



# **TIBCO BusinessConnect™ ConfigStore Management Interface Protocol**

## **User's Guide**

*Software Release 6.1  
October 2019*



## 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.

ANY SOFTWARE ITEM IDENTIFIED AS THIRD PARTY LIBRARY IS AVAILABLE UNDER SEPARATE SOFTWARE LICENSE TERMS AND IS NOT PART OF A TIBCO PRODUCT. AS SUCH, THESE SOFTWARE ITEMS ARE NOT COVERED BY THE TERMS OF YOUR AGREEMENT WITH TIBCO, INCLUDING ANY TERMS CONCERNING SUPPORT, MAINTENANCE, WARRANTIES, AND INDEMNITIES. DOWNLOAD AND USE OF THESE ITEMS IS SOLELY AT YOUR OWN DISCRETION AND SUBJECT TO THE LICENSE TERMS APPLICABLE TO THEM. BY PROCEEDING TO DOWNLOAD, INSTALL OR USE ANY OF THESE ITEMS, YOU ACKNOWLEDGE THE FOREGOING DISTINCTIONS BETWEEN THESE ITEMS AND TIBCO PRODUCTS.

This document 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, the TIBCO logo, the TIBCO O logo, TIB, Information Bus, TIBCO Hawk, TIBCO Rendezvous, TIBCO Runtime Agent, TIBCO ActiveMatrix BusinessWorks, TIBCO Administrator, TIBCO Designer, TIBCO BusinessConnect, and TIBCO EDISIM are either registered trademarks or trademarks of TIBCO Software Inc. in the United States and/or other countries.

Java and all Java based trademarks and logos are trademarks or registered trademarks of Oracle Corporation and/or its affiliates.

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.

This and other products of TIBCO Software Inc. may be covered by registered patents. Please refer to TIBCO's Virtual Patent Marking document (<https://www.tibco.com/patents>) for details.

Copyright © 1999-2019. TIBCO Software Inc. All Rights Reserved.

# Contents

<b>Preface</b> .....	<b>v</b>
Related Documentation .....	vi
TIBCO BusinessConnect ConfigStore Management Interface Protocol Documentation .....	vi
Other TIBCO Product Documentation .....	vi
Typographical Conventions .....	viii
TIBCO Product Documentation and Support Services .....	xi
How to Access TIBCO Documentation .....	xi
How to Contact TIBCO Support .....	xi
How to Join TIBCO Community .....	xi
<b>Chapter 1 Overview</b> .....	<b>1</b>
Product Overview .....	2
Using the CMI Protocol .....	4
<b>Chapter 2 Getting Started - Tutorial</b> .....	<b>5</b>
Prerequisites .....	6
Run the Tutorial .....	7
Create an Internal Application User .....	8
Create a New Internal Application User .....	8
Import the Sample Project .....	10
Configure a Connection to TIBCO BusinessConnect .....	12
Working with Sample Operations .....	14
Edit the Global Variables .....	14
Insert a Participant .....	15
Insert a Business Agreement .....	20
Get One Enabled Protocol .....	22
<b>Chapter 3 Internal Application Users</b> .....	<b>25</b>
Configure an Internal Application User .....	26
<b>Chapter 4 Transactions</b> .....	<b>27</b>
Operation Overview .....	28
Operation Categories .....	28
Command Rules for the Parent/Child Element .....	30

Read Operations .....	31
Write Operations .....	33
Transaction Overview .....	36
Sessions .....	36
Batch Transactions .....	37
Private Process Smart Routing .....	38
Using the Operations Editor .....	39
Import Operations .....	39
Display the Imported Operations .....	39
<b>Chapter 5 Viewing Audit Logs .....</b>	<b>43</b>
Log Viewer Overview .....	44
Perform a Log Search .....	45
Audit Logs .....	45
Message Queue .....	47
Send or Cancel Messages in the Log Queue .....	49
Viewing Search Results .....	50
Summary View .....	50
Saving and Reusing Queries .....	53
Save a Query .....	53
Reuse a Query .....	53
<b>Chapter 6 Private Process Communications .....</b>	<b>55</b>
Private Message Categories .....	56
Private Messages in Sessions .....	57
Private Messages in Batch Transactions .....	58
Initiator Request .....	60
Initiator Response .....	62
Advisory .....	64
Error .....	65
<b>Appendix A Error Messages .....</b>	<b>67</b>
Error Codes and Messages .....	68
<b>Index .....</b>	<b>69</b>

# Preface

TIBCO BusinessConnect™ ConfigStore Management Interface Protocol is a set of message-based interfaces that allows the private processes to interact with the TIBCO BusinessConnect™ server to retrieve or update configuration data. It acts as a service to other protocols based on TIBCO BusinessConnect to facilitate management of trading partners and business agreements.

## Topics

---

- [Related Documentation, page vi](#)
- [Typographical Conventions, page viii](#)
- [TIBCO Product Documentation and Support Services, page xi](#)

## Related Documentation

---

This section lists documentation resources you may find useful.

### TIBCO BusinessConnect ConfigStore Management Interface Protocol Documentation

The following documents form the *TIBCO BusinessConnect ConfigStore Management Interface Protocol* documentation set:

- *TIBCO BusinessConnect™ ConfigStore Management Interface Protocol Installation and Configuration*: Read this manual for instructions on site preparation and installation.
- *TIBCO BusinessConnect™ ConfigStore Management Interface Protocol User's Guide*: Read this manual for instructions on using the product.
- *TIBCO BusinessConnect™ ConfigStore Management Interface Protocol 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:

- **TIBCO BusinessConnect™ software**: A B2B (business-to-business) gateway that allows your company to engage in electronic commerce with your business partners.
- **TIBCO Administrator™ software**: The software allows you to manage users, machines and applications defined in a TIBCO Administration Domain. The TIBCO Administrator graphical user interface enables users to deploy, monitor, as well as start and stop TIBCO applications.
- **TIBCO ActiveMatrix BusinessWorks™ software**: This software is a scalable, extensible, and easy to use integration platform that allows you to develop integration projects. TIBCO BusinessWorks includes a graphical user interface (GUI) for defining business processes and an engine that executes the process
- **TIBCO Designer™ software**: This graphical user interface is used for designing and creating integration project configurations and building an Enterprise Archive (EAR) for the project. The EAR can then be used by TIBCO Administrator for deploying and running the application.
- **TIBCO Runtime Agent™ software**: This software suite is a prerequisite for other TIBCO software products. In addition to TIBCO Runtime Agent components, the software suite includes the third-party libraries used by other TIBCO products such as

TIBCO Designer, Java Runtime Environment (JRE), TIBCO Hawk®, and TIBCO Rendezvous®.

- TIBCO Rendezvous®: This software enables programs running on many different kinds of computers on a network to communicate seamlessly. It includes two main components: the Rendezvous programming language interface (API) in several languages, and the Rendezvous daemon.
- TIBCO Enterprise Message Service™ software: This software provides a message service that enables integration of applications within an enterprise based on the Java Message Service (JMS) specifications.

## Typographical Conventions

---

The following typographical conventions are used in this manual.

Table 1 General Typographical Conventions

Convention	Use
<i>ENV_NAME</i>	TIBCO products are installed into an installation environment. A product installed into an installation environment does not access components in other installation environments. Incompatible products and multiple instances of the same product must be installed into different installation environments.
<i>TIBCO_HOME</i>	
<i>bccmi_HOME</i>	
	<p>An installation environment consists of the following properties:</p> <ul style="list-style-type: none"> <li>• <b>Name</b> Identifies the installation environment. This name is referenced in documentation as <i>ENV_NAME</i>. On Microsoft Windows, the name is appended to the name of Windows services created by the installer and is a component of the path to the product shortcut in the Windows Start &gt; All Programs menu.</li> <li>• <b>Path</b> The folder into which the product is installed. This folder is referenced in documentation as <i>TIBCO_HOME</i>.</li> </ul> <p><i>TIBCO BusinessConnect ConfigStore Management Interface Protocol</i> installs into a directory within a <i>TIBCO_HOME</i>. This directory is referenced in documentation as <i>bccmi_HOME</i>. The default value of <i>bccmi_HOME</i> depends on the operating system. For example on Windows systems, the default value is <code>C:\tibco\bc\version\protocols\emi</code>.</p>
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>
bold code font	<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, <code>MyCommand</code> is enabled: <code>MyCommand [enable   disable]</code></li> </ul>

Table 1 General Typographical Conventions (Cont'd)

Convention	Use
<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: <code>MyCommand <i>PathName</i></code></li> </ul>
Key combinations	<p>Key name separated by a plus sign indicates 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> <p><code>MyCommand [optional_parameter] required_parameter</code></p>
	<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> <p><code>MyCommand para1   param2   param3</code></p>

Table 2 *Syntax Typographical Conventions (Cont'd)*

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 <code>param1</code> and <code>param2</code>, or the pair <code>param3</code> and <code>param4</code>.</p> <pre>MyCommand {param1 param2}   {param3 param4}</pre> <p>In the next example, the command requires two parameters. The first parameter can be either <code>param1</code> or <code>param2</code> and the second can be either <code>param3</code> or <code>param4</code>:</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 <code>param1</code>. You can optionally include <code>param2</code> as the second parameter. And the last parameter is either <code>param3</code> or <code>param4</code>.</p> <pre>MyCommand param1 [param2] {param3   param4}</pre>

# TIBCO Product Documentation and Support Services

---

For information about this product, you can read the documentation, contact TIBCO Support, or join TIBCO Community.

## How to Access TIBCO Documentation

Documentation for TIBCO products is available on the TIBCO Product Documentation website mainly in the HTML and PDF formats.

The TIBCO Product Documentation website is updated frequently and is more current than any other documentation included with the product. To access the latest documentation, visit <https://docs.tibco.com>.

Documentation for TIBCO BusinessConnect ConfigStore Management Interface Protocol is available on the <https://docs.tibco.com/products/tibco-businessconnect-configstore-management-interface-protocol> Product Documentation page.

## How to Contact TIBCO Support

You can contact TIBCO Support in the following ways:

- For an overview of TIBCO Support, and information about getting started with TIBCO Support, visit <http://www.tibco.com/services/support>
- For accessing the Support Knowledge Base and getting personalized content about products you are interested in, visit the TIBCO Support portal at <https://support.tibco.com>.
- For creating a Support case, you must have a valid maintenance or support contract with TIBCO. You also need a user name and password to log in to <https://support.tibco.com>. If you do not have a user name, you can request one by clicking Register on the website.

## How to Join TIBCO Community

TIBCO Community is the official channel for TIBCO customers, partners, and employee subject matter experts to share and access their collective experience. TIBCO Community offers access to Q&A forums, product wikis, and best practices. It also offers access to extensions, adapters, solution accelerators, and tools that extend and enable customers to gain full value from TIBCO products. In addition, users can submit and vote on feature requests from within the [TIBCO Ideas Portal](#). For a free registration, go to <https://community.tibco.com>.



## Chapter 1 **Overview**

This chapter describes TIBCO BusinessConnect ConfigStore Management Interface Protocol and how it is used to manage trading partners and business agreements for other protocols based on TIBCO BusinessConnect.

