 8yI--A--DGG6gUBY--3Nzzw-9-zzw
```

The trace message above indicates a successfully processed request by the adapter Request-Response service. The 8yI--A--DGG6gUBY--3Nzzw-9-zzw tracking identifier included in the trace message uniquely identifies the message.

## Trace Message Fields

Each trace message includes the following fields:

*Table 42 Tracing Fields*

Field Name	Description
Timestamp	Timestamp of occurrence. For example, 2003 Feb 22 20:14:51:718 GMT -8.
Adapter Identifier	Name of the adapter that wrote the trace message. This is a combination of the adapter acronym and adapter configuration name. For example, the application identifier, ADB.publisher1 identifies a TIBCO Adapter for ActiveDatabase service named publisher1.
Role	<p>A role can be:</p> <ul style="list-style-type: none"> <li>• <b>Information.</b> 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.</li> <li>• <b>Warning.</b> An abnormal condition was found. Processing will continue, but special attention from an administrator is recommended.</li> <li>• <b>Error.</b> An unrecoverable error occurred. Depending on the error severity, the adapter may continue with the next operation or may stop altogether.</li> <li>• <b>Debug.</b> A developer-defined tracing message. In normal operating conditions, debug messages should not display.</li> </ul> <p>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.</p>

Table 42 Tracing Fields

Field Name	Description
Category	<div>One of the following:</div> <ul style="list-style-type: none"><li>• Adapter. The adapter is processing an event.</li><li>• Application. The adapter is interacting with the Tuxedo application.</li><li>• Configuration. The adapter is reading configuration information.</li><li>• Metadata. The adapter is retrieving metadata from the Tuxedo application.</li><li>• Publisher Service. The publication service is reporting this trace message.</li><li>• Request-Response Client Service. The Request-Response Invocation service is reporting this trace message.</li><li>• Request-Response Server. The Request-Response service is reporting this trace message.</li><li>• Shutdown. The adapter is shutting down.</li><li>• Startup. The adapter is starting.</li><li>• Subscription Service. The subscription service is reporting this trace message.</li></ul>
Status Code	<div>Unique code for the message and description. Status codes are identified by a unique number and description. If a trace message includes an error or warn role, the status code documentation includes a resolution. See <a href="#">Status Messages on page 147</a> for details.</div>
Tracking Identifier	<div>A unique identifier that is "stamped" on each message by the originating adapter. The tracking identifier remains in effect from a message's beginning to its completion as it is exchanged by TIBCO applications. If the adapter is the termination point of the message, the tracking identifier is not displayed in the trace message.</div> <div>You cannot modify the tracking identifier format or configure what information is displayed.</div>

## Status Messages

Message	Role	Category	Resolution
<b>AETUX-000001</b>	<b>Started initializing the adapter</b>		
	Information	Adapter	Indicates normal adapter operation. No action necessary.
<b>AETUX-000002</b>	<b>Completed initializing the adapter</b>		
	Information	Adapter	Indicates normal adapter operation. No action necessary.
<b>AETUX-000003</b>	<b>Terminating the adapter</b>		
	Information	Adapter	Indicates normal adapter operation. No action necessary.
<b>AETUX-000004</b>	<b>Service %1 not found in the repository.</b>		
	Error	Configuration	Verify the configuration of the specified service.
<b>AETUX-000005</b>	<b>Subscriber service %1 initialized</b>		
	Information	Adapter	Indicates normal adapter operation. No action necessary.
<b>AETUX-000006</b>	<b>No Subscribers configured in this adapter instance</b>		
	Information	Adapter	Indicates normal adapter operation. No action necessary.
<b>AETUX-000007</b>	<b>Started initializing Subscriber services</b>		
	Information	Adapter	Indicates normal adapter operation. No action necessary.
<b>AETUX-000008</b>	<b>Completed initializing Subscriber services</b>		
	Information	Adapter	Indicates normal adapter operation. No action necessary.
<b>AETUX-000009</b>	<b>Missing %1 in the configuration for the service: %2</b>		
	Error	Configuration	Check the repository.

Message	Role	Category	Resolution
AETUX-000010	No RPC servers configured in this adapter instance		
	Information	Adapter	Indicates normal adapter operation. No action necessary.
AETUX-000011	Started initializing RPC servers		
	Information	Adapter	Indicates normal adapter operation. No action necessary.
AETUX-000012	Completed initializing RPC services		
	Information	Adapter	Indicates normal adapter operation. No action necessary.
AETUX-000013	Completed initializing RPC server %1		
	Information	Adapter	Indicates normal adapter operation. No action necessary.
AETUX-000014	No data in Subscriber message %1		
	Error	Data	Resend the message from publisher. Check Rendezvous.
AETUX-000015	TRACKING INFORMATION : %1		
	Information	Adapter	Indicates normal adapter operation. No action necessary.
AETUX-000016	Request received. Operation Name : %1. Tracking ID : %2		
	Information	Adapter	Indicates normal adapter operation. No action necessary.
AETUX-000017	Data missing in the operation request.		
	Error	Data	Resend the message from RPCClient. Check Rendezvous.
AETUX-000019	Processing new Request: %1.		
	Information	Adapter	Indicates normal adapter operation. No action necessary.

Message	Role	Category	Resolution
<b>AETUX-00020</b>	<b>An MException Occurred. Type: %1 Description: %2.</b>		
	Error	Adapter	Refer to the TIBCO Adapter SDK documentation for more information.
<b>AETUX-00022</b>	<b>%1 : Failed %2%3</b>		
	Error	Application	Check if the service is running. Refer to the Oracle Tuxedo documentation for more information.
<b>AETUX-00023</b>	<b>%1 succeeded%2</b>		
	Information	Application	Indicates normal adapter operation. No action necessary.
<b>AETUX-00024</b>	<b>Queue Manager Diagnostic: %1.</b>		
	Error	Application	Check the queues configured. Refer to the Oracle Tuxedo documentation for more information.
<b>AETUX-00025</b>	<b>%1 : Failed, Ferror32: %2.</b>		
	Error	Application	Refer to the Oracle Tuxedo documentation for more information.
<b>AETUX-00027</b>	<b>Initializing TIBCO ActiveMatrix Adapter for Tuxedo: %1.</b>		
	Information	Adapter	Indicates normal adapter operation. No action necessary.
<b>AETUX-00028</b>	<b>Terminating TIBCO ActiveMatrix Adapter for Tuxedo: %1.</b>		
	Information	Adapter	Indicates normal adapter operation. No action necessary.
<b>AETUX-00029</b>	<b>%1: data missing.</b>		
	Error	Data	Include the specified field as part of the incoming message.
<b>AETUX-00030</b>	<b>Terminating since connection retry attempt failed %1 times.</b>		
	Error	Connection	Check the connection parameters specified during configuration, Verify that the Work Station Listener is running.

Message	Role	Category	Resolution
<b>AETUX-00031</b>	<b>Received data corresponds to schema class %1, expected %2 .</b>		
	Error	Data	Change the schema class encapsulating the data.
<b>AETUX-00032</b>	<b>Publisher Initiated</b>		
	Information	Adapter	Indicates normal adapter operation. No action necessary.
<b>AETUX-00033</b>	<b>No Publishers configured in this adapter instance</b>		
	Information	Adapter	Indicates normal adapter operation. No action necessary.
<b>AETUX-00034</b>	<b>Publisher service %1 initialized</b>		
	Information	Adapter	Indicates normal adapter operation. No action necessary.
<b>AETUX-00035</b>	<b>Registering Callback Failed due to : %1</b>		
	Error	Adapter	Refer to the Oracle Tuxedo documentation for more information.
<b>AETUX-00036</b>	<b>Registering Subscriber for Publishing Failed due to : %1</b>		
	Error	Adapter	Check whether the services are running.
<b>AETUX-00037</b>	<b>%1</b>		
	Information	Adapter	Indicates normal adapter operation. No action necessary.
<b>AETUX-00038</b>	<b>Polling Failed due to : %1</b>		
	Error	Adapter	Refer to the Oracle Tuxedo documentation for more information.
<b>AETUX-00039</b>	<b>Schema Mismatch or Data Parsing Error : %1</b>		
	Information	Adapter	Indicates normal adapter operation. No action necessary.
