

# TIBCO Silver<sup>®</sup> Fabric Enabler for TIBCO<sup>®</sup> API Exchange Gateway

## User's Guide

*Software Release 1.1  
July 2014*

**Two-Second Advantage<sup>®</sup>**



## Important Information

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

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

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

TIBCO, Two-Second Advantage, TIBCO Silver, TIBCO Silver Fabric, TIBCO API Exchange Gateway, TIBCO Rendezvous, TIBCO Enterprise Message Service, TIBCO Runtime Agent, and TIBCO Hawk are either registered trademarks or trademarks of TIBCO Software Inc. in the United States and/or other countries.

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

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

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

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

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

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

Copyright © 2014 TIBCO Software Inc. ALL RIGHTS RESERVED.

TIBCO Software Inc. Confidential Information

# Contents

Related Documentation .....	iv
TIBCO Silver® Fabric Enabler for TIBCO® API Exchange Gateway Documentation .....	iv
Other TIBCO Product Documentation .....	iv
Typographical Conventions .....	v
Connecting with TIBCO Resources .....	vii
How to Join TIBCOcommunity .....	vii
How to Access All TIBCO Documentation .....	vii
How to Contact TIBCO Support .....	vii
Product Overview .....	2
Main Functionalities .....	2
Overview .....	4
Components .....	4
Creating TIBCO® API Exchange Gateway Components .....	5
Creating a Stack .....	27
TIBCO Silver Fabric Engine Statistics .....	29
Running a Stack .....	31
Updating a Stack .....	32



# Preface

## Topics

---

- [Related Documentation, page iv](#)
- [Typographical Conventions, page v](#)
- [Connecting with TIBCO Resources, page vii](#)

## Related Documentation

---

This section lists documentation resources you may find useful.

It is assumed that you are very familiar with the installation, configuration, and use of TIBCO® API Exchange Gateway because you are seeking to scale this product using TIBCO Silver Fabric. The following resources will prove useful for this endeavor.

The following documents form the TIBCO® API Exchange Gateway documentation set:

- *TIBCO® API Exchange Gateway Installation Guide*  
Read this manual for instructions on site preparation and installation.
- *TIBCO® API Exchange Gateway User's Guide*  
Read this manual for instructions on using the product.
- *TIBCO® API Exchange Gateway 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.

## TIBCO Silver® Fabric Enabler for TIBCO® API Exchange Gateway Documentation

The following documents form the TIBCO Silver® Fabric Enabler for TIBCO® API Exchange Gateway documentation set:

- *TIBCO Silver® Fabric Enabler for TIBCO® API Exchange Gateway Installation*  
Read this manual for instructions on site preparation and installation.
- *TIBCO Silver® Fabric Enabler for TIBCO® API Exchange Gateway User's Guide*  
Read this manual for instructions on using the product.
- *TIBCO Silver® Fabric Enabler for TIBCO® API Exchange Gateway 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

TIBCO Silver® Fabric Enabler for TIBCO® API Exchange Gateway empowers TIBCO Silver® Fabric private cloud infrastructure to run TIBCO® API Exchange Gateway.

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

- TIBCO Silver<sup>®</sup> Fabric
- TIBCO Rendezvous<sup>®</sup>
- TIBCO Enterprise Message Service<sup>™</sup>

## Typographical Conventions

---

The following typographical conventions are used in this manual.

Table 1 General Typographical Conventions

Convention	Use
<i>TIBCO_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 default value of <i>TIBCO_HOME</i> depends on the operating system. For example, on Windows systems, the default value is <code>C:\tibco</code>.</p> <p>Other TIBCO products are installed into an <i>installation environment</i>. Incompatible products and multiple instances of the same product are installed into different installation environments. An environment home directory is referenced in documentation as <i>ENV_HOME</i>. The default value of <i>ENV_HOME</i> depends on the operating system. For example, on Windows systems, the default value is <code>C:\tibco</code>.</p> <p>TIBCO Silver<sup>®</sup> Fabric Enabler for TIBCO<sup>®</sup> API Exchange Gateway is installed into a directory that is referenced in documentation as <i>SFAX_HOME</i>. The value of <i>SFAX_HOME</i> depends on the operating system. For example, on Windows systems, the default value is <code>C:\tibco\sfax</code>.</p> <p>TIBCO Silver Fabric is installed into a directory that is referenced in documentation as <i>SILVERFABRIC_HOME</i>. The value of <i>SILVERFABRIC_HOME</i> depends on the operating system. For example, on Windows systems, the default value can be <code>C:\fabric</code>.</p>
<i>ENV_HOME</i>	
<i>SFAX_HOME</i>	
<i>SILVERFABRIC_HOME</i>	
code font	<p>Code font identifies commands, code examples, filenames, pathnames, and output displayed in a command window. For example:</p> <p>Use <code>MyCommand</code> to start the foo process.</p>

Table 1 General Typographical Conventions (Cont'd)

Convention	Use
<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® API Exchange Gateway User's Guide</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>
Key combinations	<p>Key names 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>
	<p>The note icon indicates information that is of special interest or importance, for example, an additional action required only in certain circumstances.</p>
	<p>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.</p>
	<p>The warning icon indicates the potential for a damaging situation, for example, data loss or corruption if certain steps are taken or not taken.</p>

## Connecting with TIBCO Resources

---

### How to Join TIBCOmmunity

TIBCOmmunity is an online destination for TIBCO customers, partners, and resident experts, a place to share and access the collective experience of the TIBCO community. TIBCOmmunity offers forums, blogs, and access to a variety of resources. To register, go to <http://www.tibcommunity.com>.

### How to Access All TIBCO Documentation

All TIBCO product documentation is available from <https://docs.tibco.com>

TIBCO Silver<sup>®</sup> Fabric Enabler for TIBCO<sup>®</sup> API Exchange Gateway documentation is here:

<https://docs.tibco.com/products/tibco-silver-fabric-enabler-for-tibco-api-exchange-gateway-1-0-0>

### 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 **Introduction**

This chapter briefly introduces this product.

### Topics

---

- [Product Overview, page 2](#)

## Product Overview

---

TIBCO Silver Fabric Enabler for TIBCO API Exchange Gateway is a complementary software component. It allows the TIBCO API Exchange Gateway Configuration UI and Engines to be published in cloud environments using TIBCO Silver<sup>®</sup> Fabric and leveraging TIBCO Silver<sup>®</sup> Fabric capabilities.

### Main Functionalities

TIBCO Silver Fabric Enabler for TIBCO API Exchange Gateway provides the following main functionality:

- Quickly configure, publish, and launch one or multiple TIBCO<sup>®</sup> API Exchange Gateway Configuration GUI and Processing Units (Engines and Agents) to leverage features provided by the TIBCO Silver Fabric Cloud.
  - TIBCO<sup>®</sup> API Exchange Gateway - Configuration GUI (asg-configui)
  - TIBCO<sup>®</sup> API Exchange Gateway - Core Engine (asg-core)
  - TIBCO<sup>®</sup> API Exchange Gateway - Core Engine (asg-core) with the Cache Agent Enabled (asg-caching-core)
  - TIBCO<sup>®</sup> API Exchange Gateway - Global Throttle Manager (asg-gtm)
  - TIBCO<sup>®</sup> API Exchange Gateway - Cache Agent (asg-cache)
  - TIBCO<sup>®</sup> API Exchange Gateway - Cleanup Agent (asg-cache-cleanup)
  - TIBCO<sup>®</sup> API Exchange Gateway - APIX Central Logger (asg-cl)
- Run one or multiple TIBCO API Exchange Core Engines on the same TIBCO Silver Fabric Engine.
- For testing purposes, you can publish and install TIBCO API Exchange Gateway Engines without running any processes to manually test your configurations.
- Ensures that publishing of Components and Stacks follow a set of recommended and supported TIBCO practices to implement load balancing, software updates, and Stack updates.
- It collects API Exchange Gateway metrics to be utilized for TIBCO Silver<sup>®</sup> Fabric rules based scaling. Those metrics can be used to trigger scaling up of the number of API Exchange Gateway engines required to process the workload and to scale down the number of machines used when demand for resources drops providing elasticity and optimization of computing resources.