### Topics

---

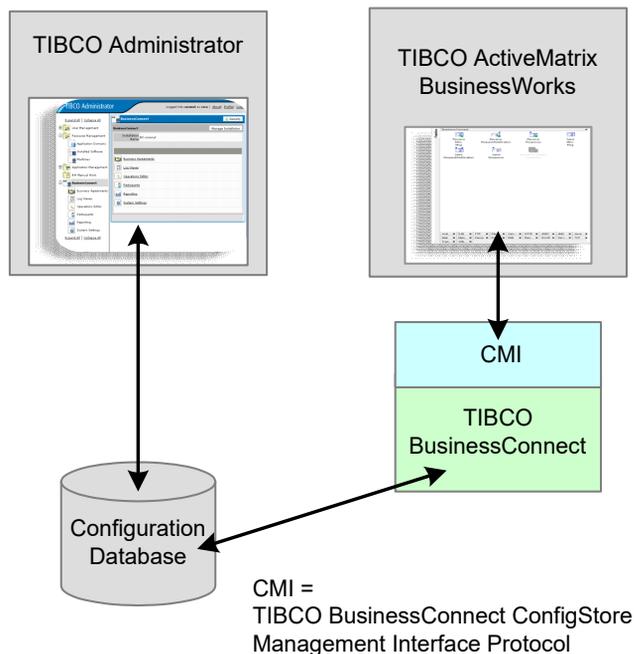
- [Product Overview, page 2](#)
- [Using the CMI Protocol, page 4](#)

## Product Overview

TIBCO BusinessConnect ConfigStore Management Interface Protocol (later in this book named *CMI Protocol*) is a set of message-based interfaces that allows the private processes to interact with the TIBCO BusinessConnect server to retrieve or update configuration data. It acts as a service to other protocols based on TIBCO BusinessConnect to facilitate management of trading partners and business agreements.

CMI Protocol performs bulk creation of trading partners and business agreements, or bulk changes in their configurations.

Figure 1 CMI Protocol



CMI Protocol has the following characteristics:

- It is integrated with TIBCO BusinessConnect for the functions Operations Editor and Audit Log Viewer.
- It is not enabled for a specific participant or bound to a specific business agreement.
- Involves interactions with the private processes, but not the interactions with trading partners.
- Protocol level properties can be retrieved or updated but not created.

- Only the TIBCO BusinessConnect level functions can be configured using CMI Protocol, such as:
  - Participants (Business Locations, Contacts, Transports, Setting Proxy, Credentials, DomainIDs, and Enabled Protocols)
  - Business Agreements (Document Exchange, Setting Transports, Setting Operation Bindings for Protocol, and Scheduled Transmissions)
  - Audit Logs
  - READ Operations

Other settings, such as user access for participants or business agreements, can be configured only using the TIBCO Administrator GUI.

## Using the CMI Protocol

---

In order to learn about CMI Protocol and start to use it, it is recommended that you read this manual in the following order:

### **Task A Getting Started**

To preview some basic operations, first install and deploy TIBCO BusinessConnect and then use the provided sample operations to manage trading partners and business agreements from the private process. Learn how to edit variables to create a trading partner or a business agreement.

See [Chapter 2, Getting Started - Tutorial, page 5](#) for details.

### **Task B Create an Internal Application User**

To work with CMI Protocol, a specific type of user named *Internal Application User* has to be created and authenticated. A quick setup for this user is provided as a part of the Tutorial.

For more information, see [Chapter 3, Internal Application Users, page 25](#).

### **Task C Learn how to Manage Transactions**

To learn about transactions and operations available in CMI Protocol, see [Chapter 4, Transactions, page 27](#).

### **Task D View Audit Logs**

Audit logs for CMI Protocol are explained in [Chapter 5, Viewing Audit Logs, page 43](#).

### **Task E Learn about Private Process Communications**

To learn more about private process messages used in CMI Protocol, see [Chapter 6, Private Process Communications, page 55](#).

### **Task F Learn how to Interpret Error Messages**

Error messages for CMI Protocol are listed in [Appendix A, Error Messages, on page 67](#).

## Chapter 2 **Getting Started - Tutorial**

This tutorial explains how to use CMI Protocol to manage trading partners and business agreements.

### Topics

---

- [Prerequisites, page 6](#)
- [Run the Tutorial, page 7](#)
- [Create an Internal Application User, page 8](#)
- [Import the Sample Project, page 10](#)
- [Configure a Connection to TIBCO BusinessConnect, page 12](#)
- [Working with Sample Operations, page 14](#)

## Prerequisites

---

To run the tutorial, you need to have the following installed and configured:

### Task A Install and Configure TIBCO BusinessConnect

Install and configure TIBCO BusinessConnect as explained in the manual *TIBCO BusinessConnect Installation and Configuration*:

- Chapter 2, Installation
- Chapter 3, Configuration



Always check the file `TIB_bccmi_version_readme.txt` for the latest release requirements.

### Task B Install and Activate CMI Protocol

Install and activate the CMI protocol using the steps explained in *TIBCO BusinessConnect ConfigStore Management Interface Protocol Installation*:

- Chapter 2, Installation Steps.
- Chapter 3, Protocol Activation and Deployment.

### Task C Deploy TIBCO BusinessConnect

Deploy TIBCO BusinessConnect as explained in the manual *TIBCO BusinessConnect Interior Server Administration*, Chapter 2, Deployment Configuration.

### Task D Create an Internal Application User

In order to use the CMI Protocol, you need to create a new user as explained in [Create an Internal Application User on page 8](#).

## Run the Tutorial

---

After the prerequisite tasks have been performed, you can manage trading partners and business agreements from the private processes using the supplied CMI Protocol operations.

The tutorial explains how to use three sample operations in order to create a host, a partner, and a business agreement.

Follow the explanations given in the following sections:

- [Import the Sample Project on page 10](#)
- [Configure a Connection to TIBCO BusinessConnect on page 12](#)
- [Working with Sample Operations on page 14](#)

## Create an Internal Application User

To use this tutorial, you must have an Internal Application User that is authorized to perform tasks related to managing trading partners and business agreements using CMI Protocol.

For more information on creating and using Internal Application Users, see [Chapter 3, Internal Application Users, on page 25](#). For this tutorial, follow the simplified procedure to get started.

### Create a New Internal Application User

To create a new Internal Application User:

1. Select **BusinessConnect > User Management > Users**.

Only one user, the TIBCO Administrator User (admin) is present by default.

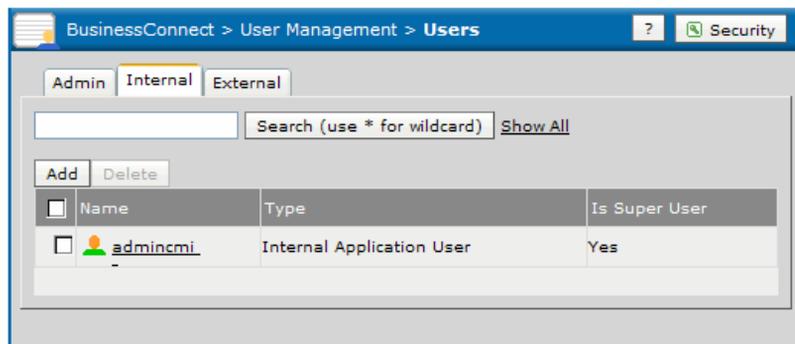
2. Select the **Internal** tab and click **Add**.

3. In the User Name window, enter the name for this new user, such as **admincmi**, and click **OK**.

4. To protect the new user, click **Set...**, enter the password twice, and then click **OK** and **Save**.

The new Internal Application User with a Super User role is added.

Figure 2 Internal Application User Added



To change the password at any time, click on the user name link and then click **change...** to set a new password.

Enter the password twice and click **OK**.

5. Leave the **Group Membership** tab and **Permissions** tab unchanged.



Group Membership and Permissions tab options don't need to be configured, since an Internal User is always added as a super user by default.

6. Click **Save** to save this user.

With the new Internal Application User **admincmi** authorized to work on all participant and business agreement tasks, you can proceed with the tutorial.

## Import the Sample Project

---

CMI Protocol contains a sample project which can be found at  
`BC_HOME\protocols\cmi\examples\bw\CMI.zip`

To open the sample TIBCO ActiveMatrix BusinessWorks project:

1. Start TIBCO Designer.
2. Click **New empty project**.
3. In the Save Project dialog, click **Cancel**.
4. Select **Project > Import Full Project**.
5. Click the **ZIP Archive** tab.
6. Navigate to the directory `BC_HOME\protocols\cmi\examples\bw`.
7. Select **CMI.zip**.
8. Click **OK**.

The Import - Options dialog appears.

9. In the Options tab, select the **Try rename in case of name conflict** radio button.
10. Click **Apply**.
11. Select **Project > Save As**.
12. In the Project Directory file chooser, navigate to the location where you want to save your project.



If you want to be able to restore the project for later use, be sure to select a directory other than `BC_HOME\protocols\cmi\examples\bw`, such as

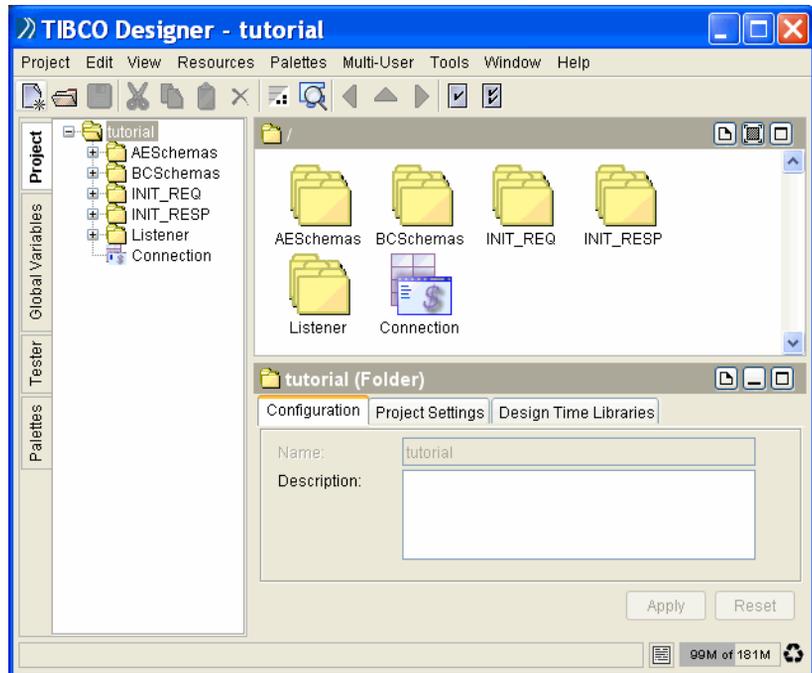
`BC_HOME\protocols\cmi\examples\bw\tutorial`.

If you select the directory `BC_HOME\protocols\cmi\examples\bw`, the zip archive file will be deleted.

13. Click **OK**.
14. Open the project by clicking on the (+) signs next to the project elements.

The window shown in [Figure 3](#) will display.

Figure 3 TIBCO ActiveMatrix BusinessWorks Project



15. Save the project but do not exit TIBCO Designer.

## Configure a Connection to TIBCO BusinessConnect

---

To configure a connection to TIBCO BusinessConnect, do the following:

1. In TIBCO Designer, click the **Project** tab.
2. Expand the folder.
3. Select the **Connection** resource.
4. Click the **BusinessConnect Server Access** tab.
  - a. Select the JDBC driver used to communicate with the TIBCO BusinessConnect configuration store from the JDBC Driver drop-down list.
  - b. Type the URL for the configuration store in the JDBC URL field.
  - c. Type the configuration store user name and password in the DB User and Password fields.
  - d. Click the **Apply** button.
5. Click the **Configuration** tab.
6. Click the **Update from Configuration Store** button. If you chose TIBCO EMS as the transport for private communication, the software displays a TIBCO EMS tab.
7. Select **CMI** from the Protocol Name drop-down list.
8. Click the **Select Operations** check mark.
9. Select either *All/None* by checking/unchecking the **All/None** check mark, or select specific operations.

For this tutorial, select **All** operations.