<b>AETUX-00040</b>	<b>Out of Memory Error in file, %1</b>		
	Error	Adapter	Try shutting down some processes and restarting the adapter.



Message	Role	Category	Resolution
<b>AETUX-00041</b>	<b>Data received, Tracking ID : %1, Processing..</b>		
	Information	Adapter	Indicates normal adapter operation. No action necessary.
<b>AETUX-00042</b>	<b>Completed processing the data, Tracking ID : %1</b>		
	Information	Adapter	Indicates normal adapter operation. No action necessary.
<b>AETUX-00043</b>	<b>Completed processing the request, reply sent. Tracking Id: %1</b>		
	Information	Adapter	Indicates normal adapter operation. No action necessary.
<b>AETUX-00044</b>	<b>Timer Not Configured : %1</b>		
	Information	Adapter	Indicates normal adapter operation. No action necessary.
<b>AETUX-00045</b>	<b>Checking unsolicited Messages and Publishing : %1</b>		
	Information	Adapter	Indicates normal adapter operation. No action necessary.
<b>AETUX-000046</b>	<b>Missing connection parameter %1 in the configuration.</b>		
	Error	Configuration	Check the repository.
<b>AETUX-000047</b>	<b>tpinit failed, could not connect to Tuxedo server, %1.</b>		
	Error	Connection	Check the connection parameters specified during configuration.
<b>AETUX-00048</b>	<b>Successfully sent the reply message%2. Tracking ID : %1</b>		
	Information	Adapter	Indicates normal adapter operation. No action necessary.
<b>AETUX-00049</b>	<b>%1: invalid communication type specified.</b>		
	Error	Data	Values for communication type are: Async, Queue, ConvI, Event.
<b>AETUX-00050</b>	<b>Terminate Subject : %1</b>		
	Information	Data	Indicates normal adapter operation. No action necessary.

Message	Role	Category	Resolution
<b>AETUX-00051</b>	<b>%1 threads created for session : %2</b>		
	Information	Data	Indicates normal adapter operation. No action necessary.
<b>AETUX-00052</b>	<b>Attempting to reconnect...</b>		
	Information	Connection	Indicates normal adapter operation. No action necessary.
<b>AETUX-00053</b>	<b>Connection Retry Interval configured : %1 milliseconds.</b>		
	Information	Connection	Indicates normal adapter operation. No action necessary.
<b>AETUX-00054</b>	<b>Number of Connection Retry Attempts configured : %1.</b>		
	Information	Connection	Indicates normal adapter operation. No action necessary.
<b>AETUX-00055</b>	<b>tpinit succeeded, connected to Tuxedo server.</b>		
	Information	Connection	Indicates normal adapter operation. No action necessary.
<b>AETUX-00056</b>	<b>Missing Encoding Type %1 in the configuration.</b>		
	Information	Configuration	Check the repository.
<b>AETUX-00057</b>	<b>No RPC clients configured in this adapter instance</b>		
	Information	Adapter	Indicates normal adapter operation. No action necessary.
<b>AETUX-00058</b>	<b>Started initializing RPC clients</b>		
	Information	Adapter	Indicates normal adapter operation. No action necessary.
<b>AETUX-00059</b>	<b>Completed initializing RPC invocation services</b>		
	Information	Adapter	Indicates normal adapter operation. No action necessary.
<b>AETUX-00060</b>	<b>Completed initializing RPC client %1</b>		
	Information	Adapter	Indicates normal adapter operation. No action necessary.

Message	Role	Category	Resolution
<b>AETUX-000061</b>	<b>The Request-Response Invocation Service %1 is not configured in the repository.</b>		
	Error	Adapter	Please check the event name passed to the callClient method of the agent
<b>AETUX-000062</b>	<b>The Publication Service %1 is not configured in the repository.</b>		
	Error	Adapter	Please check the event name passed to the callPub method of the agent
<b>AETUX-000063</b>	<b>Successfully Published the Message. Tracking ID : %1</b>		
	Information	Adapter	Indicates normal adapter operation. No action necessary.
<b>AETUX-000064</b>	<b>No data in the message from the Agent</b>		
	Error	Data	Resend the message from the agent. Check Rendezvous.
<b>AETUX-000065</b>	<b>No data in the message from the Agent</b>		
	Error	Data	Resend the message from the agent. Check Rendezvous.
<b>AETUX-000066</b>	<b>Received the Message, %1, from the Agent</b>		
	Information	Adapter	Indicates normal adapter operation. No action necessary.
<b>AETUX-000067</b>	<b>Received the Message, %1, from the Agent</b>		
	Information	Adapter	Indicates normal adapter operation. No action necessary.
<b>AETUX-000068</b>	<b>Sending the Request with the Timeout, %1. Tracking ID : %2</b>		
	Information	Adapter	Indicates normal adapter operation. No action necessary.
<b>AETUX-000069</b>	<b>The Endpoint %1 is not configured in the repository.</b>		
	Warning	Configuration	Please check the repository. Please ignore this message if you are using a repository configured using the 4.1.1 adapter.

Message	Role	Category	Resolution
<b>AETUX-000070</b>	<b>The Endpoint %1 is not configured in the repository.</b>		
	Error	Configuration	Please check the repository
<b>AETUX-000071</b>	<b>Conversational communication type is not supported for %1 buffer type.</b>		
	Error	Adapter	Please check the value specified for CommunicationType'.'
<b>AETUX-000072</b>	<b>The number of field occurrences is not uniform.</b>		
	Error	Adapter	Please check the data buffer.
<b>AETUX-000073</b>	<b>Invalid communication type value, %1' received. Expected 'Async'/'Queue'/'Conv'/'Event'.'</b>		
	Error	Adapter	Please check the value of the CommunicationType'field,intheincomingmessage.'
<b>AETUX-000074</b>	<b>Encoding Type set to %1 in the configuration.</b>		
	Information	Configuration	Indicates normal adapter operation. No action necessary.
<b>AETUX-000075</b>	<b>Stack size of the dispatcher threads set to %1 in the properties file.</b>		
	Information	Configuration	Indicates normal adapter operation. No action necessary.
<b>AETUX-000076</b>	<b>Number of Connection Retry Attempts Before Suspend configured : %1.</b>		
	Information	Connection	Indicates normal adapter operation. No action necessary.
<b>AETUX-000077</b>	<b>Polling Interval configured for replies of asynchronous Request-Response services: %1 milliseconds.</b>		
	Information	Configuration	Indicates normal adapter operation. No action necessary.
<b>AETUX-000078</b>	<b>Polling Interval configured for replies of asynchronous Request-Response services, after the maximum number of pending invocations is reached: %1 milliseconds.</b>		
	Information	Configuration	Indicates normal adapter operation. No action necessary.

Message	Role	Category	Resolution
<b>AETUX-00079</b>	<b>Polling interval used for Event-based publishers: %1 milliseconds.</b>		
	Information	Configuration	Indicates normal adapter operation. No action necessary.
<b>AETUX-00080</b>	<b>Invalid polling interval value, the value cannot be less than 1.</b>		
	Error	Configuration	Please enter a value greater than 0 for all polling interval fields.
<b>AETUX-00081</b>	<b>Invalid value specified for Maximum Number of Pending Invocations.</b>		
	Error	Configuration	Please specify a value greater than 0 for the Maximum Number of Pending Invocations.
<b>AETUX-00082</b>	<b>Maximum Number of Pending Invocations configured for asynchronous Request-Response services: %1.</b>		
	Information	Configuration	Indicates normal adapter operation. No action necessary.
<b>AETUX-00083</b>	<b>The String buffer received as the reply from Tuxedo is not null terminated. The reply length is: %1.</b>		
	Information	Application	Indicates normal adapter operation. No action necessary.
<b>AETUX-00086</b>	<b>The performance statistics microagent feature is on.</b>		
	Information	Adapter	Indicates normal adapter operation. No action necessary.
<b>AETUX-00087</b>	<b>The value provided for traceOldMessages in the .tra file is invalid.</b>		
	Information	Adapter	Indicates normal adapter operation. No action necessary.
<b>AETUX-910006</b>	<b>Startup Error. SDK Exception %1 occurred while creating a shutdown listener with parameters %2. The Repository URL is %3 and the Configuration URL is %4.</b>		
	Error	Adapter	Please verify your repository settings for validity of configuration for the shut down listener.