## Chapter 2

# Creating a TIBCO Silver Fabric Enabler for TIBCO API Exchange Gateway Stack

Configure and publish Components based on TIBCO Silver Fabric Enabler for TIBCO API Exchange Gateway.

## Topics

---

- [Overview, page 4](#)
- [Creating TIBCO® API Exchange Gateway Components, page 5](#)
- [Creating a Stack, page 27](#)
- [TIBCO Silver Fabric Engine Statistics, page 29](#)
- [Running a Stack, page 31](#)
- [Updating a Stack, page 32](#)

## Overview

---

A TIBCO Silver Fabric Enabler for TIBCO API Exchange Gateway is an entity that runs inside TIBCO Silver<sup>®</sup> Fabric. It enables creation of specifically configured Components that may be published in a Stack on one or many TIBCO Silver Fabric Engines.

## Components

TIBCO Silver Fabric Enabler for TIBCO API Exchange Gateway consists of two gridlib archives installed on the TIBCO Silver Fabric Broker that create a TIBCO Silver<sup>®</sup> Fabric Administrator Component Wizard interface.

The Component Wizard to create a new TIBCO<sup>®</sup> API Exchange Gateway Component enables configurations specifying how the TIBCO<sup>®</sup> API Exchange Gateway Distribution for TIBCO Silver<sup>®</sup> Fabric and the TIBCO Rendezvous<sup>®</sup> Distribution for TIBCO Silver<sup>®</sup> Fabric will be published and run on the TIBCO Silver Fabric Engines.

To build and run a Stack with TIBCO Silver Fabric Enabler for TIBCO API Exchange Gateway Stack, you need to perform the following tasks:

- Create and publish one or more TIBCO<sup>®</sup> API Exchange Gateway Components. Refer to [Creating TIBCO<sup>®</sup> API Exchange Gateway Components on page 5](#).
- Create a Stack with one or several TIBCO<sup>®</sup> API Exchange Gateway Components. Refer to [Creating a Stack on page 27](#).

After completing these tasks, you can run and later update Components and Stacks that were created using the TIBCO Silver Fabric Enabler for TIBCO API Exchange Gateway. See [Updating a Stack on page 32](#) for information on how to update a Stack.

## Creating TIBCO® API Exchange Gateway Components

---

Configuration of the Component created with TIBCO Silver Fabric Enabler for TIBCO API Exchange Gateway determines how the Distribution is published to the Engines.

TIBCO® API Exchange Gateway Distribution for TIBCO Silver® Fabric contains the expanded product installation used to publish and run any and all of the following:

- TIBCO® API Exchange Gateway - Configuration GUI (asg-configui)
- TIBCO® API Exchange Gateway - Core Engine (asg-core)
- TIBCO® API Exchange Gateway - Core Engine (asg-core) with the Cache Agent Enabled (asg-caching-core)
- TIBCO® API Exchange Gateway - Global Throttle Manager (asg-gtm)
- TIBCO® API Exchange Gateway - Cache Agent (asg-cache)
- TIBCO® API Exchange Gateway - Cleanup Agent (asg-cache-cleanup)
- TIBCO® API Exchange Gateway - APIX Central Logger (asg-cl)

A limited set of installation configurations are presented by the GUI of the TIBCO Silver Fabric Component Wizard, but you can pass any command line options exposed by TIBCO® API Exchange Gateway Engine or upload a startup property file (TRA file) with your custom configurations.

To create and configure a component, perform the relevant tasks according to the type of TIBCO® API Exchange Gateway Engine you will create:

- [Task A, Create a TIBCO® API Exchange Gateway Component, page 7](#)
- [Task B, Set Enabler to launch Configuration User Interface or Engine, page 9](#)
- [Task C, Configure TIBCO® API Exchange Gateway Configuration UI, page 11](#)
- [Task D, TIBCO API Exchange Gateway Configuration UI and Core Engine Settings, page 12](#)
- [Task E, SOAP/JMS and ESB for Façade and Target Transport Messaging, page 16](#)
- [Task F, Log Reporting Messaging, page 18](#)
- [Task G, Configure the Parameters Used to Launch asg-engine, page 20](#)
- [Task H, TIBCO API Exchange Gateway - Core Engine Settings, page 22](#)
- [Task I, Configure TIBCO® API Exchange Gateway Running Conditions, page 22](#)

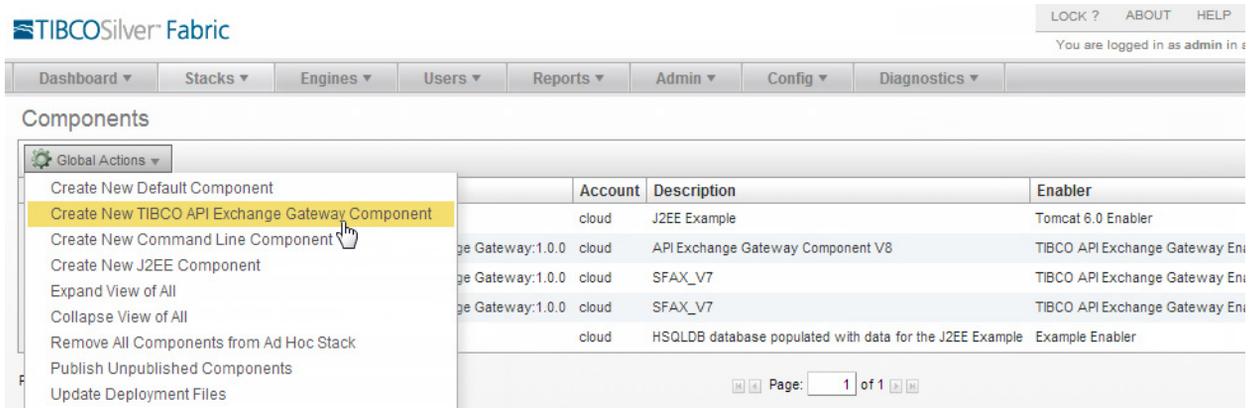
- [Task K, Add or Edit Enabler-specific Runtime Context Variables \(optional\)](#)
- [Task L, Add Allocation Rule Settings, page 24](#)
- [Task M, Finish Configuring the Component, page 25](#)

Publish the TIBCO® API Exchange Gateway component after you complete these tasks and at runtime, access TIBCO Configuration GUI, using the TIBCO Silver Fabric VirtualRouter address: `http://Virtualrouter:9200/ConfigUI/` or directly with the Engine URL: `http://SilverFabricEngine:9200/ConfigUI/`

## Task A Create a TIBCO® API Exchange Gateway Component

1. In the TIBCO Silver® Fabric Administration Tool, select **Stacks > Components**.
2. On the Components page, select **Create New TIBCO® API Exchange Gateway** from the Global Actions drop-down list as shown below.

Figure 1 Creating a TIBCO® API Exchange Gateway Component



3. Provide a name and a description for the Component.

Figure 2 Configuring General Properties

The screenshot shows a configuration dialog box titled 'TIBCO API Exchange Gateway'. Below the title is the text 'Configure general properties'. There are two input fields: 'Name (required)' with the value 'APIExG\_Configuration UI' and 'Description' with the value 'API Exchange Gateway ConfigUI'. At the bottom of the dialog, there are five buttons: 'Cancel', 'Previous', 'Menu', 'Next' (which is highlighted in blue), and 'Finish'.

4. Choose the TIBCO product distribution versions to publish. Refer to [Figure 3](#). Select a version of each of the installed distributions. It is generally a good practice to use the latest version of a product distribution.