10. Click **OK**.
11. Click the **Import Selected Business Protocol** button.

A dialog will appear with three options:

- Yes
- Yes, Perform Maintenance on 'CMI' schemas
- Cancel

12. Click the option **Yes, Perform Maintenance on 'CMI' schemas**.
13. Click **Apply**.

When you import the protocol, TIBCO ActiveMatrix BusinessWorks retrieves schema information from the TIBCO BusinessConnect configuration store and puts it in the BCSchema project folder.

A list of all imported operations for CMI Protocol appears in the Imported Operations window each preceded with the operation schema version number (1.0), and group name:

- 1.0/Batch/cancelBatch
- 1.0/Batch/executeBatch
- 1.0/Read/getManyAuditLog
- 1.0/Read/getManyBusinessAgreement
- 1.0/Read/getManyParticipant
- 1.0/Read/getManyProtocolOperation
- 1.0/Read/getOneAuditLog
- 1.0/Read/getOneBusinessAgreement
- 1.0/Read/getOneEnabledProtocol
- 1.0/Read/getOneOperationBinding
- 1.0/Read/getOneOperationTree
- 1.0/Read/getOneParticipant
- 1.0/Session/login
- 1.0/Session/logout
- 1.0/Write/deleteBusinessAgreement
- 1.0/Write/deleteParticipant
- 1.0/Write/insertBusinessAgreement
- 1.0/Write/insertParticipant
- 1.0/Write/updateBusinessAgreement
- 1.0/Write/updateEnabledProtocol
- 1.0/Write/updateOperationBinding
- 1.0/Write/updateOperationTree
- 1.0/Write/updateParticipant
- 1.0/Write/upsertBusinessAgreement
- 1.0/Write/upsertParticipant

#### 14. Save the project.

## Working with Sample Operations

This tutorial explains how to use private processes instead of TIBCO Administrator's GUI to create new trading partners and business agreements for TIBCO BusinessConnect. It also explains how to read the properties of an enabled protocol that belongs to a participant.

The sample operations are:

- 1.0/Write/insertParticipant
  - [Insert a Participant, page 15](#)
- 1.0/Write/insertBusinessAgreement
  - [Insert a Business Agreement, page 20](#)
- 1.0/Read/getOneEnabledProtocol
  - [Get One Enabled Protocol, page 22](#)



For more information see [Operation Categories on page 28](#).

## Edit the Global Variables

Before you can proceed with creating trading partners and business agreements using the provided operations, verify that all global variables are correct.

1. Select the tab **Global Variables**.
2. In the CMI section, make sure that the path to your example files as well as the Internal Application User's name and password are correct.

*Figure 4 CMI Data in Global Variables*

```

CMI
├── cmi.doc.path = "C:\tibco\bc\6.0\protocols\cmi\examples\SampleDocs"
├── cmi.log.path = "C:\tibco\bc\6.0\protocols\cmi\examples\SampleDocs\log"
├── cmi.request.password = "*****"
└── cmi.request.user = "admincmi"
  
```

The default location for sample operations is in the directory

`BC_HOME\protocols\cmi\examples\SampleDocs`.



Make a backup of these files to keep the original examples intact and to reuse them after any modifications you have made on this default location.

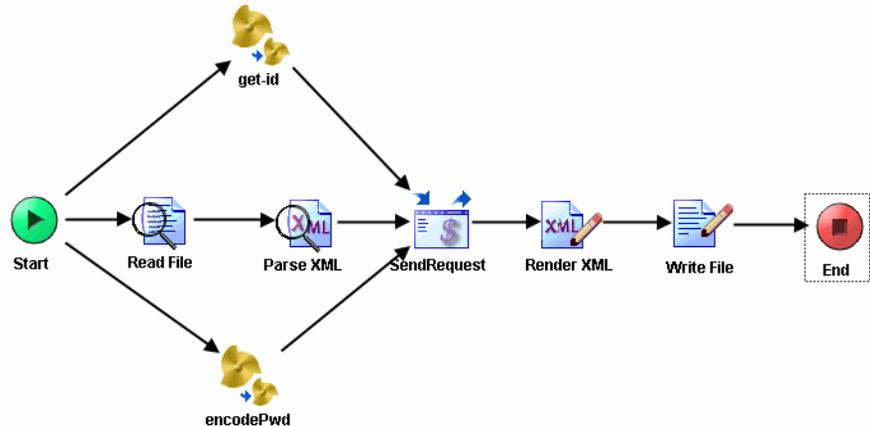
## Insert a Participant

To create a new Host or Partner participant:

1. Select the Project tab in TIBCO Designer.
2. Select *project\_name* > **INIT\_REQ** > **write** > **op\_insertparticipant**.

The process *insertparticipant* opens in the design window.

Figure 5 Insert Participant Process



3. Select the resource **ReadFile**.
4. Select the **Input** tab.

Figure 6 Read File for the Participant



In the Activity Input window, the *fileName* parameter shows which sample file will be used for this operation: *INSERTParticipantREQ.xml* listed in *GlobalVariables/CMi/cmi.doc.path*.

5. To edit the sample XML file, go to *BC\_HOME*\protocols\cmi\examples\SampleDocs\INSERTParticipantREQ.xml
6. Open the file with a text editor.

*Example 1 INSERTParticipantREQ.xml*


---

```

<?xml version="1.0" encoding="UTF-8" ?>
- <INSERTParticipantREQ xmlns="http://www.tibco.com/AX/BC/CMI" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.tibco.com/AX/BC/CMI ../../src/com/tibco/ax/cmi/resource/cmipayload.xsd">
- <Participant name="partner1" isActive="true" type="Partner"> <ListOfBizLocations>
  <BizLocation name="Headquarters" addr1="PA"/>
- <BizLocation name="loc1">
  <PrimaryLegalContact firstName="1" lastName="1"/>
  <PrimarySupportContact firstName="3" lastName="2"/>
- <ListOfContacts>
  <BizContact lname="1" fname="1" type="Legal"/>
  <BizContact lname="2" fname="3" type="Support"/>
  </ListOfContacts>
</BizLocation>
</ListOfBizLocations>
- <ListOfDomainIds>
  <DomainId domainName="AS2_ID" idName="12345"/>
  </ListOfDomainIds>
- <ListOfEnabledProtocols>
- <EnabledProtocol name="EZComm">
  <DefaultAS2DomainId idName="12345"/>
- <Transports>
  <HttpTransport name="bbe" url="http://localhost:8080/EZComm"/>
  </Transports>
</EnabledProtocol>
</ListOfEnabledProtocols>
</Participant>
</INSERTParticipantREQ>

```

---

All variables (in the example above highlighted in bold) can be modified according to your requirements.

**Insert a Host**

To insert a Host, modify the sample file as follows:

*Example 2 INSERTParticipantREQ.xml modified for a Host*


---

```

<?xml version="1.0" encoding="UTF-8" ?>
- <INSERTParticipantREQ xmlns="http://www.tibco.com/AX/BC/CMI" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.tibco.com/AX/BC/CMI ../../src/com/tibco/ax/cmi/resource/cmipayload.xsd">
- <Participant name="Host" isActive="true" type="Host"> <ListOfBizLocations>
  <BizLocation name="Headquarters" addr1="PA"/>
- <BizLocation name="loc1">
  <PrimaryLegalContact firstName="1" lastName="1"/>
  <PrimarySupportContact firstName="3" lastName="2"/>
- <ListOfContacts>
  <BizContact lname="1" fname="1" type="Legal"/>
  <BizContact lname="2" fname="3" type="Support"/>
  </ListOfContacts>
</BizLocation>

```

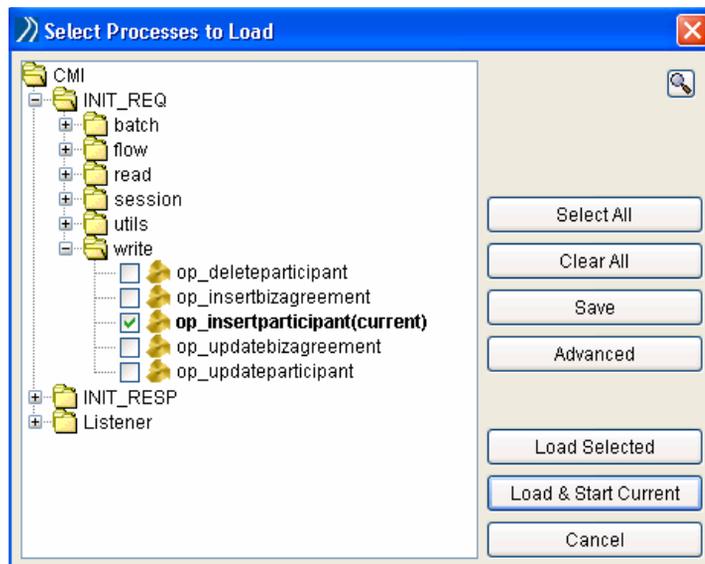
```

</ListOfBizLocations>
-<ListOfDomainIds>
  <DomainId domainName="AS2_ID" idName="12345"/>
</ListOfDomainIds>
-<ListOfEnabledProtocols>
  -<EnabledProtocol name="EZComm">
    <DefaultAS2DomainId idName="12345"/>
  </EnabledProtocol>
</ListOfEnabledProtocols>
</Participant>
</INSERTParticipantREQ>

```

1. Enter the Participant Name (Host)
2. For the Type, enter Host
3. Edit other data as desired
4. Remove the information about transports, which applies to the Partner.
5. Save the sample file.
6. In TIBCO Designer, select the **Tester** tab.
7. Click the green triangle icon  next to Jobs.  
The Select Processes to Load dialog appears.

Figure 7 Select Processes to Load



8. Select the process **op\_insertparticipant** and click on **Load and Start Current**.

9. The process will be executed.
  - If it is successful, you will be able to see the Host participant added to the TIBCO Administrator GUI under **BusinessConnect > Participants**.
  - If the operation is not successful:
    - You can debug it in TIBCO Designer by checking the output tab of the SendRequest activity in op\_instertparticipant process.  
Check the response file that has been created in the logs folder (path is provided for the global variable `cmi.log.path`).
    - You can also use the Log Viewer function in TIBCO Administrator to see the summary of this operation, including the failure causes.  
To learn more about viewing audit logs, see [Chapter 5, Viewing Audit Logs, on page 43](#).

## Insert a Partner

To insert a Partner, modify the sample file as follows:

*Example 3 INSERTParticipantREQ.xml modified for a Partner*

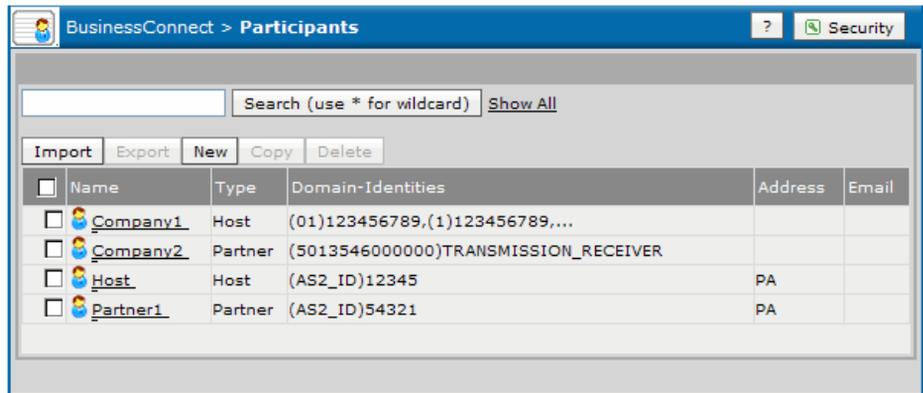
---

```
<?xml version="1.0" encoding="UTF-8" ?>
-<INSERTParticipantREQ xmlns="http://www.tibco.com/AX/BC/CMI" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.tibco.com/AX/BC/CMI ../../src/com/tibco/ax/ax/cmi/resource/cmipayload.xsd">
-<Participant name="Partner1" isActive="true" type="Partner"> <ListOfBizLocations>
  <BizLocation name="Headquarters" addr1="PA"/>
  <BizLocation name="loc1">
    <PrimaryLegalContact firstName="3" lastName="3"/>
    <PrimarySupportContact firstName="5" lastName="5"/>
  </BizLocation>
  </ListOfBizLocations>
-<ListOfContacts>
  <BizContact lname="3" fname="3" type="Legal"/>
  <BizContact lname="5" fname="5" type="Support"/>
  </ListOfContacts>
-</Participant>
-</INSERTParticipantREQ>
```

---

1. Enter the name for this Partner (**Partner1**).
2. For the Type, enter **Partner**
3. Edit other data as desired
4. Edit the transport information, which is needed for the partner participant.
5. Be sure to enter a different idName from the one used for the Host.
6. Save the sample file.
7. With TIBCO Designer, select the **Tester** tab.
8. Click the green triangle icon  next to Jobs.  
The Select Processes to Load dialog appears.
9. Select the process **op\_insertparticipant** and click on **Load and Start Current**.
10. The process will be executed.
  - If it is successful, you will be able to see the Partner participant added to the TIBCO Administrator GUI under **BusinessConnect > Participants**.