Message	Role	Category	Resolution
<b>AETUX-910007</b>	<b>Startup Error. Unable to create a connection with the target application %1 using connection parameters User Name: %2, Client Name: %3, Group Name: %4 and the target application error is %5</b>		
	Error	Connection	Please verify your repository settings for validity of connection parameters
<b>AETD-910012</b>	<b>Startup Error. Unable to create the Hawk microagent: %1.</b>		
	Error	Startup	Check the value of Hawk Daemon and Service provided during configuration.
<b>AETUX-910013</b>	<b>Startup Warning. Password decryption was unsuccessful.</b>		
	Warning	Startup	If you are running the adapter against configurations created by 4.x palette or your adapter configuration was saved in AE Version 4.0 or AE Version 5.0 format, the warning message can be ignored and the password will be handled correctly by the adapter. If your adapter configuration was saved in AE Version 5.1 format, open the configuration in TIBCO Designer and change the password to use a global variable. If the warning still displays, contact TIBCO support.
<b>AETUX-930001</b>	<b>Publication error. Publication Service, failed for Event Name : %1, encountered error %2 while trying to connect to target application Tuxedo. Connection parameters are User Name:%3, Client Name: %4, Group Name: %5, Flag: %6.</b>		
	Error	Connection	Please ensure that the target application is running. Verify your repository settings for validity of connection parameters
<b>AETUX-930003</b>	<b>Publication error. Publication service received event. Failed while converting event to Minstance as it could not get the class description. Repository URL is %1 and the Configuration URL is %2.</b>		
	Error	Publication Service	Please verify the configuration of the publication service and check that the schema/class definitions are present in the repository.
<b>AETUX-930014</b>	<b>Publication error. Publication service %1 received error while sending event over the wire. SDK Exception: %2</b>		
	Error	Publication Service	Please check repository settings for valid configuration of this publisher service.

Message	Role	Category	Resolution
<b>AETUX-940001</b>	<b>Request Response error. Request Response Service %1 received unexpected null data in incoming request. The Repository URL is %2 and the Configuration URL is %3</b>		
	Error	Request Response Service	Please check the configuration of the application that is requesting the event and make sure that it matches the inbound event definition for the above RequestResponse service.
<b>AETUX-940005</b>	<b>Request Response error. Request Response service %1 failed to deserialize the received MOperationRequest to MInstance. The Repository URL is %2 and the Configuration URL is %3</b>		
	Error	Request Response Service	Please check the configuration of the application that is requesting the event and make sure that it matches the inbound event definition for the above Request Response service.
<b>AETUX-940007</b>	<b>Request Response error. Error in incoming data for RPC service: %1. Missing mandatory parameter %2 for RPC input class %3</b>		
	Error	Request Response Service	Please specify the value for the Service Name parameter in the request data.
<b>AETUX-940009</b>	<b>Request Response error. Request Response service failed due to target application invocation error %1 for Tuxedo service: %2,Communication Type: %3 on application call %4.Tracking Id: %5</b>		
	Error	Request Response Service	Please check the target application command and the parameters passed to it and ensure that they are valid.
<b>AETUX-940010</b>	<b>Request Response error. Request Response service %1 failed to create the Reply Object.</b>		
	Error	Request Response Service	Please check the target application command and the parameters passed to it and ensure that they are valid.
<b>AETUX-940012</b>	<b>Request Response error. Request Response service %1 received an error while sending Data on Reply Address. Error Message %2</b>		
	Error	Request Response Service	Please check whether the requesting client is active.

Message	Role	Category	Resolution
<b>AETUX-950003</b>	<b>Request Response Invocation error. Request Response Invocation service: %1 received error while requesting event over the wire. Error: %2.</b>		
	Error	Request Response Invocation Service	Please check the repository settings for valid configuration of the Request Response Invocation service.
<b>AETUX-950008</b>	<b>Request Response Invocation error. Request Response Invocation service received error while requesting event over the wire. Error: %1. The timeout specified is %2</b>		
	Error	Request Response Invocation Service	Please check the repository settings for valid configuration of the Request Response Invocation service.
<b>AETUX-950010</b>	<b>Request Response Invocation error. Request Response Invocation service received null reply while requesting event over the wire.</b>		
	Error	Request Response Invocation Service	Please ensure that the target application is running. Also check the configuration of the Request Response Invocation service.
<b>AETUX-950011</b>	<b>Request Response Invocation error. Request Response Invocation service received timeout error while requesting event over the wire, Tracking ID : %1</b>		
	Error	Request Response Invocation Service	Please check repository settings for valid configuration of the Request Response Invocation endpoint for this service.
<b>AETUX-990001</b>	<b>Shutdown error. Failed to deactivate the timer: %1.SDK exception = %2</b>		
	Error	Adapter	Refer to the SDK Documentation for the description of this SDK error code.
<b>AETUX-990002</b>	<b>Shutdown error. SDK exception = %1</b>		
	Error	Adapter	Refer to the SDK Documentation for the description of this SDK error code.
<b>AETUX-990003</b>	<b>Shutdown error. Failed to cleanup Hawk microagent. SDK exception = %1</b>		
	Error	Adapter	Refer to the SDK Documentation for the description of this SDK error code.



Message	Role	Category	Resolution
<b>AETUX-990004</b>	<b>Shutdown error. Failed to cleanup threads. SDK exception = %1</b>		
	Error	Adapter	Refer to the SDK Documentation for the description of this SDK error code.
<b>AETUX-990005</b>	<b>Shutdown error. Error in disconnecting from Tuxedo. Error:%1</b>		
	Error	Adapter	Check if the Tuxedo application is up and running.
<b>AETUX-920015</b>	<b>Subscription error. Subscription service failed due to target application invocation error %1 for Tuxedo service: %2, Communication Type: %3 on application call %4. Tracking Id: %5</b>		
	Error	Subscription Service	Please check the target application command and the parameters passed to it and ensure that they are valid.
<b>AETUX-920020</b>	<b>Subscription error. Subscription service failed due to target application invocation error %1 for RequestQueueName: %2, QueueSpaceName: %3,Communication Type: %4 on application call %5.Tracking Id: %6</b>		
	Error	Subscription Service	Please check the target application command and the parameters passed to it and ensure that they are valid.
<b>AETUX-920021</b>	<b>Subscription error. Subscription service failed due to target application invocation error %1 for ReplyQueueName: %2, QueueSpaceName: %3,Communication Type: %4 on application call %5.Tracking Id: %6</b>		
	Error	Subscription Service	Please check the target application command and the parameters passed to it and ensure that they are valid.
<b>AETUX-920022</b>	<b>Subscription error. Subscription service failed due to target application invocation error %1 for Communication Type: %2,Event Name : %3 on application call %4.Tracking Id: %5</b>		
	Error	Subscription Service	Please check the target application command and the parameters passed to it and ensure that they are valid.

Message	Role	Category	Resolution
AETUX-920023	Subscription error. Subscription service failed due to target application invocation error %1 for Tuxedo Service : %2,Communication type: %3 on application call %4.Tracking Id: %5		
Error	Subscription Service	Please check the target application command and the parameters passed to it and ensure that they are valid.	
AETUX-920024	Subscription error. Subscription service with Communication type: %1 failed due to target application invocation error %2 on application call %3.Tracking Id: %4		
Error	Subscription Service	Please check the target application command and the parameters passed to it and ensure that they are valid.	
AETUX-940025	Subscription error. Subscription service failed due to target application invocation error %1 for Tuxedo service: %2,Communication Type: %3 on application call %4.Tracking Id: %5		
Error	Subscription Service	Please check the target application command and the parameters passed to it and ensure that they are valid.	
AETUX-920001	Subscription error. Subscription service: %1 received an unexpected event of type = %2.The Repository URL is %3 and the Configuration URL is %4.Error Message: %5		
Error	Subscription Service	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.	
AETUX-920002	Subscription error. Subscription service: %1 failed to deserialize the event received. SDK exception thrown is %2. The Repository URL is %3 and the Configuration URL is %4.		