Refer to the *TIBCO Silver*<sup>®</sup> Fabric Enabler for TIBCO<sup>®</sup> API Exchange Gateway *Readme* for more information on supported distributions.

If there are multiple product distribution versions which are displayed then they are compatible with the component being defined.

Figure 3 Choosing TIBCO Product Distribution Versions



Make sure that the *TIBCO\_RV\_distribution* is present as is shown above. TIBCO Rendezvous must be published with the TIBCO<sup>®</sup> API Exchange Gateway Distribution for TIBCO Silver<sup>®</sup> Fabric to provide messaging.

5. Click the **Next** button TIBCO API Exchange Gateway: Configuration User Interface or Engine panel is displayed as is shown next in [Task B, Set Enabler to launch Configuration User Interface or Engine, on page 9](#).

## Task B Set Enabler to launch Configuration User Interface or Engine

The TIBCO® API Exchange Gateway: Configuration User Interface or Engine page and each of the configuration parameters are described here. Enter appropriate values for each field as applicable.

Figure 4 TIBCO API Exchange Gateway: Configuration User Interface or Engine page

TIBCO API Exchange Gateway: APIExG\_Configuration UI

TIBCO API Exchange Gateway: Configuration User Interface or Engine

**Enabler will launch** Engine

**Configuration directory path** (When empty the location is the value of the runtime variable TIBCO\_HOME\_CFGMGMT\_DIR (set to \${ENGINE\_WORK\_DIR}/cfg\_data))

**Change parameters**

**Logging Level** Info (1) ▼

**Facade HTTP Channel Base Port** (Value will be increased by the Engine instance to avoid port conflict. To disable it set the variable APIX\_FACADE\_HTTP\_BASE\_PORT to Incremental to None.) 9222

**Facade HTTPS Channel Base Port** (Value will be increased by the Engine instance to avoid port conflict. To disable it set the variable APIX\_FACADE\_HTTPS\_BASE\_PORT to Incremental to None.) 9233

**OAuth HTTP Channel Base Port** (Value will be increased by the Engine instance to avoid port conflict. To disable it set the variable APIX\_OAUTH\_HTTP\_BASE\_PORT to Incremental to None.) 9322

**OAuth HTTPS Channel Base Port** (Value will be increased by the Engine instance to avoid port conflict. To disable it set the variable APIX\_OAUTH\_HTTPS\_BASE\_PORT to Incremental to None.) 9333

Cancel Previous Menu Next Finish

- **Enabler will launch** - The "Enabler will launch" parameter determines whether this component will publish and install the TIBCO® API Exchange Gateway Configuration UI or run one of its Engines or Agents. Another option allows for manual testing of an Engine or Agent after configuration and installation.
  - **Configuration UI** - enables component configuration of the TIBCO® API Exchange Gateway User Interface.
  - **Engine** - enables component configuration of the TIBCO® API Exchange Gateway Engines/Agents and directly running that Engine/Agent when published and run in a Stack.
    - Core Engine (asg-core)
    - Core Engine in Cache Agent Enabled mode (asg-caching-core)
    - Global Throttle Manager (asg-gtm)
    - Cache Agent (asg-cache)
    - Cache Cleanup Agent (asg-cache-cleanup)

Central Logger with Database (asg-cl)

- **Install and configure without running to allow manual test of engine parameters** - enables configuration and publishing of the TIBCO® API Exchange Gateway Engine to the TIBCO Silver Fabric Engine without automatically running the published Engine or Agent.



Because you can install this product without an EAR file to run, it is better to suspend the verification polling that would repeatedly attempt restarting the Engine in that case.

The selected **Enabler will launch** option determines what Component Wizard pages are displayed next.

- **Configuration directory path** - When empty the location is the value of the runtime variable: `TIBCO_HOME_CFGMGMT_DIR` (set to `${ENGINE_WORK_DIR}/cfg_data` To set a value other than the default simply enter a relative directory path such as: `usr/myname/someDirectory`
- **Change parameters** - When this checkbox is selected the Logging Level and the HTTP Channel Base Port parameter values will apply. When this checkbox is not checked these parameter values will not be changed regardless of what values are input in the HTTP Channel Base Port fields below.
- **Logging Level** - The logging level can be set to Debug (0), Info (1), Warning (2), Error (3), No Logging (4).
- **Façade HTTP Channel Base Port / Façade HTTPS Channel Base Port** - The client facing HTTP /HTTPS (SSL) base port starting point. This value is used as the reference so additional Engine instances derived from the same Component on the same machine get incremented HTTP/HTTPS port values to avoid port conflict. Refer to the *TIBCO API Exchange Gateway User's Guide* for more information about use of the ports.
- **OAuth HTTP Channel Base Port / OAuth HTTPS Channel Base Port** - TIBCO API Exchange Gateway OAuth Server HTTP /HTTPS (SSL) base port starting point. This value is used as the reference so additional Engine instances derived from the same Component on the same machine get incremented OAuth HTTP/HTTPS port values to avoid port conflict. Refer to the *TIBCO API Exchange Gateway User's Guide* for information about use of OAuth Server.

The next Component Wizard page displayed depends on the selection chosen in the "Enabler will launch" drop down selector.

For Components that will instantiate TIBCO® API Exchange Gateway Configuration UI see [Configure TIBCO® API Exchange Gateway Configuration UI on page 11](#)

For Components that will instantiate a TIBCO® API Exchange Gateway Engine skip to [Configure the Parameters Used to Launch asg-engine on page 20](#)

For Components that will instantiate TIBCO® API Exchange Gateway Engines without running them automatically skip to [Task I, Configure TIBCO® API Exchange Gateway Running Conditions, page 22](#)

### Task C Configure TIBCO® API Exchange Gateway Configuration UI

The TIBCO API Exchange Gateway: User Interface Configuration page enables specification of Configuration UI administrator user name, password, and the HTTP Base Port.

Figure 5 TIBCO API Exchange Gateway: User Interface Configuration

The HTTP Base Port setting will be the port value of the first instance of TIBCO API Exchange Gateway User Interface installed. To avoid HTTP port conflicts when more than one TIBCO® API Exchange Gateway application is running on the same machine, TIBCO Silver® Fabric Enabler for TIBCO® API Exchange Gateway automatically increments the port as follows:

$$\text{HTTP Base Port} + \text{EngineInstanceValue}$$

Where the *EngineInstanceValue* is the Engine number where the application is running. For example on a dual-core machine, the default number of Engines is two (one per core). The first *EngineInstanceValue* will have a value of zero, the second will have a value of one.

Continue with the configuration of [TIBCO API Exchange Gateway Configuration UI and Core Engine Settings on page 12](#)

## Task D TIBCO API Exchange Gateway Configuration UI and Core Engine Settings

Set up the Configuration UI Server or the Core Engine for reporting, facade and target JMS transport messaging, JMS delivery mode and JMS acknowledgement mode.

**Enable Reporting** - Enable reporting (event messaging) with a Central Logger. When the Enable Reporting checkbox is selected a Central Logger Component must be set as a dependency for the Configuration UI Component and it must be present in the Stack.



TIBCO Silver Fabric Enabler will perform a dependency check at runtime for a Central Logger Component and if it is not present the component will not run and it will report the failure in the Fabric Engine log.

When this is checked the Component Wizard display another page to select either TIBCO Rendezvous or JMS for the "Log Reporting Messaging Transport Type".

Figure 6 Core Engine Settings "Set up the core engine for API Exchange component" page

**Enabler Facade/Target JMS Transport** - Leave this checkbox unselected to use TIBCO Rendezvous as the default messaging transport between component servers. When Rendezvous is the preferred messaging transport no further messaging configuration is necessary.

Select the checkbox to enable and configure JMS transport as SOAP/JMS or JMS/XML as ESB channels for the façade and target messaging transports.