Figure 8 Host and Partner added to the TIBCO Administrator GUI



— If the operation is not successful:

You can debug it in TIBCO Designer by checking the output tab of the SendRequest activity in `op_insterparticipant` process.

Check the response file that has been created in the logs folder (path is provided for the global variable `cmi.log.path`).

You can also use the Log Viewer function in TIBCO Administrator to see the summary of this operation, including the failure causes.

To learn more about viewing audit logs, see [Chapter 5, Viewing Audit Logs](#), on page 43.

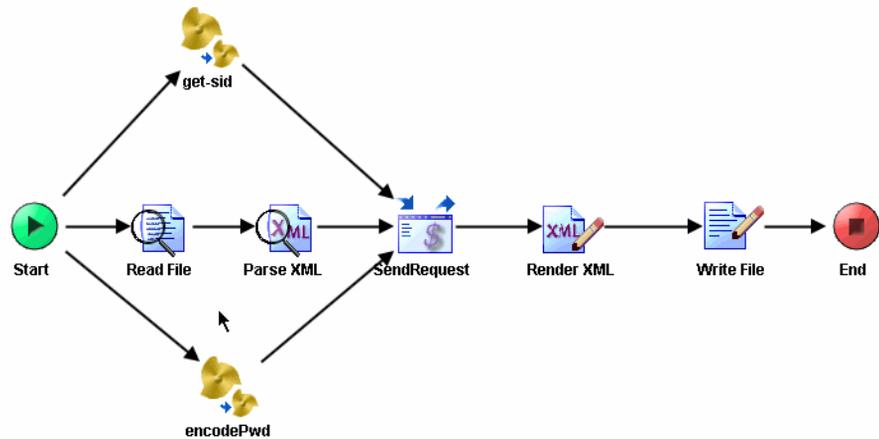
## Insert a Business Agreement

To create a new business agreement:

1. Select the Project tab in TIBCO Designer.
2. Select `project_name > INIT_REQ > write > op_insertbizagreement`.

The process for `insertbizagreement` opens in the design window.

Figure 9 Insert Business Agreement Process



3. Select the resource **ReadFile**.
4. Select the **Input** tab.

Figure 10 Read File for the Business Agreement



In the Activity Input window, the `fileName` parameter shows which sample file will be used for this operation: `INSERTBusinessAgreementREQ.xml` listed in `GlobalVariables/CMI/cmi.doc.path`.

5. To edit the sample XML file, go to `BC_HOME\protocols\cmi\examples\SampleDocs\INSERTBusinessAgreementREQ.xml`
6. Open the file with a text editor.

Example 4 `INSERTBusinessAgreementREQ.xml`

```
<?xml version="1.0" encoding="UTF-8" ?>
-<INSERTBusinessAgreementREQ xmlns="http://www.tibco.com/AX/BC/CMI" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.tibco.com/AX/BC/CMI../src/com/tibco/ax/cmi/resource/cmipayload.xsd">
<BizAgreement partner="Partner1" host="Host" validStart="2012-05-01" validEnd="2014-01-01" />
</INSERTBusinessAgreementREQ>
```

7. Modify the sample file so that it contains the appropriate information about the Host and the Partner (Host and Partner1)
8. Edit the dates when the agreement will start and end.
9. Save the sample file.
10. In TIBCO Designer, select the **Tester** tab.
11. Click the green triangle icon  next to Jobs.  
The Select Processes to Load dialog appears.
12. Select the process **op\_insertbizagreement** and click on **Load and Start Current**.
13. The process will be executed.
  - If the operation is successful, you will be able to see the Business Agreement Host-Partner1 added to the TIBCO Administrator GUI under **BusinessConnect > Business Agreements**.

Figure 11 Business Agreement added to the TIBCO Administrator GUI



- If the operation is not successful:

You can debug it in TIBCO Designer by checking the output tab of the SendRequest activity in op\_insertbizagreement process.

Check the response file that has been created in the logs folder (path is provided for the global variable `cmi.log.path`).

You can also use the Log Viewer function in TIBCO Administrator to see the summary of this operation, including the failure causes.

To learn more about viewing audit logs, see [Chapter 5, Viewing Audit Logs, on page 43](#).

## Get One Enabled Protocol

This operation will allow you to read the properties of an enabled protocol that belongs to

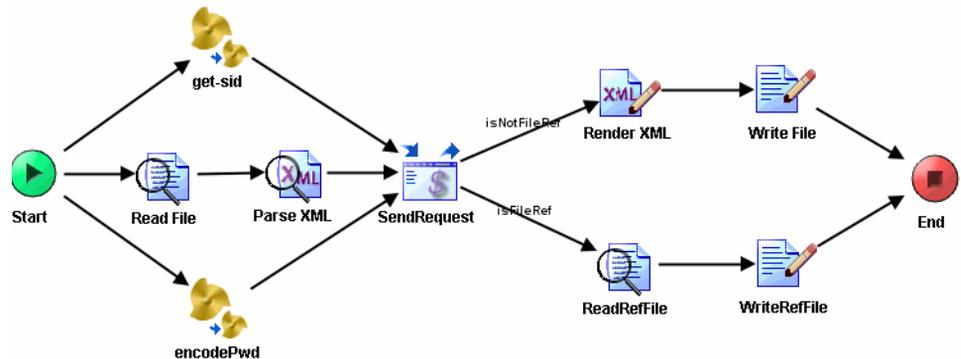
a participant. The response will be given as property key value pairs.

14. Select the Project tab in TIBCO Designer.

15. Select *project\_name* > **INIT\_REQ** > **read** > **op\_getenabledprotocol**.

The process for GetOneEnabledProtocol opens in the design window.

Figure 12 Get Enabled Protocol Process



16. Select the resource **ReadFile**.

17. Select the **Input** tab.

In the Activity Input window, the `fileName` parameter shows which sample file will be used for this operation: `GETONEEnabledProtocolREQ.xml` listed in `GlobalVariables/CMI/cmi.doc.path`.

18. To edit the sample XML file, go to `BC_HOME\protocols\cmi\examples\SampleDocs\GETONEEnabledProtocolREQ.xml`

19. Open the file with a text editor.

Figure 13 GETONEEnabledProtocolREQ.xml

```

<?xml version="1.0" encoding="UTF-8" ?>
- <GETONEEnabledProtocolREQ xmlns="http://www.tibco.com/AX/BC/CMI" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.tibco.com/AX/BC/CMI ../src/com/tibco/ax/cmi/resource/cmipayload.xsd">
  <participantName>Partner1</participantName>
  <protocol>EZComm</protocol>
</GETONEEnabledProtocolREQ>
  
```

20. If needed, modify the sample file so that it contains the appropriate information about the Protocol and the Partner, such as substituting the EZComm protocol with X12.

21. Save the sample file.

22. In TIBCO Designer, select the **Tester** tab.
23. Click the green triangle icon  next to Jobs.  
The Select Processes to Load dialog appears.
24. Select the process **op\_getenabledprotocol** and click on **Load and Start Current**.
25. The process will be executed.

If the operation is successful, the schema with values will be posted to the `ResponseFile`, which has been created in the logs folder (path is provided for the global variable `cmi.log.path`).

If the operation is not successful, you can debug it in TIBCO Designer by checking the Output tab of the `SendRequest` activity in the process `op_getenabledprotocol`.

Check the response file that has been created in the logs folder (path is provided for the global variable `cmi.log.path`). You can also use the Log Viewer in TIBCO Administrator to see the summary of this operation, including the causes of failure. To learn more about viewing audit logs, see [Chapter 5, Viewing Audit Logs, on page 43](#).

## Chapter 3 **Internal Application Users**

This chapter addresses Internal Application Users, which are created for CMI Protocol to manage trading partners and business agreements.

### Topics

---

- [Configure an Internal Application User, page 26](#)

## Configure an Internal Application User

---

To work with the CMI Protocol, you need to configure a specific type of administrative user called *Internal Application User*. These administrative users are created based on the organization needs and they have full SuperUser privileges to perform operations with trading hosts, partners and business agreements.

Each Internal Application User can be configured with permissions to handle only the administrative tasks appropriate for that user.

To create a new Internal Application User:

1. Using TIBCO Administrator, select **BusinessConnect > User Management > Users**.  
Only one user, the TIBCO Administrator User (admin) is present by default under the Admin tab..
2. Select the Internal tab.
3. Click **Add**.
4. In the Set User Name dialog, enter the name for this new user.
5. Click **OK**.

In the New User: *New User* dialog, enter configuration data in the three tabs as explained in [Table 1](#).

*Table 1 Configuring the New Internal Application User*

Field	Description
<b>General Tab</b>	
User Name	Leave the name given in Step 5, or change the name.
Password	Set the password for the new Internal Application User.



Group Membership and Permissions tab options don't need to be configured, since an Internal User is always added as a super user by default.

## Chapter 4 **Transactions**

This chapter explains types of operations and transactions used by CMI Protocol.

It also explains how to use the Operations Editor to display the operations that have been preconfigured for this release.

### Topics

---

- [Operation Overview, page 28](#)
- [Transaction Overview, page 36](#)
- [Private Process Smart Routing, page 38](#)
- [Using the Operations Editor, page 39](#)

## Operation Overview

---

CMI Protocol manages trading partners data in TIBCO BusinessConnect using operations that consist of a Request.Response pair.

### Operation Categories

CMI Protocol operations can be categorized in different ways:

- **By type:** Operations are categorized into four types:
  - Session: See [Sessions on page 36](#)
  - Read: See [Read Operations on page 31](#)
  - Write: See [Write Operations on page 33](#)
  - Batch: See [Batch Transactions on page 37](#)
- **By target:** Except for Session and Batch type, each operation has a “target” attribute, which is a particular type that the operation works on, such as Participant, Business Agreement, and so on.
- **By command:** Each operation falls into one of the pre-defined commands, as explained in [Table 2](#).