Error	Subscription Service	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	
AETUX-920003	Subscription error. Subscription service: %1 received inbound event with null data. The Repository URL is %2 and the Configuration URL is %3		
Error	Subscription Service	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	

Message	Role	Category	Resolution
<b>AETUX-920014</b>	<b>Subscription error. Subscription service %1 could not process the inbound event for Communication Type : %2,due to connection error against target application Tuxedo using parameters Client Name: %3, Group Name: %4, User Name : %5.</b>		
	Error	Subscription Service	Ensure that the target application is running. Check the validity of the connection parameters.
<b>AETUX-920017</b>	<b>Subscription error. Subscription service could not send response on reply subject. Error: %1</b>		
	Error	Subscription Service	Please check your repository settings for the publish endpoint of this subscription service.
<b>AETUX-920025</b>	<b>Method %1 : Failed, for Field Id: %2,Field Name: %3. Ferror32: %4, Error Description : %5.</b>		
	Error	Application	Check if the correct Field IDs are present in the configuration.
<b>AETUX-920029</b>	<b>Publication Service. Connection error. Error occurred while re-booting services. Please boot again...</b>		
	Error	Application	Please do tmsshutdown -y and tmboot -y again.
<b>AETUX-920030</b>	<b>Maximum number of reconnection attempts has been made. The adapter will shutdown now...</b>		
	Information	Reconnection	Indicates normal adapter operation. No action necessary.
<b>AETUX-920031</b>	<b>The Reconnect attempt number is: %1</b>		
	Information	Reconnection	Indicates normal adapter operation. No action necessary.
<b>AETUX-920033</b>	<b>Unknown Exception Caught</b>		
	Error	Adapter	Exception caught in master exception.
<b>AETUX-920034</b>	<b>Instantiating Tuxedo Connection Number: %1</b>		
	Information	Adapter	Indicates normal adapter operation. No action necessary.
<b>AETUX-920035</b>	<b>Unable to open connection with Tuxedo</b>		
	Information	Adapter	Check whether Tuxedo services are booted properly.

Message	Role	Category	Resolution
<b>AETUX-920036</b>	<b>Restoring Connection with Tuxedo</b>		
	Information	Adapter	Indicates normal adapter operation. No action necessary.
<b>AETUX-920037</b>	<b>Message received by the thread: %1</b>		
	Information	Adapter	Indicates normal adapter operation. No action necessary.
<b>AETUX-920038</b>	<b>Check if the correct Field IDs are present in the configuration.</b>		
	Error	Application	Indicates that the schema which BW configures is wrong.
<b>AETUX-920039</b>	<b>Check if the correct Field IDs are present in the configuration. Service Name: %1, Error Description: %2, trackID: %3</b>		
	Error	Application	Check if the correct Field IDs are present in the configuration.
<b>AETUX-920040</b>	<b>Publication error. Publication service failed due to target application invocation error: %1 for QueueName: %2, QueueSpaceName: %3, Communication Type: %4 on application call %5. Tracking Id: %6</b>		
	Error	Publication Service	Please check the target application command and the parameters passed to it and ensure that they are valid.
<b>AEADTUXEDO-900001</b>	<b>Mandatory Field Error/No value specified for mandatory field [%1]</b>		
	Error	CONFIG	Please specify a value for the field.
<b>AEADTUXEDO-900002</b>	<b>Palette error./The Field value [%1] must be greater than or equal to [%2], and less than or equal to [%3].</b>		
	Error	CONFIG	The value specified must be greater than or equal to 0, and less than or equal to 65535.
<b>AEADTUXEDO-900003</b>	<b>Field Validation Error/Invalid value entered for [%1]</b>		
	Warning	CONFIG	Please specify a value greater than 0
<b>AEADTUXEDO-900004</b>	<b>Name Conflict Error/Another instance with same name [%1] exists in the project. Please specify a unique instance name.</b>		
	Warning	CONFIG	Please specify a unique instance name

Message	Role	Category	Resolution
<b>AEADTUXEDO-900005</b>	<b>Invalid subject specified for [%1] of service [%2].</b>		
	Warning	CONFIG	Please specify a unique service name
<b>AEADTUXEDO-900006</b>	<b>Invalid subject specified for [%1] of service [%2].</b>		
	Warning	CONFIG	Please Specify a correct subject.
<b>AEADTUXEDO-900007</b>	<b>Illegal first character, The name of [%1] must start with an alphabet.</b>		
	Error	CONFIG	Please specify a Legal first character.
<b>AEADTUXEDO-900008</b>	<b>Unable to find the specified file specified in [%1]. Please make sure that you specified an absolute path and the file exists..</b>		
	Error	CONFIG	Specify a valid File path and make sure it exists.
<b>AEADTUXEDO-900009</b>	<b>Invalid Service name:/Service name can only contain alphanumeric characters with no embedded spaces and cannot exceed maximum length of 80 characters.</b>		
	Error	CONFIG	Please type in a valid name. Adapter Configuration names can only have alphanumeric characters and can be up to 80 characters long.
<b>AEADTUXEDO-900010</b>	<b>Incomplete selection. You must select at least one [%1].</b>		
	Error	DTA	Please select at least one
<b>AEADTUXEDO-900011</b>	<b>Endpoint [%1] is already referenced by another service. Please select a different endpoint.</b>		
	Error	DTA	Please select a different endpoint.
<b>AEADTUXEDO-900012</b>	<b>Invalid selection for [%1]. The selected endpoint must contain alphanumeric characters only. Please select an endpoint with a valid name.</b>		
	Information	DTA	Please select an endpoint with a valid name.
<b>AEADTUXEDO-900013</b>	<b>Invalid session name specified for [%1]. Please make sure the session name contains only alphanumeric characters only.</b>		
	Warning	DTA	Please make sure the session name contains only alphanumeric characters only.
<b>AEADTUXEDO-900014</b>	<b>Specified server type [%1] does not match the actual server type [%2]. Please enter the correct server type.</b>		
	Error	SRVC	Please enter the correct server type.

Message	Role	Category	Resolution
<b>AEADTUXEDO-9 10000</b>	<b>Exception in set model.</b>		
	Information	DTA	Indicates normal adapter operation. No action necessary.
<b>AEADTUXEDO-9 10001</b>	<b>Terminate Subject is null. Please provide a valid terminate subject.</b>		
	Information	DTA	Please provide a valid terminate subject.
<b>AEADTUXEDO-9 10002</b>	<b>Palette error. This field is mandatory.</b>		
	Information	DTA	This field is mandatory. Please provide correct values.
<b>AEADTUXEDO-9 10003</b>	<b>Invalid Instance Name/Adapter Configuration names must have only alphanumeric characters and can be up to 80 characters long. Please type in a valid name.</b>		
	Information	DTA	Please type in a valid name.
<b>AEADTUXEDO-9 10005</b>	<b>Field: Encoding is null</b>		
	Information	DTA	Please provide a valid encoding
<b>AEADTUXEDO-9 10006</b>	<b>Field: Encoding is null</b>		
	Warning	SRVC	Please provide a valid encoding
<b>AEADTUXEDO-9 10007</b>	<b>Incorrect Values for Initialization Flag./Please enter the Appropriate values for Initialization flags [ 1,2,4,8,10,20,40,47 ].</b>		
	Error	CONFIG	Please enter the Appropriate values for Initialization flags [ 1,2,4,8,10,20,40,47 ].
<b>AEADTUXEDO-9 10008</b>	<b>Incorrect Values for Initialization Flag./Please enter the Appropriate values for Initialization flags [ 1,2,4,8,10,20,40,47 ]..</b>		
	Information	SRVC	Please enter the Appropriate values for Initialization flags [ 1,2,4,8,10,20,40,47 ].
<b>AEADTUXEDO-9 10009</b>	<b>Field: No Retries Before Suspend/The Port must be greater than or equal to 0, and less than or equal to 65535.</b>		
	Information	SRVC	The Port must be greater than or equal to 0, and less than or equal to 65535.

Message	Role	Category	Resolution
<b>AEADTUXEDO-9 10010</b>	<b>Field: No Retries Before Suspend/The Port must be greater than or equal to 0, and less than or equal to 65535.</b>		
	Error	CONFIG	The Port must be greater than or equal to 0, and less than or equal to 65535.
<b>AEADTUXEDO-9 10011</b>	<b>Field: Max No of Retries/Palette error. The Maximum Number of Retries must be greater than or equal to -1, and less than or equal to 65535.</b>		
	Error	CONFIG	The Maximum Number of Retries must be greater than or equal to -1, and less than or equal to 65535.