Enabling use of Facade/Target JMS Transport will expose JMS Delivery Mode options, JMS Acknowledgement Mode options, and four additional Component Wizard configuration pages for configuring SOAP/JMS Facade Transport, ESB Facade Transport, SOAP/JMS Target Transport, and ESB Target Transport.

**JMS Delivery Mode** - Select a JMS delivery mode for TIBCO API Exchange Gateway servers. The delivery mode may be `NON_PERSISTENT`, `PERSISTENT`, or `RELIABLE` as is described in the table below:

Table 2: *JMS Message Delivery Modes*

Mode	Description
<code>NON_PERSISTENT</code>	Message is not persisted on the disk/database by the server, so in-transit messages may be lost when the server is restarted.
<code>PERSISTENT</code>	Ensures the delivery of messages to the destination on the server in almost all circumstances. This is applicable when a producer sends a <code>PERSISTENT</code> message as the producer waits for the server to reply with a confirmation. The message is persisted on disk by the server.
<code>RELIABLE</code> (requires use of TIBCO Enterprise Message Service)	<ul style="list-style-type: none"> <li>Reliable delivery mode is an extension of the JMS standard delivery modes.</li> <li>When this delivery mode is used, it offers increased performance of the message producers.</li> </ul>

**JMS Acknowledgment Mode** - consult the following table

Table 3: *JMS Message Acknowledgement Modes*

Mode	Description
<code>AUTO_ACKNOWLEDGE</code>	Specifies that the session is to automatically acknowledge consumer receipt of messages when message processing has finished.
<code>CLIENT_ACKNOWLEDGE</code>	Specifies that the consumer is to acknowledge all messages delivered in this session. With this acknowledgment mode, the client acknowledges a consumed message by calling the message's <code>acknowledge</code> method.

Table 3: JMS Message Acknowledgement Modes

Mode	Description
DUPS_OK_ACKNOWLEDGE	<p>Specifies that the session is to "lazily" acknowledge the delivery of messages to the consumer. "Lazy" means that the consumer can delay acknowledgement of messages to the server until a convenient time; meanwhile the server might redeliver messages. This mode reduces session overhead. However, should JMS fail, the consumer may receive duplicate messages.</p>
NO_ACKNOWLEDGE (requires use of TIBCO Enterprise Message Service)	<p>TIBCO Enterprise Message Service extension of JMS acknowledge modes suppresses the acknowledgement of received messages. After the server sends a message to the client, all information regarding that message for that consumer is eliminated from the server. Therefore, there is no need for the client application to send an acknowledgement to the server about the received message. Not sending acknowledgements reduces message traffic and saves time for the receiver, therefore allowing better utilization of system resources.</p> <p><b>Note</b> Sessions created in NO_ACKNOWLEDGE receipt mode cannot be used to create durable subscribers.</p> <p><b>Note</b> Also, queue receivers on a queue that is routed from another server are not permitted to specify NO_ACKNOWLEDGE mode.</p>
EXPLICIT_CLIENT_ACKNOWLEDGE (requires use of TIBCO Enterprise Message Service)	<ul style="list-style-type: none"> <li>• TIBCO Enterprise Message Service extension to JMS acknowledge modes.</li> <li>• EXPLICIT_CLIENT_ACKNOWLEDGE is like CLIENT_ACKNOWLEDGE except it acknowledges only the individual message, rather than all messages received in the session.</li> <li>• One example of when EXPLICIT_CLIENT_ACKNOWLEDGE would be used is when receiving messages and putting the information in a database. If the database insert operation is slow, you may want to use multiple application threads all doing simultaneous inserts. As each thread finishes its insert, it can use EXPLICIT_CLIENT_ACKNOWLEDGE to acknowledge only the message that it is currently working on.</li> </ul>

Table 3: JMS Message Acknowledgement Modes

Mode	Description
EXPLICIT_CLIENT_DUPS_OK_ACKNOWLEDGE	(requires use of TIBCO EMS)
	<ul style="list-style-type: none"> <li>TIBCO Enterprise Message Service extension to JMS acknowledge modes.</li> <li>EXPLICIT_CLIENT_DUPS_OK_ACKNOWLEDGE mode is like TIBEMS-DUPS-OK-ACKNOWLEDGE except it "lazily" acknowledges only the individual message, rather than all messages received so far on the session.</li> </ul>

Refer to the *TIBCO Enterprise Message Service User's Guide* for more details on the JMS message delivery and acknowledgement modes.

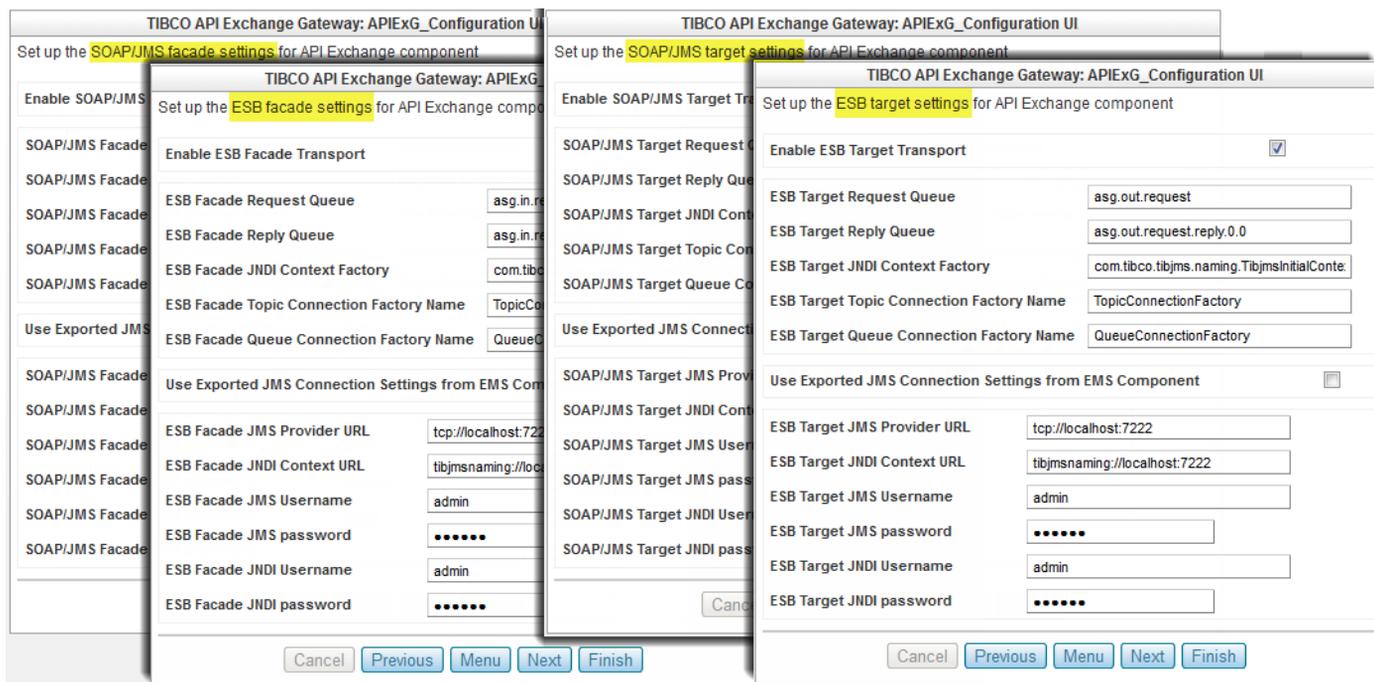
When **Enable Facade/Target JMS Transport** is checked, four additional component configuration pages are added to enable settings for SOAP/JMS and ESD for both Facade and Target transport messaging where it may be applicable for your implementation. Click **Next** to enable and configure those settings.