*Table 2 CMI Protocol Commands*

Command	Definition
INSERT	<p>Insert new data. If the data already exists, an error message is returned.</p> <p>Used only for the Write operation. See also <a href="#">Table 3, Command Rules for the Parent/Child Element</a>.</p>
UPDATE	<p>Update existing data. If the data does not exist, an error message is returned.</p> <p>Used only for the Write operation. See also <a href="#">Table 3, Command Rules for the Parent/Child Element</a>.</p>
UPSERT	<p>Update or insert data. CMI Protocol will try to insert the data if it does not exist, or update the data if it exists. In either case, a success response is returned.</p> <p>Used only for the Write operation. See also <a href="#">Table 3, Command Rules for the Parent/Child Element</a>.</p>

Table 2 CMI Protocol Commands

Command	Definition
DELETE	<p>Delete data. If the data does not exist, an <code>Error</code> message is returned.</p> <p>Used only for the <code>Write</code> operation. See also <a href="#">Table 3, Command Rules for the Parent/Child Element</a>.</p>
GETONE	<p>Get one data. If no data exists, an empty response will be returned. If more than one data exists, the first matched data will be returned.</p> <p>Used only for the <code>Read</code> operation.</p>
GETMANY	<p>Get a set of data. If no data exists, an empty response will be returned.</p> <p>Used only for the <code>Read</code> operation.</p>
LOGIN	<p>Log in and create a new session.</p> <ul style="list-style-type: none"> <li>• If authentication check is passed, a session resource is created and a response containing the new generated session ID is returned.</li> <li>• If authentication check is not passed, a new session ID is still internally generated for audit log purpose, but an error message is returned and no session resources are created.</li> </ul>
LOGOUT	<p>Logout and destroy session.</p> <ul style="list-style-type: none"> <li>• If the target session resource exists, any session resources are destroyed and success response returned.</li> <li>• If the target session resource does not exist, an error message is returned.</li> </ul>
EXECUTE	<p>Execute a batch of requests.</p> <ul style="list-style-type: none"> <li>• If the given <code>batchId</code> does not exist (may be already executed or cancelled), an error message is returned.</li> <li>• If the given <code>batchId</code> does exist, CMI Protocol will always try to execute the requests of the batch and return a success response, even if all requests fail.</li> <li>• Transactions are executed based on the timestamp and in the order they have been received.</li> </ul>
CANCEL	<p>Cancel a batch of requests.</p> <ul style="list-style-type: none"> <li>• If the given <code>batchId</code> does not exist (may be already executed or cancelled), an error message is returned.</li> <li>• If the given <code>batchId</code> does exist, CMI Protocol will try to delete all stored data and return a success response.</li> </ul>

## Command Rules for the Parent/Child Element

One example of the operation `ParticipantRequest` is described in the [Example 5](#), where the Parent element `UPDATEParticipantREQ` is highlighted in *Italic*, and the children (or grand children) are highlighted in **Bold**.

*Example 5 Command Rule Example for the Parent/Child Element*

---

```

<UPDATEParticipantREQ>
  <Participant name="partner1" type="Partner">
    <Locations>
      <Location name="loc1" addr1="newaddr" />
      <Contacts>
        <Contact name="Mark" command="INSERT" />
        <Contact name="Mary" command="DELETE" />
        <Contact name="John" />
      </Contacts>
    </Location>
  </Locations>
</Participant>
</UPDATEParticipantREQ>

```

---

Rules that follow the commands performed on the Parent and Child elements are described in [Table 3](#).

*Table 3 Command Rules for the Parent/Child Element*

Commands	INSERT	UPSERT	UPDATE	DELETE
Commands on Parent Element	Only this operation is allowed when the parent or root element are executing the insert command.	Not allowed for children when the parent executes the insert command.	Not allowed for children when the parent executes the insert command.	Not allowed for children when the parent executes the insert command.
	Allowed. Inserting a child element adds the contents specified while the parent is executing the UPSERT command.	Allowed. UPSERT on a child element with the UPSERT command on the parent either inserts the child (if not present) or updates the child details appropriately.	Allowed. UPDATE on a child element with the UPSERT command on the parent updates only the child details appropriately. If the child details are not present, an Error is thrown.	Allowed. DELETE on a child element with the UPSERT command on the parent deletes the child data appropriately.

---

Table 3 Command Rules for the Parent/Child Element

Commands	INSERT	UPSERT	UPDATE	DELETE
Commands on Parent Element	Allowed. Inserting a child element adds the contents specified while the parent is executing an UPDATE command.	Allowed. UPSERT on a child element with the UPDATE command on the parent either inserts the child (if not present), or updates the child details appropriately.	Allowed. UPDATE on the child element with the UPDATE command on the parent updates only the child details appropriately. If the child details are not present, an Error is thrown.	Allowed. DELETE on a child element with the UPDATE command on the parent deletes the child data appropriately
	Not recommended for the category children when the parent does the delete command.	Not recommended for the category children when the parent does the delete command.	Not recommended for children when the parent does the delete command.	Allowed

## Read Operations

Read operations follow the rules explained in [Table 4](#).

Table 4 Read Operation Rules

Read Operation	Rules and Descriptions
getOneAuditLog	<p>Search criteria to retrieve only one exact result when the required parameters protocolName, startDate, and endDate are provided. Rules are:</p> <ul style="list-style-type: none"> <li>Criteria can also be set by field names of that protocol, values to search, or based on operation condition such as contains, is, is not and is not like.</li> <li>Criteria can also be set based on status and host attributes that are not required.</li> <li>Results are returned as a name-value attribute pairs to be protocol agnostic.</li> </ul>
getOneBusinessAgreement	<p>Search is based on the name attribute value of the root element GETONEBusinessAgreementREQ.</p> <p>Rules are:</p> <ul style="list-style-type: none"> <li>Other attributes or elements are ignored.</li> <li>Sorting is not supported.</li> </ul>

Table 4 Read Operation Rules

Read Operation	Rules and Descriptions
getOneEnabledProtocol	<p>This operation allows the user to read the enabled protocol properties of a participant.</p> <p>The response is given as property key value pairs.</p>
getOneOperationBinding	<p>This operation allows the user to read the properties on a business agreement operation binding.</p> <p>The response is given in a form of property key value pairs.</p>
getOneOperationTree	<p>This operation allows the user to read the properties on an operation tree node.</p> <p>The response is given in a form of property key value pairs.</p>
getOneParticipant	<p>Search is based on the name attribute value of the root element GETONEParticipantREQ.</p> <p>Rules are:</p> <ul style="list-style-type: none"> <li>• Other attributes or elements are ignored</li> <li>• Search is always accurate and returns one GETONEParticipantRESP response.</li> </ul> <p>Searching for a participant based on a domain ID or AS2 ID yield results only when this participant has a domain ID or AS2 ID assigned to it. This search doesn't work for protocols that do not support domain IDs or AS2 IDs. For these situations, only search by name is applicable.</p>
getManyAuditLog	<p>Search criteria to retrieve one or more results when the required parameters protocolName, startDate, and endDate are provided. Rules are:</p> <ul style="list-style-type: none"> <li>• How many rows are returned can be restricted by size attribute.</li> <li>• Criteria can also be set by field names of that protocol, values to search, or based on operation condition like contains, is, is not and is not like.</li> <li>• Criteria can also be set based on status and host attributes that are not required</li> <li>• Results are returned as name-value attribute pairs to be protocol agnostic.</li> </ul>

Table 4 Read Operation Rules

Read Operation	Rules and Descriptions
getManyBusinessAgreement	<p>Search is based on the name attribute value of the root element GETMANYBusinessAgreementREQ. Rules are:</p> <ul style="list-style-type: none"> <li>• Other attributes or elements are ignored.</li> <li>• Wildcard search is allowed and based on partner names; for example, <code>Company*</code> will return all business agreements containing this string, such as <code>Company1-Company2 businessagreement</code>, where <code>Company1</code> is the Host and <code>Company2</code> is the Partner.</li> <li>• Sorting is not supported.</li> </ul>
getManyParticipant	<p>Search is based on the name attribute value of the root element GETMANYParticipantREQ. Rules are:</p> <ul style="list-style-type: none"> <li>• Other attributes or elements are ignored.</li> <li>• Wildcard search is allowed; for example, the string <code>*y*</code> would return all participants containing <code>*y*</code>, such as <code>Company1</code>.</li> <li>• Sorting is not supported.</li> </ul> <p>Searching for a participant based on a domain ID or AS2 ID yield results only when this participant has a domain ID or AS2 ID assigned to it. This search doesn't work for protocols that do not support domain IDs or AS2 IDs. For these situations, only search by name is applicable.</p>
getManyProtocolOperation	<p>This operation returns all operations for a particular protocol. Rule is:</p> <ul style="list-style-type: none"> <li>• Multiple protocol operations can be returned by specifying multiple protocols.</li> </ul>

## Write Operations

This operations category defines any operations that store or delete participants or business agreements in the configuration database.

The WRITE operations will perform the following:

- Allow creation of new participants and business agreements
- Delete existing participants or business agreements
- Update participants or business agreements
- Update enabled protocol specific properties and Operations Tree properties

Write operations follow the rules explained in [Table 5](#).

Table 5 Write Operation Rules

Write Operation	Rules and Descriptions
deleteBusinessAgreement	This operation allows the user to delete an existing business agreement.
deleteParticipant	This operation allows the user to delete an existing participant.
insertBusinessAgreement	This operation allows the user to insert a new business agreement.
insertParticipant	This operation allows the user to insert a new participant.
updateBusinessAgreement	This operation allows the user to update an existing business agreement.
updateEnabledProtocol	<p>This operation allows the user to set enabled protocol properties for a participant. The input schema for this operation is constructed when the user clicks on the button 'Update from Configuration Store' on the BusinessConnect connection shared resource. The schema takes into consideration all activated protocols in the configuration store.</p> <p>On runtime, several checks are performed to verify whether the following is true:</p> <ul style="list-style-type: none"> <li>• The specified participant exists.</li> <li>• The participant is a partner if the 'partnerProtocols' element has been specified.</li> <li>• The participant is a host if the 'hostProtocols' element has been specified.</li> <li>• The protocols underneath, 'partnerProtocols' or 'hostProtocols', have been enabled for the participant.</li> </ul>
updateOperationBinding	<p>This operation allows the user to set the properties on a business agreement operation binding. The input schema for this operation is constructed when the user clicks 'Update from Configuration Store' on the BusinessConnect connection shared resource. The schema takes into consideration all activated protocols in the configuration store.</p> <p>On runtime, the following checks are performed to verify whether:</p> <ul style="list-style-type: none"> <li>• The business agreement exists.</li> <li>• The operation path corresponds to the request schema selected for the protocol and level.</li> </ul>

Table 5 Write Operation Rules

Write Operation	Rules and Descriptions
updateOperationTree	<p>This operation allows the user to set the properties on an operation tree node. The input schema for this operation is constructed when the user clicks 'Update from Configuration Store' on the BusinessConnect connection shared resource. The schema takes into consideration all activated protocols in the configuration store.</p> <p>On runtime, a check is performed to verify that the operation path corresponds to the request schema selected for the protocol and level.</p>
updateParticipant	This operation allows the user to update an existing participant
upsertBusinessAgreement	This operation allows the user to update an existing business agreement or insert a new one.
upsertParticipant	This operation allows the user to update an existing participant or insert a new one.

## Transaction Overview

---

The CMI Protocol interacts with TIBCO BusinessConnect private processes. Two types of transactions can occur based on their execution time:

- **Real-time transactions:** When real-time transactions are used, all requests will be immediately executed by the CMI Protocol engine. The real-time transactions between private processes and CMI Protocol are called *sessions*.

For more information, see [Sessions on page 36](#).

- **Batch transactions:** When batch transactions are used, all data access requests are initially stored by the CMI Protocol engine. Later, the batch can be executed or cancelled by sending a special request from the private process or via the message queue logs from the Log Viewer.

For more information, see [Batch Transactions on page 37](#).