<b>AEADTUXEDO-9 10012</b>	<b>Field: Max No of Retries/Palette error. The Maximum Number of Retries must be greater than or equal to -1, and less than or equal to 65535.</b>		
	Error	CONFIG	The Maximum Number of Retries must be greater than or equal to -1, and less than or equal to 65535.
<b>AEADTUXEDO-9 10013</b>	<b>Field: Sleep Between Retries/Palette error. The Sleep Between Retries must be greater than or equal to 0, and less than or equal to 65535.</b>		
	Error	CONFIG	The Sleep Between Retries must be greater than or equal to 0, and less than or equal to 65535.
<b>AEADTUXEDO-9 10014</b>	<b>Polling Interval For Outstanding Invocation/Palette error. The value entered should be greater than 0 and less than 65535.</b>		
	Error	CONFIG	The value entered should be greater than 0 and less than 65535.
<b>AEADTUXEDO-9 10015</b>	<b>The field is empty. Please provide valid value for it.</b>		
	Error	CONFIG	Please provide valid value for the empty field
<b>AEADTUXEDO-9 10016</b>	<b>Exception in loadGlobalVariablesForTuxedo method.</b>		
	Error	CONFIG	Contact TIBCO Support.
<b>AEADTUXEDO-9 10017</b>	<b>Exception in createTuxedoSchemaResourcesOnDnd method.</b>		
	Error	CONFIG	Contact TIBCO Support.
<b>AEADTUXEDO-9 10018</b>	<b>Exception in didAddToDocument method.</b>		
	Error	CONFIG	Contact TIBCO Support.

Message	Role	Category	Resolution
<b>AEADTUXEDO-9 10020</b>	<b>JMS Service Configured/This adapter version does not support JMSservices, but one was found. Use the goto button to select it.'</b>		
	Error	CONFIG	Please do not configure JMS version for this version of adapter'
<b>AEADTUXEDO-9 10022</b>	<b>XML Wire Format found/This adapter version does not support XML Wire Format.</b>		
	Error	CONFIG	Please do not use the XML wireformat. This adapter version does not support XML WireFormat.'
<b>AEADTUXEDO-9 10025</b>	<b>Exception in createTerminatorSubEndpoint method: While creating terminate subscriber endpoint.</b>		
	Error	CONFIG	Contact TIBCO Support.
<b>AEADTUXEDO-9 10026</b>	<b>Invalid Service name/Service name can only contain alphanumeric characters with no embedded spaces and cannot exceed maximum length of 80 characters.</b>		
	Error	CONFIG	Please follow the naming convention for naming of services
<b>AEADTUXEDO-9 10028</b>	<b>Invalid Field: CArray Request Length. Please enter a CArray Request Length value greater than 0.</b>		
	Error	CONFIG	Please enter a CArray Request Length value greater than 0.
<b>AEADTUXEDO-9 10030</b>	<b>Invalid Field: CArray Reply Length. Please enter a CArray Reply Length value greater than 0.</b>		
	Error	CONFIG	Please enter a CArray Reply Length value greater than 0.
<b>AEADTUXEDO-9 10032</b>	<b>Please enter all fields.</b>		
	Error	CONFIG	Please enter all fields
<b>AEADTUXEDO-9 10033</b>	<b>Invalid File. Please check the FML32 file</b>		
	Error	CONFIG	Please check the FML32 file. Make sure that you are using a valid FML32 file



Message	Role	Category	Resolution
<b>AEADTUXEDO-9 10037</b>	<b>Error During Rename: Read-Only File</b>  Error	CONFIG	Please checkout the resource and try renaming again. You can select the resource to be checked out by clicking the Go To Resource button.
<b>AEADTUXEDO-9 10038</b>	<b>Error During Delete.</b>  Error	CONFIG	Please checkout the resource and try deleting again. You can select the resource to be checked out by clicking the "Go To Resource" button.
<b>AEADTUXEDO-9 10039</b>	<b>Connection Retry Mechanism Warning</b>  Error	CONFIG	Please do not use the transient retry mechanism as this adapter version does not support.'
<b>AEADTUXEDO-9 10040</b>	<b>Use Async Invocation found</b>  Error	CONFIG	Please do not use the 'UseAsyncInvocation'
<b>AEADTUXEDO-9 10041</b>	<b>Polling Interval for Outstanding Invocation found</b>  Error	CONFIG	Please do not use 'PollingIntervalforOutstandingInvocations'
<b>AEADTUXEDO-9 10042</b>	<b>Polling Interval for Replies found</b>  Error	CONFIG	Please do not use 'PollingIntervalforReplies'
<b>AEADTUXEDO-9 10043</b>	<b>Field is empty.</b>  Error	CONFIG	Please make sure that the values in the field are provided
<b>AEADTUXEDO-9 10044_A</b>	<b>The no of threads should not be less than 0 for sessions other than Hawk Session.</b>  Error	CONFIG	Please make sure that the aechema file created for the service is writable
<b>AEADTUXEDO-9 10044_B</b>	<b>Cannot Rename/The adapter configuration could not be renamed. Please ensure that the aeschema file is writable.</b>  Error	CONFIG	The no of threads should not be less than 1 for Hawk Session.

Message	Role	Category	Resolution
<b>AEADTUXEDO-9 10045</b>	<b>Palette error./ The value should be a valid integer greater than 0 Please type in a valid value.</b>		
	Error	CONFIG	The value should be a valid integer greater than 0 Please type in a valid value.
<b>AEADTUXEDO-9 10046</b>	<b>Palette error. The value for Request Schema Reference should not be empty.</b>		
	Error	CONFIG	The value for Request Schema Reference should not be empty.
<b>AEADTUXEDO-9 10047</b>	<b>Palette error. The value for Reply Schema Reference should not be empty.</b>		
	Error	CONFIG	The value for Reply Schema Reference should not be empty.
<b>AEADTUXEDO-9 10052</b>	<b>Palette error. The value for Request Header Reference is invalid.</b>		
	Error	CONFIG	Please specify the correct path for the Request Header Reference file.
<b>AEADTUXEDO-9 10053</b>	<b>Palette error. The value for Reply Header Reference is invalid.</b>		
	Error	CONFIG	Please specify the correct path for the Reply Header Reference file.
<b>AEADTUXEDO-9 10054</b>	<b>Invalid Queue cofiguration. The value of [%1] is null.</b>		
	Error	CONFIG	Please specify both the Queue Space Name and the Queue Name.





## Appendix C **Message Formats**

This appendix details the additional request fields available in the message schema for subscription and request-response messages.

### Topics

---

- [Subscription Service Additional Fields, page 172](#)
- [Request-Response Service Additional Fields, page 174](#)

## Subscription Service Additional Fields

Table 43 lists the additional fields that are available in the request schema for a subscription service.

Table 43 Subscription Service Additional Fields for the Request Schema

Field	Description	Usage	Type
CommunicationType	The Tuxedo communication paradigm used to invoke the Tuxedo service. The possible values are: <ul style="list-style-type: none"><li>• Async</li><li>• Queue</li><li>• Event</li><li>• Convl</li></ul>	Used to specify the type of Tuxedo communication paradigm to be used for an adapter service.	String
ServiceName	The name of the Tuxedo service to be invoked by the adapter.	Used when the Tuxedo communication paradigm is set to Async, Queue or Convl. It is not used when it is set to Event. If set to Queue, the service name and the request queue name must be identical.	String
QueueSpace	The name of the queue space configured in the Tuxedo integrated application. This queue space will contain the request queue, reply queue, and error queue.	Used only if the Tuxedo communication paradigm is set to Queue-based.	String
ReplyQueue	The name of the reply queue defined in the Tuxedo queue space. Replies from the Tuxedo services will be posted in this queue.	Used only if the Tuxedo communication paradigm is set to Queue-based.	String
ConvlBatchField	The field in the request class that contains the name of the Tuxedo field where the batch size for the transfer of data is specified.	Used only if the Tuxedo communication paradigm is set to Conversational and appears in the schema only if the buffer type is configured as FML32.	String

**Table 43** *Subscription Service Additional Fields for the Request Schema*

Field	Description	Usage	Type
EventNameField	The field in the request class that contains the name of the Tuxedo field where the event is specified. This event name is used to post subscribed data to Tuxedo.	Used only if the Tuxedo communication paradigm is set to Event-based and appears in the schema only if the buffer type is configured as FML32.	String
EventName	The field in the request class where the event name is specified. This event name is used to post subscribed data to Tuxedo.	Used only if the Tuxedo communication paradigm is set to Event-based and appears in the schema only if the buffer type is configured String or CArray.	String

[Table 44](#) lists the additional fields that are available in the reply schema for a Subscription service. The adapter populates these fields in the reply.