### Task E SOAP/JMS and ESB for Façade and Target Transport Messaging

Whether you are configuring the TIBCO API Exchange Gateway Component to instantiate the Configuration GUI, Core Engine, Core Engine with the Cache Agent Enabled, Global Throttle Manager, Cache Agent, Cleanup Agent, or the Central Logger... the following Four Component Wizard configuration pages are present in all Component Wizards to enable and configure transport messaging settings between the various component servers for both the Façade and Target. The following pages all have the same parameters and serve to connect TIBCO API Service Gateway Configuration UI with the appropriate messaging transport.

- SOAP/JMS Façade Transport
- ESB Façade Transport
- SOAP/JMS Target Transport
- ESB Target Transport

Figure 7 Similarities between SOAP/JMS and ESB, façade and target settings pages



To avoid repetition of the same parameter descriptions for each page and for each TIBCO API Exchange Gateway Configuration UI or Engine type, parameters will be explained once, for all applicable Component configurations.

**Enable Transport** - You may enable one or many of the transport settings for either the Façade (client request) or Target (data services/sources) using either SOAP/JMS and/or ESB transports. Configuration settings may be modified if you enable a transport for the component.

**Request Queue** - Specifies the queue name for the requests channel communication. An exact match request queue must exist on the JMS provider at the facade/target side when the component is started.

**Reply Queue** - Specifies the queue name for the reply (response) channel communication. An exact match reply (response) queue must exist on the JMS provider at the facade/target side when the component is started.

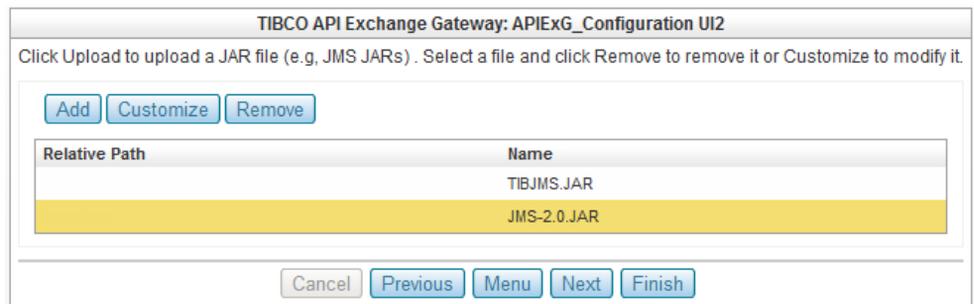
**JNDI Context Factory** - The default value for the initial Java Naming and Directory Interface (JNDI) context factory points to TIBCO Enterprise Message Service. If you're using another JMS provider you can change the JNDI context factory here.

**Topic Connection Factory Name** - Specifies the name of TopicConnectionFactory object stored in JNDI. This object is used to create a topic connection on either the facade or target side depending on which configuration page it is being specified.

**Queue Connection Factory Name** - Specifies the name of QueueConnectionFactory object stored in JNDI. This object is used to create a queue connection with ESB services at either the facade or target side depending on which configuration page it is being specified. The Queue Connection Factory Name specified has to match the Queue Connection Factory Name defined for use in the JMS provider. When using a third-party JMS provider the Queue Connection Factory Name must match in this field and in the JMS provider.

**Use Exported JMS Connection Settings from EMS Component** - If you are using TIBCO Enterprise Message Service you can check this box and skip configuration of the parameters described below. Checking this box will add a Component Wizard page (shown below) to allow you upload the appropriate JAR files (jms-2.0.jar and tibjms.jar for TIBCO EMS) to set and establish connections.

Figure 8 Upload JAR files



**Note:** A relative path can be specified, but it is not necessary.

**JMS Provider URL** - Specifies the connection URL for either the JMS provider or TIBCO EMS used for either the facade or target operation requests.

**JNDI Context URL** - Specifies the URL to the JNDI service provider used for facade or target operation requests within the communication domain.

**JMS Username** - Specifies the user name for logging into the JMS provider or TIBCO EMS in the domain at either the facade or the target side.

**JMS Password** - Specifies the password for logging into the JMS provider

**JNDI Username** - Specifies the user name for logging into the JNDI server in the domain at either the facade or the target side. If the JNDI provider does not require access control, this field can be empty.

**JNDI Password** - Specifies the password for logging into the JNDI server communication domain at the facade or target side. If the JNDI provider does not require access control, this field can be empty.

Refer to the *TIBCO Enterprise Message Service User's Guide* and the *TIBCO API Exchange Gateway User's Guide* for more details.

## Task F Log Reporting Messaging

When the **Enable Reporting** checkbox is selected, this page will appear to enable configuration of the log reporting.

**Log Reporting Messaging Transport Type:** Rendezvous or JMS. If TIBCO Rendezvous is used no further settings are necessary. If your implementation requires use of JMS for reporting, then consider proper values for the parameters.

Figure 9 Log Reporting Messaging

The screenshot displays the configuration interface for log reporting messaging. The title bar reads "TIBCO API Exchange Gateway: APIExG\_Configuration UI". Below the title, the instruction "Set up the log reporting messaging settings for API Exchange component" is shown. The "Log Reporting Messaging Transport Type" is set to "JMS". The configuration fields are as follows:

JMS Transaction Report Destination Name	asg.cl.transaction.queue
JMS Transaction Report Destination Type	Queue
Use Exported JMS Connection Settings from EMS Component	<input checked="" type="checkbox"/>
JMS Provider URL	tcp://localhost:7222
JNDI Context URL	tibjmsnaming://localhost:7222
JMS Username	admin
JMS password	*****
JNDI Username	admin
JNDI password	*****

At the bottom of the form, there are five buttons: "Cancel", "Previous", "Menu", "Next", and "Finish".

**JMS Transaction Report Destination Name** - Specifies the name of the JMS destination to which the transaction reports are sent to the Central Logger by the Core Engine.

**JMS Transaction Report Destination Type** - Queue or Topic. Specifies the type of the JMS destination to which the transaction reports are sent to the Central Logger by the Core Engine. The default value is queue.

The Central Logger always listens on a queue. If the value of destination type is changed to **Topic**, then the JMS administrator must configure a bridge between the topic and the queue.

**Use Exported JMS Connection Settings from EMS Component**- If you are using TIBCO Enterprise Message Service you can check this box and skip configuration of JMS Provider URL, JNDI Context URL, JMS Username, JMS password, JNDI Username, and JNDI password.

Checking this box makes a Component Wizard page display to allow you upload the appropriate JAR files (`jms-2.0.jar` and `tibjms.jar`) to set and establish connection settings.

Otherwise use the default values or other appropriate values for the parameters that become available when this checkbox is cleared.

Refer to the descriptions of these parameters in [SOAP/JMS and ESB for Façade and Target Transport Messaging on page 16](#)

### Task G Configure the Parameters Used to Launch asg-engine

When the Enabler is set to launch a TIBCO® API Exchange Gateway Engine, Agent, or processing unit you will want to specify the parameters and properties for launching the API Service Gateway Engine (asg-engine).

Refer to the TIBCO® API Exchange Gateway User's Guide for information on the command line parameters, options, and properties files used to run the TIBCO® API Exchange Gateway Core Engine.

Figure 10 TIBCO API Exchange Gateway: Parameters Used to Launch asg-engine

**TIBCO API Exchange Gateway: APIExG\_CoreEngine**

TIBCO API Exchange Gateway : Parameters Used to Launch asg-engine

Command Line (Optional. Any parameters defined below -optional- will be added to the values entered in this text box)

Upload property file : -p parameter (Optional)

Property path name : -p parameter (Optional, value used if property file not uploaded)

Upload CDD file : -c parameter (Optional)

CDD path name : -c parameter (Optional, value used if CDD file not uploaded)

Launch action : -u parameter

Configuration name : -a parameter (Optional)

Upload EAR file (Optional)

EAR path name (Optional, value used if EAR file not uploaded)

This Enabler Component Wizard is used to automate execution of the command line options for the installation of the asg-engine, the TIBCO® API Exchange Gateway Core Engine.

