



# TIBCO BusinessConnect™ Container Edition ConfigStore Management Interface Protocol

## User Guide

Version 1.0.0 | October 2024

# Contents

---

<b>Contents</b> .....	<b>2</b>
<b>Overview</b> .....	<b>4</b>
Product Overview .....	4
Using the CMI Protocol .....	5
<b>Getting Started - Tutorial</b> .....	<b>7</b>
Import the Sample Project .....	8
Configuring Private Processes in TIBCO Business Studio .....	8
Setting Up Sample Project .....	9
<b>Working with Sample Operations</b> .....	<b>10</b>
Edit the Global Variables .....	10
Insert a Participant .....	11
Insert a Business Agreement .....	15
Get One Enabled Protocol .....	17
<b>Transactions</b> .....	<b>19</b>
Operation Overview .....	19
Operation Categories .....	19
Command Rules for the Parent/Child Element .....	21
Read Operations .....	23
Write Operations .....	26
Transaction Overview .....	28
Sessions .....	29
Batch Transactions .....	29
<b>Private Process Smart Routing</b> .....	<b>31</b>
<b>Using the Operations Editor</b> .....	<b>32</b>

<b>Viewing Logs</b> .....	<b>35</b>
Transactions .....	35
Sort Transactions .....	35
Search for Transactions .....	36
Customize Views .....	37
<b>Private Process Communications</b> .....	<b>40</b>
Private Message Categories .....	40
Private Messages in Sessions .....	41
Private Messages in Batch Transactions .....	41
Initiator Request .....	43
Initiator Response .....	45
Advisory .....	46
Error .....	48
<b>Error Messages</b> .....	<b>50</b>
Error Codes and Messages .....	50
<b>TIBCO Documentation and Support Services</b> .....	<b>52</b>
<b>Legal and Third-Party Notices</b> .....	<b>54</b>

# Overview

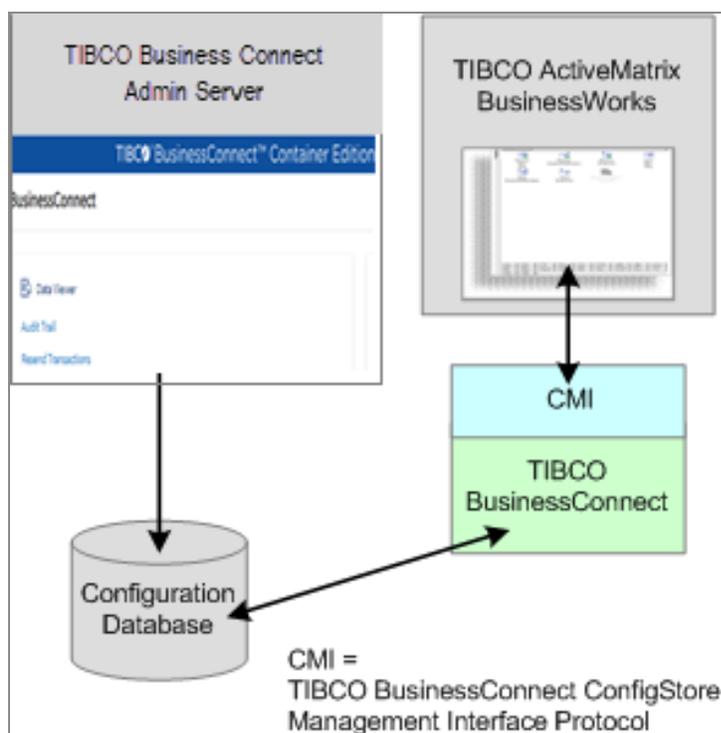
---

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

## Product Overview

BusinessConnect™ Container Edition ConfigStore Management Interface Protocol is a set of message-based interfaces that allows the private processes to interact with the TIBCO BusinessConnect Container Edition server to retrieve or update configuration data. It acts as a service to other protocols based on TIBCO BusinessConnect Container Edition 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. Here is the high-level architecture diagram for the CMI Protocol.



CMI Protocol has the following characteristics:

- It is integrated with BusinessConnect™ Container Edition for the functions Operations Editor and Audit Log Viewer.
- It is not enabled for a specific participant or bound to a specific business agreement.
- It 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 BusinessConnect Container Edition 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 BusinessConnect Container Edition Admin server.

## 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:

### Getting Started

To preview some basic operations, first install and deploy BusinessConnect Container Edition 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 [Getting Started - Tutorial](#) for details.

## Create an Internal User

To work with CMI Protocol, a specific type of user named *Internal User* has to be created and authenticated. For more information, see *TIBCO BusinessConnect™ Container Edition Trading Partner Management* guide.