**Table 44** *Subscription Service Additional Fields for the Reply Schema*

Field	Description	Type
ErrorCode	The field that contains the error code. Example: AETUX-00031	String
ErrorType	The field that contains the error type. The possible values are: <ul style="list-style-type: none"> <li>Tuxedo System</li> <li>Connection</li> </ul>	String
ErrorDetails	The field that contains the complete error description. Tuxedo returned error code and details will be part of this description.	String

## Request-Response Service Additional Fields

Table 45 lists the additional fields that are available in the request schema used for a Request-Response service.

Table 45 Request-Response Service Additional Fields for the Request Schema

Field	Description	Usage	Type
ServiceName	The name of the Tuxedo service to be invoked by the adapter.	Used when the Tuxedo communication paradigm is set to Async, Queue or Convl. It is not used when it is set to Event.  If set to Queue, the service name and the request queue name must be identical.	String

Table 46 lists the additional fields that are available in the reply schema for a Request-Response service. The adapter populates these fields in the reply shcema.

Table 46 Request-Response Service Additional Fields for the Reply Schema

Field	Description	Type
ErrorCode	The field that contains the error code. Example: AETUX-00031	String
ErrorType	The field that contains the error type. The possible values are: <ul style="list-style-type: none"><li>Tuxedo System</li><li>Connection</li></ul>	String
ErrorDetails	The field that contains the complete error description. Tuxedo returned error code and details will be part of this description.	String
isEnd	This field is set by the adapter. This happens only if at least 1 row has been retrieved.	Boolean
tpurcode	This field can get the value of rcode , an application defined return code.	String



## Appendix D **Adapter Agent API References**

This appendix explains some adapter agent methods based on TIBCO Rendezvous or TIBCO EMS.

### Topics

---

- [TIBCO Rendezvous Adapter Agent Methods, page 176](#)
- [TIBCO EMS Adapter Agent Methods, page 182](#)

## TIBCO Rendezvous Adapter Agent Methods

---

The following adapter agent methods use TIBCO Rendezvous. The header file relating to them is `adapteragent.h` and the link library relating to them is `agent.lib` for Microsoft Windows or `libagent.so/libagent.sl` for Unix.

With regard to publication services, there are two methods for Asynchronous Rendezvous Adapter Agent, `callPub()` and `callArrayPub()`. There are two methods for Synchronous Rendezvous Adapter Agent, `callSyncPub()` and `callSyncArrayPub()`.

### The `callPub` Method

- Function

```
int callPub(char * serviceName, char * sendbuf, char * service, char * network,
char * daemon, char * subject);
```

- Description

This method is for publication services based on Asynchronous Rendezvous Adapter Agent. The user's Tuxedo service invokes the method to send the request which uses FML32/STRING buffers to TIBCO Rendezvous.

- Parameters

`char * serviceName`– The name of the adapter service to be used

`char * sendbuf`– The request buffer of type FML32/STRING

`char * service`– The service parameter of the TIBCO Rendezvous transport which is used internally

`char * network`– The network parameter of the TIBCO Rendezvous transport which is used internally

`char * daemon`– The daemon parameter of the TIBCO Rendezvous transport which is used internally

`char * subject`– The message subject on which the message is to be sent from the Adapter Agent to the adapter

- Return Value

The method returns 1 on success, and 0 on failure. Error messages are written into the Tuxedo ULOG file.

### The `callArrayPub` Method

- Function

```
int callCArrayPub(char * serviceName, char * sendbuf, char * service, char *
network, char * daemon, char * subject, long sendlen)
```

- **Description**

This method is for publication services Asynchronous Rendezvous Adapter Agent. The user's Tuxedo service invokes the method to send the request which uses CArray buffers to TIBCO Rendezvous.

- **Parameters**

char \* serviceName– The name of the adapter service to be used

char \* sendbuf– The request buffer of type CArray

char \* service– The service parameter of the TIBCO Rendezvous transport that is used internally

char \* network– The network parameter of the TIBCO Rendezvous transport that is used internally

char \* daemon– The daemon parameter of the TIBCO Rendezvous transport that is used internally

char \* subject– The message subject on which the message is to be sent

long sendlen– The length of the CArray buffer to be sent

- **Return Value**

The method returns 1 on success, and 0 on failure. Error messages are written into the Tuxedo ULOG file.

## The callSyncPub Method

- **Function**

```
int callSyncPub(char * serviceName, char * sendbuf, char * service, char *
network, char * daemon, char * subject, unsigned long agentTimeout);
```

- **Description**

This method is for publication services based on Synchronous Rendezvous Adapter Agent. The user's Tuxedo service invokes the method to send the request which uses FML32/STRING buffers to TIBCO Rendezvous, and then the agent waits for the feedback from the adapter.

- **Parameters**

char \* serviceName– The name of the adapter service to be used

char \* sendbuf– The request buffer of type FML32/STRING

`char * service`– The service parameter of the TIBCO Rendezvous transport which is used internally

`char * network`– The network parameter of the TIBCO Rendezvous transport which is used internally

`char * daemon`– The daemon parameter of the TIBCO Rendezvous transport which is used internally

`char * subject`– The message subject on which the message is to be sent from the agent to the adapter

`unsigned long agentTimeout` – The timeout of the agent in seconds.

- **Return Value**

The method returns 1 on success, and 0 on failure. Error messages are written into the Tuxedo ULOG file.

### The `callSyncCArrayPub` Method

- **Function**

`int callSyncCArrayPub(char * serviceName, char * sendbuf, char * service, char * network, char * daemon, char * subject, long sendlen, unsigned long agentTimeout)`

- **Description**

This method is for publication services based on Synchronous Rendezvous Adapter Agent. The user's Tuxedo service invokes the method to send the request which uses `CArray` buffers to TIBCO Rendezvous, and then the agent waits for the feedback from the adapter.

- **Parameters**

`char * serviceName`– The name of the adapter service to be used

`char * sendbuf`– The request buffer of type `CArray`

`char * service`– The service parameter of the TIBCO Rendezvous transport that is used internally

`char * network`– The network parameter of the TIBCO Rendezvous transport that is used internally

`char * daemon`– The daemon parameter of the TIBCO Rendezvous transport that is used internally

`char * subject`– The message subject on which the message is to be sent

`long sendlen`– The length of the `CArray` buffer to be sent

`unsigned long agentTimeout` – The timeout of the agent in seconds.

- **Return Value**

The method returns 1 on success, and 0 on failure. Error messages are written into the Tuxedo ULOG file.

When some abnormal situations occur, `callPub` or `callArrayPub` will still return a success message, but `callSyncPub` or `callSyncArrayPub` will report the errors by returning a failure message, which is received from the adapter. Those situations are as follows:

- The `serviceName` (the first parameter of each method) is different from the Name in the Configuration tab of Publication Service configured in TIBCO Designer.
- The subject (the six parameter of each method) is different from the Subject in the Configuration tab of PubAgentSubscriber (TuxedoAdapterConfiguration>Advanced>Sessions>DefaultRVSession>PubAgentSubscriber) configured in TIBCO Designer.
- The adapter suddenly terminates.
- The adapter fails to publish data to TIBCO environment.

## The `callClient` Method

- **Function**

```
int callClient (char * serviceName, char * sendbuf, unsigned long
adapterTimeout, char * service, char * network, char * daemon, unsigned long
agentTimeout, char * subject, void** replyBuf)
```

- **Description**

This method is for Request-Response invocation services. The user's Tuxedo service invokes the method to send the request which uses FML32/STRING buffers to TIBCO Rendezvous.