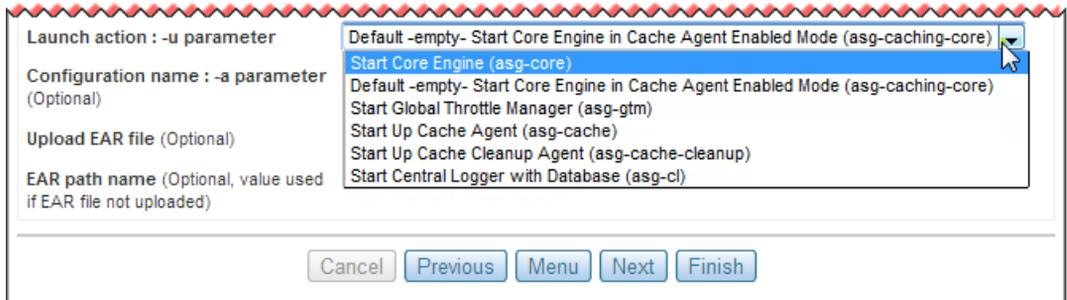
Using the various command line options presented in the Component Wizard you can programmatically launch and run a specific TIBCO® API Exchange Gateway processing unit or an agent as is described in the *Processing Units of TIBCO API Exchange Gateway Engine* section of the *TIBCO® API Exchange Gateway User's Guide*.

**Upload EAR file (optional)** - Upload a TIBCO® API Exchange Gateway enterprise archive (EAR) to publish and run when the Component is instantiated at runtime.

**Launch action: -u parameter** - allows selection, configuration, installation, and launch from the following list of options:

- Core Engine (asg-core)
- Core Engine in Cache Agent Enabled mode (asg-caching-core)
- Global Throttle Manager (asg-gtm)
- Cache Agent (asg-cache)
- Cache Cleanup Agent (asg-cache-cleanup)
- Central Logger with Database (asg-cl)

Figure 11 Launch action: -u parameter options



## Task H TIBCO API Exchange Gateway - Core Engine Settings

Set up reporting and messaging for the TIBCO API Exchange Gateway Core Engine. Enable reporting, facade and target JMS transport messaging, JMS delivery mode and JMS acknowledgement mode.

Figure 12 Set up the core engine for API Exchange component

Refer to [TIBCO API Exchange Gateway Configuration UI and Core Engine Settings on page 12](#) for descriptions on these settings.

If Facade/Target JMS Transport is enabled then proceed to [SOAP/JMS and ESB for Façade and Target Transport Messaging on page 16](#).

When Reporting is enabled refer to [Log Reporting Messaging on page 18](#) for a description of how to set those identical settings parameters.

## Task I Configure TIBCO® API Exchange Gateway Running Conditions

When the Enabler launches and runs the Gateway UI Configuration or an Engine, TIBCO Silver Fabric verifies that the product is running with a periodic check. If this component has specified the option to "install and configure without running to allow manual test of engine parameters" then this setting will not apply.

**Polling Period (in seconds) for TIBCO® API Exchange Gateway verification (required)** - This setting specifies the interval of time (in seconds) between verifications that TIBCO® API Exchange Gateway is still running. If TIBCO® API Exchange Gateway becomes unresponsive to this verification then the process is automatically restarted.

Figure 13 TIBCO® API Exchange Gateway Running Condition

The screenshot shows a configuration window titled "TIBCO API Exchange Gateway: APIExG\_Configuration UI". The main heading is "TIBCO API Exchange Gateway Running Condition". There are two input fields: "Polling period (in seconds) for TIBCO API Exchange Gateway verification (required)" with a value of 30, and "Automatically Restart Silver Fabric Engine if TIBCO API Exchange Gateway fails to restart N successive times (required)" with a value of 3. At the bottom, there are five buttons: "Cancel", "Previous", "Menu", "Next", and "Finish".

**Automatically Restart Silver Fabric Engine if TIBCO® API Exchange Gateway fails to restart N successive times (required)** - Enter an integer to specify the number of restart retries for the TIBCO® API Exchange Gateway before the TIBCO Silver Fabric Engine will be restarted. A successful restart will reset the count.

### Task J Central Logger Database Configuration

When the Component is being configured to launch the TIBCO API Exchange Gateway Central Logger with Database, the Database JDBC Connection configurations must be specified for use by the Central Logger.

Figure 14 Central Logger - Database Configuration

The screenshot shows a configuration window titled "TIBCO API Exchange Gateway: APIExG\_CentralLogger". The main heading is "TIBCO API Exchange Gateway Central Logger - Database Configuration". There are five input fields: "JDBC URL (required)" with a value of "jdbc:mysql://lin64vm217.us.safeway.com", "Username (required)" with a value of "asguser", "Password (required)" with a value of "\*\*\*\*\*", "JDBC Driver Class (required)" with a value of "com.mysql.jdbc.Driver", and "Upload JDBC Jar" with a value of "/cl\_jdbc/mysql-connector-java-5.0.8-bin.jar" and an "Upload" button. At the bottom, there are five buttons: "Cancel", "Previous", "Menu", "Next", and "Finish".

**JDBC URL** - Provides the protocol, database type, host and port to locate your database for the Central Logger

**Username and Password** - Make sure that a matching profile exists on the database. Required privileges are listed in the *TIBCO API Exchange Gateway User's Guide* in the section, Database Setup and Configuration for Central Logger.

**JDBC Driver Class** - Sets the class path for the driver.

**Upload JDBC Jar** - Upload the third-party JDBC driver for your database so the component can be published with the Distribution.

### Task K Add or Edit Enabler-specific Runtime Context Variables (optional)

**String, Environment, System, or Encrypted** variables may be added to the Component to define and set runtime specific context variables. Review the settings that were made previously and if required you can add or edit variables as necessary.

Select a variable type from the **Add Variable** pull-down list or **Add from Enabler**.

To change a variable value select the row and click **Edit** to change its value. Selected rows may also be removed.

Figure 15 Selecting a Variable prior to edit

TIBCO API Exchange Gateway: APIExG\_Configuration UI

Click Add Variable to add a new Runtime Context Variable that's specific to this Component. Click Add from Enabler to copy a variable from the Enabler to the Component. Select a variable and click Remove to remove it or Edit to modify it.

-- Add Variable -- Add from Enabler Edit Remove

Name	Value	Type	Description	Export	Auto Increment	Overridden	Enabler Variable
APIX_RUNNING_CONDITION_NB_APIX_RESTART_BEFORE_RESTART_ENGINE	3	String		False	None	False	
APIX_RUNNING_CONDITION_POLLPERIOD	30	String		False	None	False	
APIX_ENABLER_ACTION	Configuration UI	Environment		False	None	False	
APIX_ENABLER_CFGMGMT_DIR		Environment	When empty the location is the value of the runtime variable <code>APIX_CFGMGMT_DIR</code> (set in <code>APIXEnvCfg_data</code> )	False	None	False	
APIX_ENG_EAR_PATH_PARA <sub>APIX</sub>		Environment	Optional, value used if <code>APIX_CFGMGMT_DIR</code> is not uploaded	False	None	False	
APIX_ENG_COMMAND_LINE_PROPERTY		Environment	Optional. Any parameters defined below -optional- will be added to the values entered in this text box	False	None	False	
APIX_JMX_BASE_PORT	7777	Environment	Port used to connect to the engine to get statistics using JMX. To avoid port conflict, set <code>autoIncrementType</code> to Numeric. To have a predictable value set <code>autoIncrementType</code> to None.	False	Numeric	False	
APIX_CHECK_SERVER_RUNNING_AT_STARTUP	true	Environment	Will check if the server is running before notify the container is started. Set it to false to disable the check.	False	None	False	

Cancel Previous Menu Next Finish

Variable values from an Enabler may be added to the runtime as well. Use the **Add from Enabler** button to add Enabler-specific context variables.