## Learn how to Manage Transactions

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

## View Audit Logs

Audit logs for CMI Protocol are explained in [Viewing Logs](#).

## Learn about Private Process Communications

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

## Learn how to Interpret Error Messages

Error messages for CMI Protocol are listed in [Error Messages](#).

# Getting Started - Tutorial

---

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

## Prerequisites

To run the tutorial, you must install and configure the following:

## Install and Configure BusinessConnect Container Edition

Install and configure BusinessConnect Container Edition as explained in the manual *TIBCO BusinessConnect™ Container Edition Installation and Deployment* guide.

 **Note:** Always check the `TIB_bcce-cmi_version_readme.txt` file for the latest release requirements.

## Install and Activate CMI Protocol

Install and activate the CMI protocol using the steps explained in *TIBCO BusinessConnect™ Container Edition ConfigStore Management Interface Protocol Installation* guide.

## Deploy TIBCO BusinessConnect Container Edition

Deploy TIBCO BusinessConnect Container Edition as explained in *TIBCO BusinessConnect™ Container Edition Installation and Deployment* guide.

## Create an Internal User

In order to use the CMI Protocol, you can use the default admin user. You can also create a new user as explained in *TIBCO BusinessConnect™ Container Edition Trading Partner Management* guide.

# Import the Sample Project

The TIBCO ActiveMatrix BusinessWorks™ project files used in this tutorial can be found in the `bcce-cmi-1.0.0/samples/tutorial` directory. This example project demonstrates how to send and receive messages from TIBCO BusinessConnect™ Container Edition ConfigStore Management Interface Protocol with TIBCO ActiveMatrix BusinessWorks private processes.

## Before you begin

To use this tutorial, you must have TIBCO ActiveMatrix BusinessWorks installed before installing BusinessConnect Container Edition on the two machines.

# Configuring Private Processes in TIBCO Business Studio

To configure the tutorial in TIBCO Business Studio, perform the following steps:

## Opening the TIBCO ActiveMatrix BusinessWorks Project

To open the TIBCO ActiveMatrix BusinessWorks project in TIBCO Business Studio, perform the following steps:

### Procedure

1. Start TIBCO Business Studio.
2. Click **File > Import**.
3. On the Import page, expand the **General** folder and select **Existing Studio Projects** into Workspace and click **Next**.
4. Click **Browse** next to the **Select archive file** field to download the file from `/bcce-cmi-1.0.0/samples/BW6` directory. Select the **CMI\_BW6.zip** file and click **Open**.
5. Click **Finish**.

# Setting Up Sample Project

To set up the sample project, perform the following steps:

## Procedure

1. Click the **Project** tab and expand your project in the project panel.
2. In the Studio panel, click the **BCServerConfig** icon. This is the BusinessConnect Connection shared resource your project uses to connect to your BusinessConnect server and the configuration store.
3. On the **BCServerConfig.bcResource** tab, click the **Apply** button.
4. Click the **Configuration** tab.
5. If the **Update Transport Settings** checkbox is not selected, select it.
6. Click the **Update from Configuration Store** button.
7. Select CMI from the **Protocol Name** dropdown list.
8. Click the **Import Selected Business Protocol** button. You should now see all installed CMI operations in the BusinessConnect configuration store and click **Apply**.

# Working with Sample Operations

---

This tutorial explains how to use private processes instead of TIBCO BusinessConnect Container Edition Admin Server to create new trading partners and business agreements for TIBCO BusinessConnect Container Edition. 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](#)
- 1.0/Write/insertBusinessAgreement
  - [Insert a Business Agreement](#)
- 1.0/Read/getOneEnabledProtocol
  - [Get One Enabled Protocol](#)

For more information, see [Operation Categories](#).

## 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.

### Procedure

1. Select the **Global Variables** tab.
2. In the CMI section, make sure that the path to your sample files as well as the Internal User's name and password are correct. The default location for sample operations is in the directory `BCCE_Home\protocols\cmi\examples\SampleDocs`.

**i** **Note:** Make a backup of these files to keep the original samples 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:

## Procedure

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

The process insertparticipant opens in the design window.

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

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

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

To insert the participant, modify the sample file. The `INSERTParticipantREQ.xml` file for a participant is shown below.

```
<?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="bbc" url="http://localhost:8080/EZComm"/>
</Transports>
</EnabledProtocol>
</ListOfEnabledProtocols>
</Participant>
</INSERTParticipantREQ>

```

All the 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. INSERTParticipantREQ.xml modified for a host is shown below.

```

<?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>

```

## Procedure

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 the opened TIBCO Business Studio project, click the dropdown icon next to the **BWApplication**  icon.
7. In the **Run Configurations** dialog, expand **BusinessWorks Application** > **BWApplication**.
8. From the list of the processes displayed, you can select one or multiple items based on the type of transaction you want to process. In this tutorial, click the **Receive Notification** message.
9. Click **Apply** > **Run**.