- **Parameters**

`char * serviceName`—The name of the adapter service to be used

`char * sendbuf`—The request buffer of type FML32/STRING

`unsigned long adapterTimeout`— The timeout of the adapter in seconds

`char * service`—The service parameter of the TIBCO Rendezvous transport that is used internally

`char * network`— The network parameter of the TIBCO Rendezvous transport that is used internally

`char * daemon`— The daemon parameter of the TIBCO Rendezvous transport that is used internally

`unsigned long agentTimeout`– The timeout of the agent in seconds

`char * subject`– The message subject on which the message is to be sent from the Adapter Agent to the adapter

`void ** replyBuf`– The address of the reply buffer of type FML32/STRING. The replyBuf must be allocated by `tpalloc`.

- **Return Value**

The method returns 1 on success, and 0 on failure. Error messages are written into the Tuxedo ULOG file.

## The `callCArrayClient` Method

- **Function**

`int callCArrayClient(char * serviceName, char * sendbuf, unsigned long adapterTimeout, char * service, char * network, char * daemon, unsigned long agentTimeout, char* subject, void** replyBuf, long sendlen, unsigned int* rcvlen)`

- **Description**

This method is for Request-Response invocation services. The user's Tuxedo service invokes the method to send the request which uses `CArray` buffers to TIBCO Rendezvous.

- **Parameters**

`char * serviceName`–The name of the adapter service to be used

`char * sendbuf`–The request buffer of type `CArray`

`unsigned long adapterTimeout`– The timeout of the adapter in seconds

`char * service`–The service parameter of the TIBCO Rendezvous transport that is used internally

`char * network`– The network parameter of the TIBCO Rendezvous transport that is used internally

`char * daemon`– The daemon parameter of the TIBCO Rendezvous transport that is used internally

`unsigned long agentTimeout`– The timeout of the agent in seconds

`char * subject`– The message subject on which the message is to be sent from the Adapter Agent to the adapter

`void ** replyBuf`– The address of the reply `CArray` buffer. The replyBuf must be allocated by `tpalloc`

`long sendlen`–The length of the `CArray` buffer to be sent

`unsigned int* rcvlen`—The address of a integer to return the length of `replyBuf`

- **Return Value**

The method returns 1 if successful, and 0 if it failed. Error messages are written into the Tuxedo ULOG file.

## TIBCO EMS Adapter Agent Methods

---

The following adapter agent methods use TIBCO EMS. The header file relating to them is `adapteragentJms.h` and the link library relating to them is `agentJMS.lib.lib` for Microsoft Windows or `libagentJMS.so/libagentJMS.sl` for Unix.

### The callPubJMS Method

- Function

```
int callPubJMS(char *serviceName, char *sendbuf, char *serverUrl, char
*jmsType, char *destination, char *userName, char *password)
```

- Description

This method is for JMS Adapter Agent-based publication services. The user's Tuxedo service invokes the method to send the request which uses FML32/STRING buffers to TIBCO EMS.

- Parameters

`char * serviceName`—The name of the adapter service to be used

`char * sendbuf`—The request buffer of the type FML32/STRING

`char * serverUrl`— The EMS server location with URL format, and the format is "<protocol>://<host-name>:<port-number>", for example: "tcp://myhost:7222". Please see EMS documents for further details.

`char * jmsType`— The JMS transport type. The value must be equal with "queue" or "topic".

`char * destination`— The JMS destination. the queue name or the topic name.

`char * userName`— The user name of the EMS server

`char * password`— The password of the EMS server

- Return Value

The method returns 1 on success, and 0 on failure. Error messages are written into the Tuxedo ULOG file.

### The callArrayPubJMS Method

- Function



```
int callCArrayPubJMS(char *serviceName, char *sendbuf, char *serverUrl,
char *jmsType, char *destination, char *userName, char *password, unsigned
int sendlen)
```

- Description

This method is for JMS Adapter Agent-based publication services. The user's Tuxedo service invokes the method to send the request which uses CArray buffers to TIBCO EMS.

- Parameters

`char * serviceName`—The name of the adapter service to be used

`char * sendbuf`—The request buffer of type CArray

`char * serverUrl`— The EMS server location with URL format, and the format is "<protocol>://<host-name>:<port-number>", for example: "tcp://myhost:7222". Please see EMS documents for further details.

`char * jmsType`— The JMS transport type. The value must be equal with "queue" or "topic".

`char * destination`— The JMS destination. the queue name or the topic name.

`char * userName`— The user name of the EMS server

`char * password`— The password of the EMS server

`unsigned int sendlen`— The length of sendbuf

- Return Value

The method returns 1 on success, and 0 on failure. Error messages are written into the Tuxedo ULOG file.



## Appendix E **FAQs and Troubleshooting**

This appendix contains answers to some frequently asked questions and some common errors along with their causes and solutions.

### Topics

---

- [Frequently Asked Questions, page 186](#)
- [Troubleshooting, page 187](#)

## Frequently Asked Questions

---

### Can I bring up TIBCO Designer from a UNIX command-line?

No. TIBCO Designer is a GUI based tool and a UNIX GUI environment is mandatory to run it. It cannot be brought up from a terminal.

### When starting the adapter, what if the repository is not found?

Start the repository server before starting the adapter. If you are starting a remote repository, ensure that TIBCO Administrator is installed on the remote location. Ensure that a properly configured `.dat` file is available in the path specified (local or remote). Ensure that the `repourl` has been specified accurately in the adapter's `.tra` file.

### When saving an adapter configuration to the project, if an error occurs where is it logged?

TIBCO Designer error messages are logged to the files `stderr.log` and `designer.log` under the `TIBCO_HOME\Designer\<ver>\logs` directory.

### When an error occurs in a Subscription Service adapter service, where is it displayed?

Errors that occur in a request-response operation are sent to the client. Errors that occur in a subscription operation are logged to a trace file. The log file path and name is set in the `.tra` file corresponding to the adapter instance. All logs are sent to `TIBCO_HOME\adapter\adutxedo\veriosn_number\logs` unless otherwise specified.

### Is “-” character allowed in the machine name parameter specified in the UBBCONFIG file?

No, Tuxedo does not allow the “-” (hyphen) character for machine name parameter specified in the `UBBCONFIG` file.

### Where to find the Tuxedo .h files?

Any FML32 buffer type will have a corresponding field file which needs to be converted to a `.h` file before configuring the service for it. To convert the field file to the `.h` file, the `tuxdev.exe` which is available under the *Tuxedo Installation Directory\Tuxedo\bin* is to be used.

## Troubleshooting

---

### The adapter startup fails

Either the repository file (.dat) is not found or the .dat file is not properly configured.

#### Solution

- Ensure that the `repourl` syntax has been specified accurately in the adapter's .tra file.
- Ensure that the path specified for the .tra file is correct.

### The adapter startup fails after specifying the appropriate DAT file

You must start the repository server before starting the adapter.

If it is a remote repository, ensure that the `repourl` syntax has been specified accurately in the .tra file. Ensure that the path specified for the .tra file is correct.

### The adapter fails to respond to a request

The subject name may be inconsistent.

#### Solution

Ensure that the subject name to which the adapter listens is the same as the subject name of the client.

### The adapter fails to respond to a request after successfully receiving it

The adapter may fail to respond due to various reasons, such as:

- errors resulting from class mismatch
- records not available in the target application
- connectivity problems with the target application

### When saving an adapter configuration to a project, an error occurs indicating network problems

Check if the `TIBCO_HOME\TRA\bin` directory is present in your PATH settings.

**The adapter return the following error message even though the adapter has initialized successfully**

`tpcall/tpacall/tppost/tpenqueue failed, TPESYSTEM Internal System Error`

Check if the Tuxedo service is running and all parameters have been specified correctly in the incoming message.

**Stopping the adapter sometimes results in a coredump on UNIX**

This can happen if the connection to Tuxedo server is not established properly.

**Solution**

Leave the adapter idle for about a minute, so that the connection to Tuxedo is established and then stop the adapter.