## Sessions

There are two types of sessions:

- **Explicit sessions:** During an explicit session, login or logout requests are initiated in order to start or to terminate a session. Each login request contains client's identity information. Once this identity is accepted and session established, any following requests carry only the `sessionID`.
- **Implicit sessions:** During an implicit session, each request is initiated with the client's identity information. CMI Protocol creates such session on-the-fly, executes operations, and terminates the session once the response is sent.
- **Session Timeout:** For security reasons, each user session will time out, and each `sessionID` will become invalid after a configurable time period.

If a session times out, CMI Protocol updates the audit log entry to status `COMPLETED WITH ERRORS`, sends an `ADVISORY` message to the private process, and removes the session content. Any following incoming requests using the timed-out `sessionID` will trigger an error message.

## Edit Plug-in Properties - Session Expiry Settings

To define the time interval after which a session will expire:

1. Select **BusinessConnect > System Settings > Activated Protocol Plug-ins and Properties**.
2. Click on the **CMI** link.

3. In the dialog **Edit Plug-in Properties**: CMI enter the time interval for session expiry in the window `cmi.session.expiry`. Default is 1800 seconds.
4. Click **Save** to save the session settings.

## Batch Transactions

CMI Protocol will interpret a transaction request to be of the type batch if a property `batchID` is defined in `AESchema`. All requests with the same `batchID` will be executed one by one, according to the sequence in which they have been received. Batch transactions follow these rules:

- Explicit sessions are not allowed: all batch requests use only implicit sessions.
- Currently, batch transactions are allowed to write data, but not to read.
- Allowed actions for batch transactions are `EXECUTE` and `CANCEL` (used to execute or cancel a batch execution).
- Stored batch requests will never time out or get deleted, unless an explicit request is received. Such request can be either a private message, or it can be triggered from the GUI.
- Upon receiving a batched data access request, CMI Protocol executes validation tasks involved with parsing payload: validating the private message header, user logins and passwords; validating the payload against the XML schema; and so on. These basic validation tasks won't be repeated at execution time.

For more details, see [Private Messages in Batch Transactions](#), page 58.

## Message Queues

As an alternative to executing or cancelling batch transaction using private messages, you can use the Message Queue GUI in TIBCO Administrator. When a batch is process of execution and cancellation, cancel or execute cannot be done.

For each `batchID`, you will be able to see a record with information such as how many requests are in the batch. You can then decide whether to execute or cancel such a batch.

## Private Process Smart Routing

---

The CMI Protocol supports private process smart routing, same as other protocols that are based on TIBCO BusinessConnect.

To learn more, see *TIBCO BusinessConnect Trading Partner Administration*, Chapter 4, System Settings, Private Process Smart Routing.

However, when working with this protocol some common properties defined for TIBCO BusinessConnect smart routing rules, such as `from` or `to`, have a new meaning or constraints:

- `protocol`: the string CMI
- `from`: the string CMI
- `to`: the userID used to log in and execute the operation
- `direction`: fixed to Outbound

In order to catch error messages when errors occur, smart routing rules have to be set as explained in [Table 6](#).

*Table 6 Private Process Smart Routing Rules*

Scenarios	From	To	Operation	Direction
Session does not exist	CMI	No value	Current operation	Outbound
Session timeout	CMI	userId of this session	Session Timeout	Outbound

# Using the Operations Editor

## Import Operations



In this release, preconfigured operations are automatically imported when the CMI Protocol is activated. You need to verify that the protocol is activated.

Properties of these preconfigured operations are read only and you cannot update or delete them.

See *TIBCO BusinessConnect ConfigStore Management Interface Protocol Installation, Protocol Activation*.

## Display the Imported Operations

To view the available operations for CMI Protocol, start the Operations Editor:

1. Select **BusinessConnect > Operations Editor**.
2. Click on the **CMI** link.
3. Click on the “+” sign in the upper left corner to display all available transactions that come predefined with CMI Protocol.

All available operations, grouped by their corresponding categories, will be displayed as shown in [Table 7](#).

Detail schemas for each of the operations is provided in `BC_HOME\protocols\emi\examples\SampleDocs\emipayload.xsd`.

*Table 7 CMI Protocol Transactions*

Field	Explanation
<b>Version</b>	
1.0	Version number. In this release, you cannot add new versions.
<b>Category: Batch</b>	
cancelBatch	Cancel a batch.  See also <a href="#">CANCEL on page 29</a> .

Table 7 CMI Protocol Transactions

Field	Explanation
executeBatch	Execute a batch. See also <a href="#">EXECUTE</a> on page 29.
<b>Category: Read</b>	
getManyAuditLog	Get a set of audit logs. See also <a href="#">GETMANY</a> on page 29.
getManyBusinessAgreement	Get a set of business agreements. See also <a href="#">GETMANY</a> on page 29.
getManyParticipant	Get a set of participants. See also <a href="#">GETMANY</a> on page 29.
getManyProtocolOperation	Get a set of operations for a particular protocol. See also <a href="#">GETMANY</a> on page 29.
getOneAuditLog	Get one audit log. See also <a href="#">GETONE</a> on page 29.
getOneBusinessAgreement	Get one business agreement. See also <a href="#">GETONE</a> on page 29.
getOneEnabledProtocol	Get one protocol that has been enabled. See also <a href="#">GETONE</a> on page 29.
getOneOperationBinding	Get one operation binding. See also <a href="#">GETONE</a> on page 29.
getOneOperationTree	Get one operation tree. See also <a href="#">GETONE</a> on page 29.
getOneParticipant	Get one participant. See also <a href="#">GETONE</a> on page 29.
<b>Category: Session</b>	

Table 7 CMI Protocol Transactions

Field	Explanation
login	Login and create a new session. See also <a href="#">LOGIN on page 29</a> .
logout	Logout and destroy session. See also <a href="#">LOGOUT on page 29</a> .
<b>Category: Write</b>	
deleteBusinessAgreement	Delete the business agreement. See also <a href="#">DELETE on page 29</a> .
deleteParticipant	Delete the participant. See also <a href="#">DELETE on page 29</a> .
insertBusinessAgreement	Insert a new business agreement. See also <a href="#">INSERT on page 28</a> .
insertParticipant	Insert a new participant. See also <a href="#">INSERT on page 28</a> .
updateBusinessAgreement	Update the existing business agreement. See also <a href="#">UPDATE on page 28</a> .
updateEnabledProtocol	Update the existing enabled protocol. See also <a href="#">UPDATE on page 28</a> .
updateOperationBinding	Update the existing operation binding. See also <a href="#">UPDATE on page 28</a> .
updateOperationTree	Update the existing operation tree. See also <a href="#">UPDATE on page 28</a> .
updateParticipant	Update the existing participant. See also <a href="#">UPDATE on page 28</a> .
upsertBusinessAgreement	Update or inserts a business agreement. See also <a href="#">UPSERT on page 28</a> .

*Table 7 CMI Protocol Transactions*

Field	Explanation
upsertParticipant	Updates or inserts a participant. Unlike in the GUI, creating a new Participant of Host type would result in all the protocols being disabled, unless they are explicitly enabled from the CMI Protocol requests.  See also <a href="#">UPSERT on page 28</a> .

---

## Chapter 5 **Viewing Audit Logs**

This chapter explains how to use and interpret the audit logs for CMI Protocol.

### Topics

---

- [Log Viewer Overview, page 44](#)
- [Perform a Log Search, page 45](#)
- [Viewing Search Results, page 50](#)

## Log Viewer Overview

---

Only audit logs are supported for the CMI Protocol (non-repudiation and resend logs are not supported).

When you are using the CMI Protocol, the log viewer allows you to search and view information from the audit logs, which are used to store information about the messages and events.

Some of the types of information stored in the audit log include:

- Sent and received documents
- Document originator
- User name
- Time stamp
- TransactionID
- BatchID (if batched)
- SessionID
- OperationID
- Processing status
- Validation errors

In each audit log, you can search for specific log entries and you can save and reuse queries.

You can search all logs based on the following criteria:

- Protocol
- Host
- Status
- Date Range
- Advanced search by: session ID, user name, last operation ID, and batch ID.



When doing searches, keep in mind that the character “\*” is not considered to work as a wild card, but it represents a part of a name instead.

## Perform a Log Search

1. Using TIBCO Administrator, select **BusinessConnect>Log Viewer**.
2. To choose which type of log to view, select the radio button next to CMI. The following log searches will be available:
  - Audit
  - Message Queue
  - Preferences

### Audit Logs

To perform the audit log search:

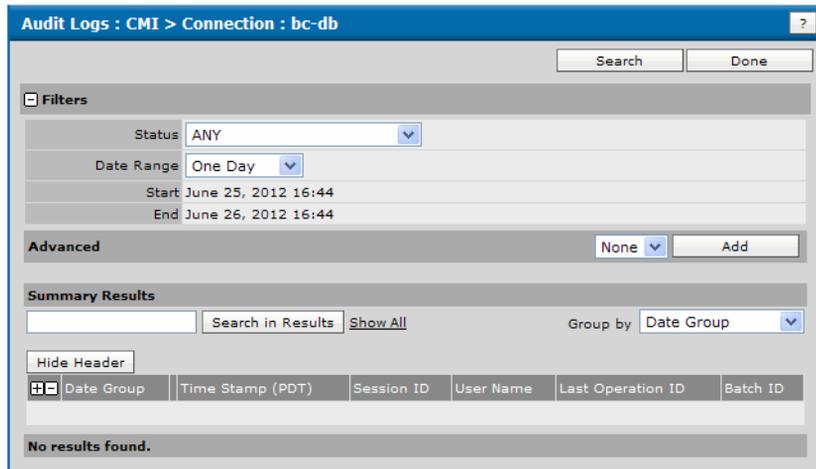
1. Click the Audit tab.

The Audit Logs dialog appears with the default settings:

Status ANY

Date Range One Day

*Figure 14 Audit Log*



- Click the **Search** button to use the default search settings, or configure the search using the information from [Table 8](#).

Table 8 Search Transaction

Field	Description
<b>Search Transactions</b>	
Status	<p>Find log entries for transactions that ended with a specific status:</p> <ul style="list-style-type: none"> <li><b>ANY</b> -All transactions for this installation</li> <li><b>BATCHED</b> - Status to indicate and retrieve all batched requests that need to be executed or cancelled.</li> <li><b>COMPLETED</b> - Status to indicate that the CMI Protocol requests have been completed either for Implicit or for Explicit session requests.</li> <li><b>COMPLETED WITH ERRORS</b> - This status is used to query for sessions that have timed out.</li> <li><b>ERROR</b> - Transactions that failed.</li> <li><b>REQ_COMPLETED</b> - Transaction that completed successfully.</li> <li><b>USER_VERIFIED</b> - Status useful to be queried during an explicit session. It indicates that the user was verified and that neither new requests have been received by the runtime engine, nor the session was terminated.</li> </ul>
Date Range	The Custom option allows you to specify a range with a specific Start and End Date Time. The Predefined option allows you to specify a range with as follows: One Day, One Week, One Month and One Year.
<b>Advanced</b>	
Add	<p>Add advanced filters to define the criteria such as: Session ID, User Name, Last Operation ID, and Batch ID. Each of these variables can be searched by choosing one of the following options from the dropdown list: is, contains, is not, is not like.</p> <p>While you can perform a search without adding a query, it will save you time in the future if you set up queries. Keep in mind that a query can be used again <i>only</i> if it is saved under a specific name; if you fill all required query details and click Save without providing a query name, such query will be performed as an advanced query but cannot be re-used.</p> <p>Once the filter is specified, select it in the drop-down list for executing the search, or click <b>Edit</b> to change the filter criteria..</p>

- Click **Save**.

Only the log entries that meet all of the criteria you specify will be returned.