### Task L Add Allocation Rule Settings

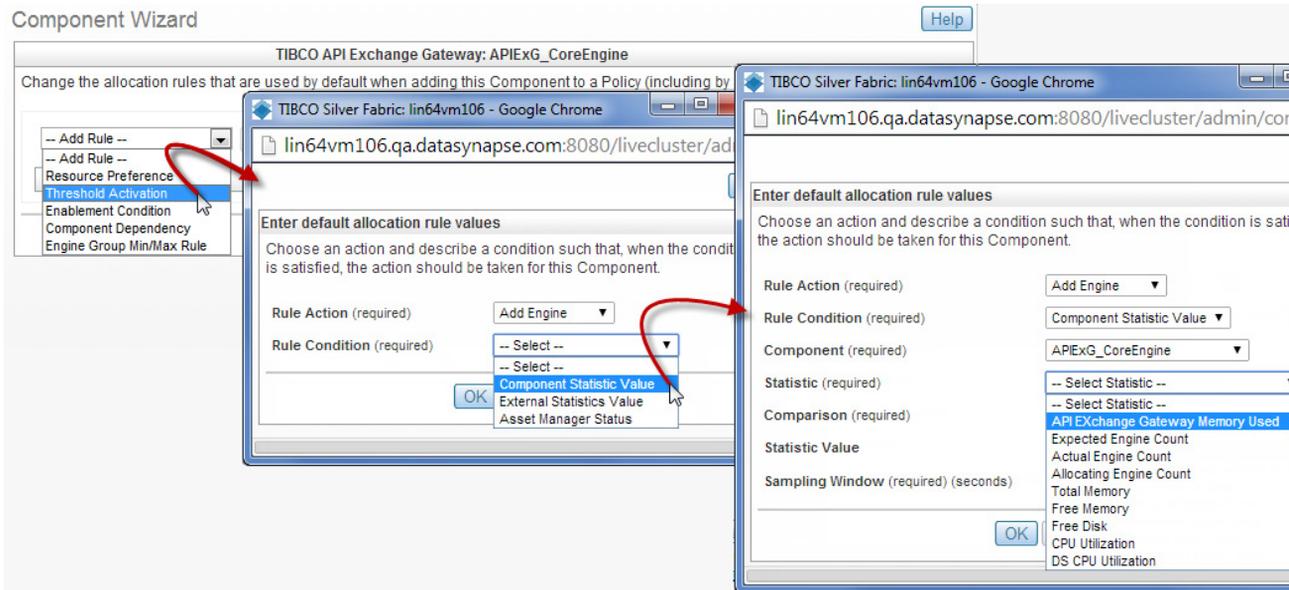
Add rules to specify and set Component behavior. Add rules to do the following:

- Specify Resource Preferences,

- Set Thresholds for Activation,
- Set Enablement Conditions,
- Specify Component Dependency, or
- Set Engine Group Minimums or Maximums

Each rule selection will bring up a slightly different dialog window that allows property selection of a tracked Engine or Component Archive statistic to be evaluated according to a logical operator and a value you specify to define an action. In some cases the Component Archive Statistics may be selected as is shown in the example below.

Figure 16 Setting a Threshold Activation



More information on using statistics for micro-scaling or archive scaling is available in the *TIBCO Silver® Fabric Cloud Administration Guide* and more about Component Archive scaling within a Stack is covered in this guide.

### Task M Finish Configuring the Component

The remaining screens of the Component Wizard are generic for all Silver® Fabric Enablers. The configuration of these is optional for the TIBCO® API Exchange Gateway component.

Refer to *TIBCO Silver® Fabric User's Guide* for more information on these configuration screens.

After you click the **Finish** button, make sure that the Component is published to make it available when creating a Stack.

To do this, select **Publish Component** in the Actions drop-down list located at the line of the Component you just created.

## Creating a Stack

Components are deployed within a Stack. Any number of Components could be included in your Stack depending on your implementation. Once you have created Components, you can create a Stack so they will be published to a Silver Fabric Engine as a unit.

After initially defining a Stack, you can still update it by adding, editing, or removing API Exchange Gateway Components.

To create a TIBCO Silver Fabric Enabler for TIBCO API Exchange Gateway Stack:

1. In the TIBCO Silver<sup>®</sup> Fabric Administration Tool, select **Stacks > Stacks**.
2. Click the **Create New Stack** button as shown in [Figure 17](#).

Figure 17 Creating an Stack



The screenshot shows the TIBCO Silver Fabric Administration Tool interface. The top navigation bar includes 'Dashboard', 'Stacks', 'Engines', 'Users', 'Reports', 'Admin', 'Config', and 'Diagnostic'. The main content area is titled 'Stacks' and features a 'Create New Stack' button with a yellow star icon, which is being clicked by a mouse cursor. To the right of this button is a 'More Actions' dropdown menu. Below the buttons is a table with the following data:

#	Actions	Name▲	Account	Description
1	  ▶	Example Stack	cloud	An example stack consisting of an HSQLDB database component and a J2EE web application component.
2	  ▶	My_Stack	cloud	This Stack will be dynamically published based upon tracked engine statistics.

At the bottom of the table, there is a 'Results Per Page' dropdown set to '20' and a pagination control showing 'Page: 1 of 1'.

3. Enter a Stack name in the Stack Builder page as shown in [Figure 18](#).
4. In the Components area, add the TIBCO Silver Fabric Enabler for TIBCO API Exchange Gateway Components that your implementation requires on a single machine.

- In the Policies area, expand the Component you just added to view the details of the Component.

Figure 18 Stack Builder page - Adding a Component Dependency

**Stack Builder**

**Name:** My\_Stack **Icon:**  **Change** **Components:** My Component, SF Administrator Component

**Account:** cloud **Contact:** admin

**Description:** This stack will be dynamically published based upon tracked engine statistics. **Technology:**  **Template:** local template

**Display Name:** My\_Stack

**Available Components**

- J2EE Example
- My Application Component
- Scripting Database Example
- TIBCO Administrator 2.0.0

**Selected Components**

- My Adapter Component
- SF Component

**Policies** | URLs | Components | Properties | Files | Archive Scaling

**Policies**

My\_Stack - Manual Mode

Component	Rules	min	max	priority
My Component	0	1	1	medium
SF Component	0	1	1	

**Rules**

Add a rule:  Choose a rule type

- Choose a rule type
- Component Dependency
- Enablement Condition
- Resource Preference
- Threshold Activation
- Engine Group Min/Max

**Create Component Dependency Rule**

Choose a Component that 'My Component' should depend on in this policy.

Depend on: SF Component

Shutdown dependency:

Ordered shutdown:

Restart Component for new rules:

Pack by host:

Save Cancel

## TIBCO Silver Fabric Engine Statistics

---

If you want TIBCO API Exchange Gateway Components to scale automatically, you can define rules that add or remove TIBCO Silver<sup>®</sup> Fabric Engines based on Engine statistics.

Data collected are aggregated. The aggregate is used to average raw statistic values by using a source ID. The average is calculated by individually averaging the statistic values for each source ID (for example, for each Engine), and then averaging the results across all Engines.

If the aggregated value triggers the rule, but the normalized geometric variance across the Engines is less than 0.85, then it does not add an Engine. Removing Engines is not affected by variance.

When an Engine is added, it will automatically publish the TIBCO API Exchange Gateway archive that was loaded in the component on a new Engine. For HTTP activities, for example, web services using SOAP over HTTP transport and HTTP Receiver activities, the Engine will add the URL to the load balancer automatically.

You can set up rules on Enablement Condition. The Engine will start upon statistics rules on other Engines. You also can setup rules for threshold activation, based on statistics gathered on the Engine itself or on that of other Engines.