**While trying to create executables for Tuxedo client or server programs using the buildclient or buildserver utilities, the error 'cannot execute cl.exe' occurs**

This may be due to compilation errors in the client or server programs.

**Solution**

Check the console for detailed error messages. If there are no compilation issues, check the environment variable settings.

**While running Tuxedo's tmloadcf program to compile the UBBCONFIG file, the error 'cannot run on non-master node' occurs**

Check the machine name in the UBBCONFIG file. Make sure the case is specified correctly, since this value is case-sensitive.

**While running Tuxedo's tmloadcf program to compile the UBBCONFIG file, the error 'cannot run on active node' occurs**

The `tmloadcf` program cannot run if an application is booted. Run Tuxedo's `tmshutdown` program to shutdown the Tuxedo servers before running `tmloadcf`.

**Running as a Subscriber or RPC server returns the following error 'tpcall/tpacall/tpenqueue failed, TPENOENT - no entry found'**

Check if the service name, specified in the message sent to the adapter, is correct. Ensure that the service is currently running. Tuxedo's `tmadmin` tool allows you to examine available services using the `psc` command.

**While running Tuxedo's tmboot program to boot the Tuxedo services, the error 'No BBL available on site <site name> Will not attempt to boot server processes on that site.' occurs**

Check if the transaction log (TLOG device) was created using the `crdl` command.

### The following error occurs

```
021722.MAGIC!tmboot.295.515.-2: 10-07-2002: Tuxedo Version 8.0
32-bit Windows.
021722.MAGIC!tmboot.295.515.-2: FATAL: internal error:
CMDTUX_CAT:4578: ERROR: Unlicensed System Binary
```

If you chose not to install your software license when you installed your software, you can install the license later, using the Oracle License Utility. To install your Oracle Tuxedo product license on a Microsoft Windows system, try the following procedure instead of reinstalling:

- Transfer the license file `lic.txt` to your machine.
- Enter the drive and location of the `lic.txt` file and click **OK**. The Oracle license utility installs `lic.txt` in the `TUXDIR/udataobj` directory, where `TUXDIR` represents the product directory in which you installed the Oracle Tuxedo software.



As an alternative to using the Oracle license utility to install your product license, you may manually copy `lic.txt` to the `TUXDIR/udataobj` directory.





# Index

## A

Adapter Instance Tabs  
 Adapter Services [30](#)  
 Configuration [26](#)  
 General [31](#)  
 Logging [33](#)  
 Monitoring [36](#)  
 MultiThreading [32](#)  
 Run-time Connection [28](#)  
 Startup [36](#)  
 Application Password [29](#)  
 Available Microagents [99](#)

## C

Class Microagent Name field, adapter [37](#)  
 Client Name [29, 30](#)  
 clientvar property [138](#)  
 command line arguments [114](#)  
 Configuration Tasks [22](#)  
 Configuring a Remote Adapter [86](#)  
 Connection Factory Type [42](#)  
   Queue [42](#)  
   Topic [42](#)  
 Connection Management [23](#)  
 Create Tuxedo Configuration Files [5](#)  
 customer support [xvii](#)

## D

Defining a TIBCO Hawk Session [80](#)

Delivery Mode [41](#)  
   Durable [41](#)  
   Non-durable [42](#)  
   Non-Persistent [41](#)  
   Persistent [41](#)

## E

Edit the ubbconfig File [4](#)  
 Encoding [32](#)  
 Endpoint Reference [51](#)  
 ENV\_HOME [xiv](#)  
 Examples [79](#)

## F

Frequently Asked Questions [186](#)

## G

global variables [81](#)  
   setting for monitoring [36](#)  
 Group Name [29](#)  
 Guidelines for Choosing an Instance Name [27](#)

## I

Initialization Flags [29, 30](#)  
 Initialization Flags Values [30](#)  
 Installing the Adapter as a Service on Microsoft  
   Windows [79](#)

Instance Name [27](#)  
 Internationalization [24](#)  
 Invoking Microagent Methods [97](#)

## L

load balancing [75](#)  
 Local Repository Locator String [79](#)  
 local repository locator string [79](#)  
 locator string [79](#)  
 Log File field, adapter [33](#)  
 Log to Standard field, adapter [33](#)

## M

Message Transports [40](#)  
 Microagent Session field, adapter [37](#)  
 Multithreading [23](#)

## O

obfuscation [137](#)  
 Other TIBCO Product Documentation [xii](#)

## P

password property [138](#)  
 Predefined Global Variables [83](#)  
 Preparing Tuxedo [2](#)  
 preRegisterListener [126](#)  
 Publication Service Tabs  
   Configuration [44](#)  
   Fetch Schema [46](#)  
   Publisher Options [45](#)  
   Publisher Schema View [47](#)  
   Schema [48](#)  
   Transport [48](#)

## Q

QoS  
   Certified [40](#)  
   Distributed Queue [41](#)  
   Reliable [40](#)  
 Quality of Service [40](#)

## R

Related Documentation [xii](#)  
 Request-Response Invocation Service Tabs  
   Client Options Tab [67](#)  
   Configuration [65](#)  
   Fetch Schema [68](#)  
   Reply Schema View [69](#)  
   Request Schema View [69](#)  
   Schema [70](#)  
   Transport [70](#)  
 Request-Response Service Additional Fields [174](#)  
 Request-Response Service Tabs  
   Configuration [58](#)  
   Fetch Schema [61](#)  
   Reply Schema View [62](#)  
   Request Schema View [61](#)  
   RPC Server Options [60](#)  
   Schema [62](#)  
   Transport [62](#)

## S

Saving the Project [24](#)  
 Session Reference [51](#)  
 setting global variables [81](#)  
 Show Startup Banner [36](#)  
 Specifying Variables Using TIBCO Designer [81](#)  
 Starting and Stopping the Run-time Adapter [79](#)  
 Starting TIBCO Hawk Software [95](#)  
 Status Messages [147](#)  
 Subscription Service Additional Fields [172](#), [172](#)

## Subscription Service Tabs

Configuration [51](#)Fetch Schema [54](#)Reply Schema View [55](#)Schema [55](#)Subscriber Options [53](#)Subscriber Schema View [54](#)Transport [55](#)support, contacting [xvii](#)**T**technical support [xvii](#)Testing the Adapter [25](#)The Auto-Discovery Process [96](#)Third-Party Documentation [xiii](#)

## TIBCO Hawk

background information [94](#)enterprise monitor components [94](#)

## TIBCO Hawk methods

`_onUnsolicitedMsg` [103](#)`activateTraceRole` [104](#)`deactivateTraceRole` [105, 105](#)`getActivityStatistics` [106](#)`getActivityStatisticsByOperation` [107](#)`getActivityStatisticsBySchema` [111, 111](#)`getActivityStatisticsByService` [109](#)`getAdapterServiceInformation` [112](#)`getComponents` [113](#)`getConfig` [114](#)`getConfigProperties` [115](#)`getConnectionStatistics` [116](#)`getHostInformation` [117](#)`getPerfMonSetting` [118](#)`getPollingInterval` [119](#)`getQueueStatistics` [120](#)`getRvConfig` [121, 121](#)`getStatus` [122, 122](#)`getThreadStatistics` [123](#)`getTraceSinks` [124](#)`getVersion` [125](#)`preRegisterListener` [126](#)`resetActivityStatistics` [127](#)`resetConnectionStatistics` [128](#)`resetThreadStatistics` [129](#)`reviewLedger` [130, 130](#)`setPollingInterval` [131](#)`setTraceSinks` [132](#)`stopApplicationInstance` [133](#)`unRegisterListener` [134](#)

TIBCO Rendezvous, retrieving configuration through

TIBCO Hawk [121](#)TIBCO\_HOME [xiv](#)Trace Message Fields [145](#)Tracing [144](#)Tracing Levels and Fields [145, 145](#)Troubleshooting [187](#)**U**URL [79](#)username property [138](#)

Using Global Variables [81](#)

## V

Variable Specification [81](#)

variables, global [81](#)

## W

Wire Formats [42](#)

    ActiveEnterprise Message [43](#)

    Rendezvous Message [42](#)

    XML Message [42](#)