## Message Queue

To perform the message queue log search:

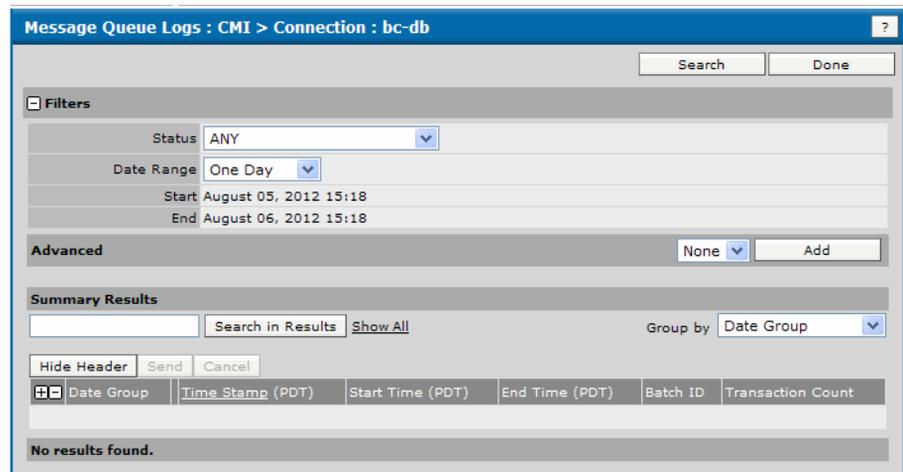
1. Click the Message Queue tab.

The Message Queue Logs dialog appears with the default settings:

Status ANY

Date Range One Day

Figure 15 Message Queue Log



Message Queue Logs : CMI > Connection : bc-db

Search Done

**Filters**

Status ANY

Date Range One Day

Start August 05, 2012 15:18

End August 06, 2012 15:18

**Advanced** None Add

**Summary Results**

Search in Results Show All Group by Date Group

Hide Header Send Cancel

Date Group	Time Stamp (PDT)	Start Time (PDT)	End Time (PDT)	Batch ID	Transaction Count
No results found.					

- Click the **Search** button to use the default search settings, or configure the search using the information from [Table 9](#)

Table 9 Message Queue Logs

Field	Description
<b>Search Transactions</b>	
Status	<p>Find log entries for transactions that ended with a specific status:</p> <ul style="list-style-type: none"> <li><b>ANY</b> -All transactions for this installation</li> <li><b>BATCHED</b> - Status to indicate and retrieve all batched requests that need to be executed or cancelled.</li> <li><b>COMPLETED</b> - Status to indicate that the CMI Protocol requests have been completed either for Implicit or for Explicit session requests.</li> <li><b>COMPLETED WITH ERRORS</b> - This status is used to query for sessions that have timed out.</li> <li><b>ERROR</b> - Transactions that failed.</li> <li><b>REQ_COMPLETED</b> - Transaction that completed successfully.</li> <li><b>USER_VERIFIED</b> - Status useful to be queried during an explicit session. It indicates that the user was verified and that neither new requests have been received by the runtime engine, nor the session was terminated.</li> </ul>
Date Range	The Custom option allows you to specify a range with a specific Start and End Date Time. The Predefined option allows you to specify a range with as follows: One Day, One Week, One Month and One Year.
<b>Advanced</b>	
Batch ID Transaction Count	<p>Each of these variables can be searched by choosing one of the following options from the dropdown list:</p> <ul style="list-style-type: none"> <li>is (an exact match is required)</li> <li>contains (any part of the string that matches is returned)</li> <li>is not (the returned results do not match the value that was searched for)</li> <li>is not like (returns result that does not match the value)</li> </ul>

- Once the batch ID or the transaction count information has been defined, click **Search**.

## Send or Cancel Messages in the Log Queue

From the left-most column of the log summary view, you can select transactions to send or cancel:

- Check the checkbox and click the Send button.
- Check the checkbox and click the Cancel button.

## Viewing Search Results

Several views are accessible from the search results.

### Summary View

When you perform a search in any log, the results are returned in a table. Each entry in the table represents a transaction.

1. After defining the search criteria, click on **Search**.
2. The Search Summary window appears.

Figure 16 Summary Search View

Date Group	Time Stamp (PDT)	Session ID	User Name	Last Operation ID	Batch ID
YESTERDAY	Jul-23-2012 04:17:38 PM	CMI-admincmi-5FK94P-9AFKPP-00071L-HK7BLM-01P2	admincmi	1.0/Write/insertParticipant	
YESTERDAY	Jul-23-2012 04:15:17 PM	CMI-admincmi-5FK94P-9AFKPP-00064S-946WN5-01G6	admincmi	1.0/Write/insertParticipant	
YESTERDAY	Jul-23-2012 04:14:14 PM	CMI-admincmi-5FK94P-9AFKPP-00029K-QM1S21-0185	admincmi	1.0/Write/insertParticipant	
YESTERDAY	Jul-23-2012 04:01:41 PM	CMI-admincmi-5FK94P-9AFKPP-00086W-KQ3UP9-00R2	admincmi	1.0/Write/insertParticipant	

To view the details of a transaction, click the active document icon  in the left-most column of a transaction in the Summary View.

Figure 17 Transaction Detail View

**Transaction Details** ?

**Filters > Status : ANY > Jul-17-2012 18:18 ~ Jul-24-2012 18:18**

**Summary : 1 of 4**

Session ID	CMI-admincmi-5FK94P-9AFKPP-00071L-HK7BLM-01P2
User Name	admincmi
Last Operation ID	1.0/Write/insertParticipant
Batch ID	

**States** [change view](#)

	Time Stamp ↓	Operation ID	Transaction ID	State	Status	Description
	Jul-23-2012 04:17:37 PM	1.0/Write/insertParticipant	5FK94P-9AFKPP-00071L-H6EZFU-01P0	RECEIVED_REQUEST	REQ_STARTED	Initiator request message was received.
	Jul-23-2012 04:17:37 PM	1.0/Write/insertParticipant	5FK94P-9AFKPP-00071L-H6EZFU-01P0	AUTHENTICATED	USER_VERIFIED	User authenticated successful for user : admincmi.
	Jul-23-2012 04:17:37 PM	1.0/Write/insertParticipant	5FK94P-9AFKPP-00071L-H6EZFU-01P0	VALIDATION	PROCESSED	Payload schema validation successful.
	Jul-23-2012 04:17:38 PM	1.0/Write/insertParticipant	5FK94P-9AFKPP-00071L-H6EZFU-01P0	SENT_RESPONSE	REQ_COMPLETED	Response message sent.
	Jul-23-2012 04:17:38 PM	1.0/Write/insertParticipant	5FK94P-9AFKPP-00071L-H6EZFU-01P0	PKGD_RESPONSE	RESP_PACKAGED	Response message packaged successful.

- To move between various transactions, use the buttons **Back** and **Next**.
- To view the details of a state, click the active document icon in the left-most column of an entry in the Transaction Details View.

Figure 18 State Detail View

**Transaction Details** ?

**Filters > Status : ANY > Jul-17-2012 18:18 ~ Jul-24-2012 18:18**

**Summary : 1 of 4**

Session ID	CMI-admincmi-5FK94P-9AFKKP-00071L-HK7BLM-01P2
User Name	admincmi
Last Operation ID	1.0/Write/insertParticipant
Batch ID	

**State : 1 of 5** [change view](#)

Time Stamp	Jul-23-2012 04:17:37 PM
Operation ID	1.0/Write/insertParticipant
Transaction ID	5FK94P-9AFKKP-00071L-H6EZFU-01P0
State	RECEIVED_REQUEST
Status	REQ_STARTED
Description	Initiator request message was received.

## Saving and Reusing Queries

---

In the context of the TIBCO BusinessConnect log viewer, a *search* is a one-time occurrence and a *query* is a search that one has saved for reuse. Upon saving a search, it becomes a named query and is available from the Queries drop-down list.

### Save a Query

To save a query, perform these steps:

1. Click the **Add** button to define the search criteria under the Advanced Filters section.
2. Define the search criteria in the Advance Filers dialog.
3. Specify a name for the optional field Save As Query.
4. Click **Save**.



The saved query does not include the settings for the database connection and the date range criteria.

### Reuse a Query

To reuse a query, perform these steps:

1. Select the desired query in the Queries dropdown list. The Advanced Filters are restored from the selected saved query.
2. Click **Search**.



## Chapter 6 **Private Process Communications**

This chapter describes private message formats in CMI Protocol transactions.

### Topics

---

- [Private Message Categories, page 56](#)
- [Private Messages in Sessions, page 57](#)
- [Private Messages in Batch Transactions, page 58](#)
- [Initiator Request, page 60](#)
- [Initiator Response, page 62](#)
- [Advisory, page 64](#)
- [Error, page 65](#)

## Private Message Categories

---

The private process messages used for exchange in CMI Protocol belong to these categories:

- **INITIATOR.REQUEST**

This message is sent from the private process to CMI Protocol.

For more details, see [Initiator Request on page 60](#).

- **INITIATOR.RESPONSE**

CMI Protocol sends back an INITIATOR.RESPONSE message indicating that:

- The batch request has been received, but that it has not yet been executed.
- Execution of the session succeeded.
- Execution of the session failed: this response includes an ERROR message with the error code details.

For more details, see [Initiator Response on page 62](#).

- **ADVISORY**

This message can have special status such as batch execution, cancellation, and so on.

Batch transactions can be either executed or cancelled (see [Private Messages in Batch Transactions on page 58](#)). Once cancellation is completed, an ADVISORY message is sent indicating success of cancellation. It is also sent whenever there are errors in processing the requests or when session times out

For more details, see [Advisory on page 64](#).

- **ERROR**

Error message, which indicates the error status or failures of the CMI Protocol request, is sent from the private process.

For more details, see [Error on page 65](#).

## Private Messages in Sessions

---

When real-time transactions are used, all requests will be immediately executed by the CMI Protocol engine. The real-time transactions between private processes and CMI Protocol are called *sessions*.

For more information, see [Sessions on page 36](#).

When you use private processes in sessions, the following will happen:

1. The private process sends an INITIATOR.REQUEST message to CMI Protocol.
2. CMI Protocol responds in one of these three ways:
  - **Execution succeeded:** It sends an INITIATOR.RESPONSE message back.
  - **Execution failed:** It sends an ERROR message and an INITIATOR.RESPONSE message including error code and error details.
  - **Session timed out:** Session Timeout Advisory is triggered by the TIBCO BusinessConnect engine after the time has passed.

If the sessionID has already timed out at the time when a request comes in, TIBCO BusinessConnect returns a normal ERROR message with the code that points to the invalid sessionID.

## Private Messages in Batch Transactions

---

When batch transactions are used, all data access requests are initially stored by the CMI Protocol engine. Later, if a private processes used to trigger a batch, it will send a separate special request to either execute or to the cancel the batch. Another way to trigger a batch is through a message queue log from a Log Viewer.

For more in formation, see [Batch Transactions on page 37](#).

When you use private processes for batch transactions, the following will happen:

1. A Private Process sends an `INITIATOR.REQUEST` message to CMI Protocol.
2. After performing a message validation, CMI Protocol will store the data in database tables and send back an `INITIATOR.RESPONSE` message indicating that the batch request has been received, but that it has not yet been executed.

If the request is not valid, such as if it failed to pass the message validation, CMI Protocol sends an `ERROR` message and an `INITIATOR.RESPONSE` message that includes the error code and details.



A batch transaction can be executed or cancelled from two places: from the private processes (using TIBCO Designer) or from the TIBCO Administrator GUI. To use the Message Queue function in the GUI, see [Message Queues on page 37](#).

When a batch is triggered from a message queue logs, only an Advisory message is published to mark the successful completion of the batch request.