To set up rules for an Engine, follow these steps:

1. In the Policies area of the TIBCO Silver Fabric Enabler for TIBCO API Exchange Gateway Stack builder page, select the component on which you want to set up rules.
2. Select **Threshold Activation** or **Enablement Condition** in the Add A Rule list.
3. If you select **Threshold Activation**, specify the following parameters in the Create Threshold Activation Rule panel:
  - In Condition Type list: select the **Component Statistic** item.
  - In the Action list, select **Add Engine** or **Remove Engine**.
  - In the Component list, select the Component where the statistic rules apply.
  - In the Statistics list, select the type of statistics.
  - In the Comparison list, select **Greater Than** or **Less Than**.
  - In the Value field, select the value of the statistic when it triggers the rules.
  - In the Sampling Window field, set the time interval (in seconds). It specifies how often the statistics is calculated.

Figure 19 Creating Rules

**Stack Builder**

**Name:** HA-APIX-Core-Cache-Cle  
**Account:** cloud  
**Description:** API Exch Gateway High-Availability with Core Engine, Caching Agent, and Cache-Cleanup Agent  
**Icon:** (none) [Change](#)  
**Contact:** admin  
**Technology:**  
**Template:** not template  
**Components:** APIExG\_CoreEngine, AI

**Display Name:** HA-APIX-Core+

Policies | URLs | Components | Properties | Files | Additional Info | Archive Scaling

**Policies**

HA-APIX-Core-Cache-Cleanup - Manual Mode

Component	Rules	min	max	priority
APIExG_CoreEngine	1	1	1	medium
<b>Rules</b> Add a rule: Choose a rule type				
<b>Type</b>		<b>Description</b>		
Threshold Activation <a href="#">Remove</a> <a href="#">Edit</a>		If APIExG_CoreEngine's Free Memory Less		
APIExG_CacheAgent	0	1	1	
APIExG_CacheCleanupAgent	0	1	1	

**Create Threshold Activation Rule**

Choose an action and describe a condition such that, when the condition is satisfied, the action should be taken for 'APIExG\_CoreEngine' in this policy.

**Condition type:** Asset Manager Status, **Component:** APIExG\_CoreEngine

**Component Statistic:** Free Memory

**Comparison:** Less than

**Action:** Add Engine

**Value:** 1000.0 KB

**Sampling Window:** 120 seconds

[Save](#) [Cancel](#)

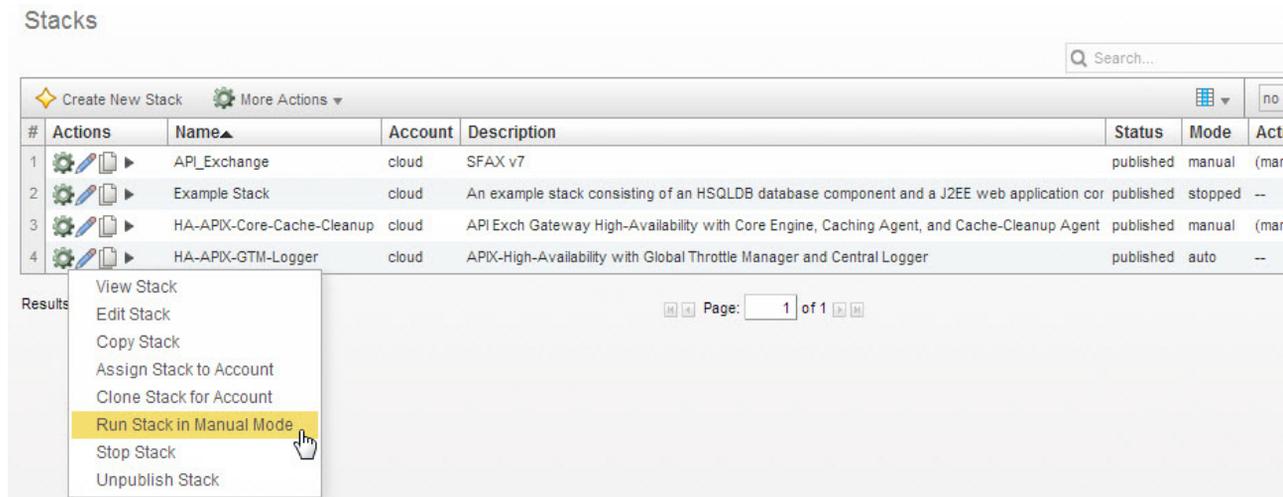
Engine statistics consist of statistic information about the TIBCO Silver Fabric Engine, and are independent of the TIBCO API Exchange Gateway Components aside from the Core Engine free memory available.

As shown in the graphic above, the TIBCO Silver Fabric Enabler for TIBCO API Exchange Gateway Core Engine Free Memory usage is the lone statistic (aside from Engine based stats) that can be used to set conditions for activation and enablement of new components at this time.

## Running a Stack

After you have created your Stack, use **Publish Stack** to make it available to run. After publishing click **Run Stack In Manual Mode** in the Actions drop-down list as shown in [Figure 20](#) to run the Stack immediately on available resources..

Figure 20 Running a Stack in Manual Mode



Alternatively, if the Stack is defined with a Policy (schedule) you can **Run (the) Stack in Auto Mode**. The Stack will run according to the schedule selected for the Stack on available resources.

## Updating a Stack

---

When a Stack is published and running, you can still make changes to the Stack such as adding other components, changing allocation rules, changing threshold activation rules, or deploying and starting archives on the runtime API Exchange Gateway Application instantiated on the Engine.

Making changes to the Stack is as easy as editing, saving, and publishing those changes to any instances that may be running. Some changes may require restart of the changed resource, so consult the TIBCO Silver Fabric documentation for best practices prior to making changes to a production system.

After making any changes to a Stack, **Save** the changes and then from the Actions list in the main Stack page, select **Publish Changes**. The specified Engines will be affected by the changes immediately.

# Index

- AUTO\_ACKNOWLEDGE 13
- CLIENT\_ACKNOWLEDGE 13
  - Components 4
  - distributions 7
  - Documentation iv
  - Documentation vii
- DUPS\_OK\_ACKNOWLEDGE 14
  - Enable Reporting 12
  - Enablement condition 29
  - Enabler Facade/Target JMS Transport 12
  - entity 4
  - environment variable 24
- EXPLICIT\_CLIENT\_ACKNOWLEDGE; explicit acknowledgement 14
- EXPLICIT\_CLIENT\_DUPS\_OK\_ACKNOWLEDGE; explicit acknowledgement 15
- Exported JMS Connection Settings 17
- Exported JMS Connection Settings from EMS Component 19
- Functionalities 2
- JDBC URL 23
- JMS Acknowledgment Mode 13
- JMS Delivery Mode 13
- JMS Delivery Mode, Non Persistent 13
- JMS Delivery Mode, Reliable 13
- JMS Delivery Mode; Persistent 13
- JMS password 18
- JMS Provider URL 18
- JMS Transaction Report Destination Name 19
- JMS Transaction Report Destination Type 19
- JMS Username 18
- JNDI Context Factory 17
- JNDI Context URL 18
- JNDI Password 18
- JNDI Username 18
- Log Reporting Messaging Transport Type 18
- NO\_ACKNOWLEDGE 14
  - Non-persistent 13
  - Persistent 13
- Queue Connection Factory Name 17
- RELIABLE 13
- Reply Queue 17
- Request Queue 17
- runtime specific context variables 24
- SILVERFABRIC\_HOME v
  - source ID 29
- Stack Builder Page 28
- Stacks 27
  - system variable 24
- tasks, creating an application 4
- technical support; customer support; support, contacting vii
- Threshold Activation 29
- TIBCO Silver Fabric Enabler for TIBCO API Exchange Gateway 2
- TIBCO Silver Fabric Engine 29
- TIBCO\_HOME v
- TIBCOCommunity vii
- Topic Connection Factory Name 17
  - variable, environment 24
  - variable, string 24
  - variables, encrypted 24
  - variables, System 24