3. The Private process now sends a special `INITIATOR.REQUEST` message to CMI Protocol. The operation type of this message is `batch` and the action type is either `execute` OR `cancel`.
  - If the request is valid (the batch is available to execute or cancel), CMI Protocol sends an `INITIATOR.RESPONSE` message for the special request, indicating that it will start to execute or cancel the request.
  - If the request is not valid (such as when the batch does not exist or has been executed), an `ERROR` message will be sent. In addition, an `INITIATOR.RESPONSE` message is sent that includes the error code and details.
4. Regardless of where the batch was triggered (from the GUI or from the Private process), the following messages indicate results:
  - **Execute Request:** CMI Protocol starts to iterate stored requests and execute them. Upon successful execution of each single request, no message will be sent. However upon failure, an `ERROR` message is sent and execution continues to the

next request. No INITIATOR.RESPONSE message is sent, since it was already sent upon receiving the request.

Once all requests have been executed, an ADVISORY message is sent that includes a descriptive summary information, such as how many requests succeeded, how many request failed, when the batch started to execute, and when the execution was completed.

- **Cancel Request:** CMI Protocol starts to delete all stored requests.

Since the Cancel action for any single request could never fail, no ERROR or INITIATOR.RESPONSE messages is sent. An ERROR messages will be sent only to indicate fatal errors, such as when the database connection failed. In a case of a fatal error, an ERROR message is sent and processing is stopped.

A Cancel request is invalid for a batch that is currently being executed.

Once cancellation is completed, an ADVISORY message is sent indicating success of cancellation.

## Initiator Request

---

This message is sent from the private process to CMI Protocol.

**Rendezvous Subject Name**     *prefix.installation.standardID*.INITIATOR.REQUEST  
 Example: AX.BC.BC-ACME.CMI.INITIATOR.REQUEST

**JMS Queue Name**     *prefix.installation*.INITIATOR.REQUEST  
 Example: AX.BC.BC-ACME.INITIATOR.REQUEST

**Message Name**     InitiatorRequest

Table 10 InitiatorRequest

Field	Type	Required	Description
standardID	String	yes	The string CMI
transactionID	String	no	A unique ID to identity the current request/response in CMI Protocol. If not specified, it will be automatically generated.
batchID	String	no	Batch ID of the request  Requests from the private process with the same batchID are grouped together and processed in one batch.
operationID	String	yes	CMI Operation Schema version and the operation name in the following format: <i>version/Category/operationName</i> For example: 1.0/Write/UpdateParticipant.
userID	String	no	A userID to identify the BusinessConnect configuration store database as an Internal Application User. Either a UserID/password or sessionID must be supplied.
passWord	base64 Binary	no	Internal Application User's password, which should be base64-encoded outside before it is set.
sessionID	String	no	A sessionID to identify the current session. Either UserID/passwd or sessionID must be supplied.  If both UserID/passwd and sessionID are not empty, sessionID takes precedence.

Table 10 InitiatorRequest

Field	Type	Required	Description
request	String	no	<p>A string representing the message body or TIBCO Rendezvous/JMS representation of an XML file: CMI &lt;XXXREQ&gt; element</p> <p>Example:</p> <pre>&lt;UPDATEParticipantREQ&gt;   &lt;Participant name="partner1"&gt;     &lt;ListOfBizLocations&gt;       &lt;BizLocation name="loc1" addr1="xxx" /&gt;     &lt;/ListOfBizLocations &gt;   &lt;/Participant&gt; &lt;/ UPDATEParticipantREQ &gt;</pre>
closure	String	no	<p>A unique identifier used between the private process and BusinessConnect to match up the Request and Response documents in a transaction. The Response to this Request or Notify message from BusinessConnect will contain the same ID in the closure field.</p>

## Initiator Response

This message shows whether the `InitiatorRequest` has been successfully sent to your trading partner.

**Rendezvous Subject Name**     *prefix.installation.standardID*.INITIATOR.RESPONSE  
 Example: AX.BC.BC-ACME.CMI.INITIATOR.RESPONSE

**JMS Queue Name**     *prefix.installation*.INITIATOR.RESPONSE  
 Example: AX.BC.BC-ACME.INITIATOR.RESPONSE

**Message Name**     InitiatorResponse

Table 11 InitiatorResponse

Field	Type	Required	Description
standardID	String	yes	The string CMI
transactionID	String	no	A unique ID to identify the current request/response in CMI Protocol. If not specified, it will be automatically generated.
batchID	String	no	Batch ID of the executed request
operationID	String	yes	CMI Operation Schema version and the operation name in the following format: <i>version/Category/operationName</i> For example: 1.0/Write/UpdateParticipant.
sessionID	String	no	A sessionID to identify the current session.
response	String	no	An XML string representing the message body or TIBCO Rendezvous/JMS representation of an XML file. The CMI XXXRESP element contains either the response body or an ERRORMsg element: <pre>&lt;UPDATEParticipantRESP&gt;   &lt;Status code="100" message="OK" /&gt; &lt;/UPDATEParticipantRESP &gt;</pre> or <pre>&lt;UPDATEParticipantRESP&gt;   &lt;ERRORMsg code="403" message="Data Violation" desc="Exception ..."/&gt; &lt;/UPDATEParticipantRESP&gt;</pre>

Table 11 InitiatorResponse

Field	Type	Required	Description
fileReference	String	no	The URL to the response saved as a file in the TIBCO BusinessConnect shared folder
closure	String	yes	<p>A unique identifier used between the private process and BusinessConnect to match up the Request and Response documents in a transaction. The Response to this Request or Notify message from BusinessConnect will contain the same ID in the closure field.</p> <p><b>Note:</b> TIBCO BusinessConnect cXML Protocol does not enforce nor verify the uniqueness of message closures.</p>
statusCode	String	no	A code value indicating success or failure.
statusMsg	String	no	A status message indicating the status of the request sent from BusinessConnect to the private process.

## Advisory

This message can have special status such as batch execution, cancellation, and so on.

**Rendezvous Subject Name**     *prefix.installation.standardID.ADVISORY*  
 Example: AX.BC.BC-ACME.CMI.ADVISORY

**JMS Queue Name**     *prefix.installation.ADVISORY*  
 Example: AX.BC.BC-ACME.ADVISORY

**Message Name**     Advisory

Table 12 Advisory Message

Field	Type	Required	Description
standardID	String	yes	The string CMI
transactionID	String	no	A unique ID that defines a request/response in CMI Protocol.
batchID	String	no	For Error or Advisory messages related to batch execution, this field is filled with corresponding batchID.
closure	String	no	A unique identifier used between the private process and BusinessConnect to match up the Request and Response documents in a transaction. The Response to this Request or Notify message from BusinessConnect will contain the same ID in the closure field.
operationID	String	no	Operation schema version and the operation name in the following format: <i>version/Category/operationName</i> For example: 1.0/Write/UpdateParticipant
sessionID	String	no	A sessionID to identify the current session.
statusCode	String	no	A code value indicating success or failure.
statusMsg	String	no	A status message indicating the status of the request sent from TIBCO BusinessConnect to the private process.
detailMsg	String	no	Detailed information about an error is given from CMI Protocol to the private process.

## Error

---

Error message, indicates the error status or failures of the CMI Protocol request sent from the private process.

**Rendezvous**     *prefix.installation.standardID.ERROR*

**Subject Name**     Example: AX.BC.BC-ACME.CMI.ERROR

**JMS Queue Name**     *prefix.installation.ERROR*  
Example: AX.BC.BC-ACME.ERROR

**Message Name**     ERROR

Table 13 Error Message

Field	Type	Required	Description
standardID	String	yes	The string CMI
transactionID	String	no	A unique ID that defines a request/response in CMI Protocol.
batchID	String	no	For Error or Advisory messages related to batch execution, this field is filled with corresponding batchID.
closure	String	no	A unique identifier used between the private process and BusinessConnect to match up the Request and Response documents in a transaction. The Response to this Request or Notify message from BusinessConnect will contain the same ID in the closure field.
operationID	String	no	Operation Schema version and the operation name in the following format: <i>version/Category/operationName</i> For example: 1.0/Write/UpdateParticipant
sessionID	String	no	A sessionID to identify the current session.
statusCode	String	no	A code value indicating success or failure.
statusMsg	String	no	A status message indicating the status of the request sent from TIBCO BusinessConnect to the private process.
detailMsg	String	no	Detailed information about an error is given from CMI Protocol to the private process.



## Appendix A **Error Messages**

This appendix lists error messages related to CMI Protocol.

### Topics

---

- [Error Codes and Messages, page 68](#)

## Error Codes and Messages

CMI Protocol defines error codes and corresponding message that appear in the response messages and in the description column of the audit log viewer.

An error response, or `<ERRORMsg>`, includes three parts: code, message, and description as described in [Table 14](#).

Table 14 Error Codes and Messages

Code	Message	Description
100	No error occurred	The CMI protocol operation was executed successfully.
101	Session Time out	The session has timed out after a configurable time interval.
401	Authentication Failed	User does not exist or the pair <code>username/password</code> does not match.
402	Bad Request	Request is not correct, such as the request's payload does not conform to a schema, or some attributes in the request's <code>aeschema</code> are missing, or are in wrong format.
403	Data Violation	Request cannot be fulfilled because it conflicts with the existing data, such as when the update data does not exist in the configuration store.
404	Operation Not Supported	The requesting operation is not supported.
406	Authorization Error	The requesting user is not authorized to access the data.
407	ConfigStore Runtime Error	Unexpected exception happened in the configuration store layer.
500	Internal Error	Unexpected exception happened on an unknown place.

### Examples

```
<ERRORMsg code=100 message="successful">
```

```
<ERRORMsg code=500 message="internal error" desc="db is down, can not access it for the time being">
```

# Index

## A

ADVISORY 64  
audit logs 45

## B

batch transactions 37

## C

configure a connection to TIBCO BusinessConnect 12  
configure an Internal Application User 26  
create an Internal Application User 8  
customer support xi

## D

deleteBusinessAgreement 34  
deleteParticipant 34  
display imported operations 39

## E

edit the Global Variables 14  
ERROR 65  
error codes and messages 68  
explicit sessions 36

## G

getManyAuditLog 32  
getManyBusinessAgreement 33  
getManyParticipant 33  
getManyProtocolOperation 33  
getOneAuditLog 31  
getOneBusinessAgreement 31  
getOneEnabledProtocol 32  
getOneOperationBinding 32  
getOneOperationTree 32  
getOneParticipant 32

## I

implicit sessions 36  
import operations 39  
import the sample project 10  
INITIATOR.REQUEST 60  
INITIATOR.RESPONSE 62  
insert a Business Agreement 20  
insert a Host Participant 16  
insert a Participant 15  
insert a Partner Participant 18  
insertBusinessAgreement 34  
insertParticipant 34

## L

log viewer overview 44

## M

message queue [47](#)

## O

operation categories [28](#)

operation overview [28](#)

## P

performing a log search [45](#)

private message categories [56](#)

private messages in batch transactions [58](#)

private messages in sessions [57](#)

private process communications [55](#)

private process smart routing [38](#)

## R

reuse a log query [53](#)

## S

save a log query [53](#)

saving and reusing search queries [53](#)

send or cancel messages in the log queue [49](#)

session expiry settings [36](#)

session timeout [36](#)

sessions [36](#)

support, contacting [xi](#)

## T

technical support [xi](#)

TIBCO support

    TIBCOCommunity [xi](#)

TIBCO\_HOME [viii](#)

transaction overview [36](#)

## U

updateBusinessAgreement [34](#)

updateEnabledProtocol [34](#)

updateOperationBinding [34](#)

updateOperationTree [35](#)

updateParticipant [35](#)

upsertBusinessAgreement [35](#)

upsertParticipant [35](#)

using the CMI Protocol [4](#)

using the Operations Editor [39](#)

## V

viewing log search results [50](#)

## W

Write operations [33](#)