The Select Processes to Load dialog appears.

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

11. The process is executed.

- If it is successful, you can see the Host participant added to the BusinessConnect Container Edition Admin Server under **BusinessConnect** > **Participants**.

- If the operation is not successful:

- You can debug it in TIBCO Business Studio 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 BusinessConnect Container Edition Admin Server to see the summary of this operation, including the failure causes.

To learn more about viewing audit logs, see [Viewing Logs](#).

## Insert a Partner

To insert a Partner, modify the sample file. INSERTParticipantREQ.xml file modified for a partner is shown below.

```

<?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="3" lastName="3"/>
    <PrimarySupportContact firstName="5" lastName="5"/>
  - <ListOfContacts>
    <BizContact lname="3" fname="3" type="Legal"/>
    <BizContact lname="5" fname="5" type="Support"/>
  </ListOfContacts>
  </BizLocation>
</ListOfBizLocations>
- <ListOfDomainIds>
  <DomainId domainName="AS2_ID" idName="54321"/>
</ListOfDomainIds>
- <ListOfEnabledProtocols>
  - <EnabledProtocol name="EZComm">
    <DefaultAS2DomainId idName="54321"/>
  - <Transports>
    <HttpTransport name="bbc" url="http://localhost:8080/EZComm"/>
  </Transports>
  </EnabledProtocol>
</ListOfEnabledProtocols>
</Participant>
</INSERTParticipantREQ>

```

## Procedure

1. Enter the name for the Partner (**Partner1**).
2. For the Type, enter **Partner**. Edit other data as desired
3. Edit the transport information, which is needed for the partner participant.
4. Be sure to enter a different idName from the one used for the Host.
5. Save the sample file.
6. In the opened TIBCO Business Studio project, click the dropdown icon next to the **BWApplication**  icon.
7. In the **Run Configurations** dialog, expand **BusinessWorks Application** > **BWApplication**.

The Select Processes to Load dialog appears.

8. Select the process **op\_insertparticipant** and click on **Load and Start Current**.
9. The process is executed.
  - If it is successful, you can see the Partner participant added to the BusinessConnect Container Edition Admin Server under **Partner Management**.
  - If the operation is not successful:

You can debug it in TIBCO Business Studio 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 BusinessConnect Container Edition Admin Server to see the summary of this operation, including the failure causes. To learn more about viewing audit logs, see [Viewing Logs](#).

## Insert a Business Agreement

To create a new business agreement:

1. Select the Project tab in TIBCO Business Studio.
2. Select *project\_name* > **INIT\_REQ** > **write** > **op\_insertbizagreement**.

The process for insertbizagreement opens in the design window.

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

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

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

`INSERTBusinessAgreementREQ.xml` is as follows:

```
<?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 starts and end.
9. Save the sample file.
10. In the opened TIBCO Business Studio project, click the dropdown icon next to the **BWApplication**  icon.
11. In the **Run Configurations** dialog, expand **BusinessWorks Application** > **BWApplication**.

The Select Processes to Load dialog appears.

12. Select the process **op\_insertbizagreement** and click on **Load and Start Current**.
13. The process is executed.

- If the operation is successful, you can see the Business Agreement Host-Partner1 added to the BusinessConnect Container Edition Admin Server under **Partner Management**> **Business Agreements**.

- If the operation is not successful:

You can debug it in TIBCO Business Studio 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 BusinessConnect Container Edition Admin Server to see the summary of this operation, including the failure causes.

To learn more about viewing audit logs, see [Viewing Logs](#).

## Get One Enabled Protocol

This operation allows you to read the properties of an enabled protocol that belongs to a participant. The response is given as property key value pairs.

1. Select the Project tab in TIBCO Business Studio.
2. Select *project\_name* > **INIT\_REQ** > **read** > **op\_getenabledprotocol**.  
The process for GetOneEnabledProtocol opens in the design window.

3. Select the resource **ReadFile**.

4. Select the **Input** tab.

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

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

`GETONEEnabledProtocolREQ.xml` is as follows:

```
<?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>
```

7. 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.
8. Save the sample file.
9. In the opened TIBCO Business Studio project, click the dropdown icon next to the **BWApplication**  icon.
10. In the **Run Configurations** dialog, expand **BusinessWorks Application** > **BWApplication**.

The Select Processes to Load dialog appears.

11. Select the process **op\_getenabledprotocol** and click on **Load and Start Current**.

12. The process is be executed.

- If the operation is successful, the schema with values are 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 Business Studio 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 BusinessConnect Container Edition Admin Server to see the summary of this operation, including the causes of failure. To learn more about viewing audit logs, see [Viewing Logs](#).

# Transactions

---

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

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

## 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](#)
  - Read: See [Read Operations](#)
  - Write: See [Write Operations](#)
  - Batch: See [Batch Transactions](#)
- **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 [CMI Protocol Commands](#).

**CMI Protocol Commands**

Command	Definition
INSERT	<p>Insert new data. If the data exists, an error message is returned.</p> <p>Used only for the <code>write</code> operation. See also <a href="#">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 <code>write</code> operation. See also <a href="#">Command Rules for the Parent/Child Element</a>.</p>
UPSERT	<p>Update or insert data. The CMI Protocol tries 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 <code>write</code> operation. See also <a href="#">Command Rules for the Parent/Child Element</a>.</p>
DELETE	<p>Delete data. If the data does not exist, an Error message is returned.</p> <p>Used only for the <code>write</code> operation. See also <a href="#">Command Rules for the Parent/Child Element</a>.</p>
GETONE	<p>Get one data. If no data exists, an empty response is returned. If more than one data exists, the first matched data is 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 is returned.</p> <p>Used only for the <code>Read</code> operation.</p>
LOGIN	<p>Log in and create a session.</p> <ul style="list-style-type: none"> <li>• If the authentication check is passed, a session resource is created and a response containing the new generated session ID is returned.</li> <li>• If the authentication check is not passed, a new session ID is still internally generated for audit log purpose, but an error message is</li> </ul>

Command	Definition
	returned and no session resources are created.
LOGOUT	<p>Log out and destroy the 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>Run a batch of requests.</p> <ul style="list-style-type: none"> <li>If the given batchId does not exist (already run or canceled), an error message is returned.</li> <li>If the given batchId does exist, CMI Protocol always tries to run the requests of the batch and return a success response, even if all requests fail.</li> <li>Transactions are performed based on the timestamp and in the order that they have been received.</li> </ul>
CANCEL	<p>Cancel a batch of requests.</p> <ul style="list-style-type: none"> <li>If the given batchId does not exist (already run or canceled), an error message is returned.</li> <li>If the given batchId does exist, CMI Protocol tries to delete all the 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 [Command Rule Example for the Parent/Child Element](#), where the Parent element *UPDATEParticipantREQ* is highlighted in *Italic*, and the children (or grand children) are highlighted in **Bold**.

Command Rule Example for the Parent/Child Element

```
<UPDATEParticipantREQ>
  <Participant name="partner1" type="Partner">
    <Locations>
      <Location name="loc1" addr1="newaddr" />
    </Locations>
  </Participant>
</UPDATEParticipantREQ>
```

```

    <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 [Command Rules for the Parent/Child Element](#).

### Command Rules for the Parent/Child Element

Commands	INSERT	UPSERT	UPDATE	DELETE
<b>Commands on Parent Element</b>				
INSERT	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.
UPSERT	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	Allowed. DELETE on a child element with the UPSERT command on the parent deletes the child data appropriately.

Commands	INSERT	UPSERT	UPDATE	DELETE
<b>Commands on Parent Element</b>				
			thrown.	
UPDATE	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.
DELETE	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 [Read Operation Rules](#).

### Read Operation Rules

Read Operation	Rules and Descriptions
getOneBusinessAgreement	Search is based on the name attribute value of the root

Read Operation	Rules and Descriptions
	<p>element GETONEBusinessAgreementREQ.</p> <p>The rules are:</p> <ul style="list-style-type: none"> <li>• Other attributes or elements are ignored.</li> <li>• Sorting is not supported.</li> </ul>
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>The rules are:</p> <ul style="list-style-type: none"> <li>• Other attributes or elements are ignored</li> <li>• The search is always accurate and returns one GETONEParticipantRESP response.</li> </ul> <p>Searching for a participant based on a domain ID or AS2 ID yields results only when this participant has a domain ID or AS2 ID assigned to it. This search works for protocols that support domain IDs or AS2 IDs. For these situations, only searching 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. The rules are:</p>

Read Operation	Rules and Descriptions
	<ul style="list-style-type: none"> <li>• How many rows are returned can be restricted by the size attribute.</li> <li>• You can set the criteria 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>• You can set the criteria based on the status and host attributes that are not required</li> <li>• Results are returned as name-value attribute pairs to be protocol agnostic.</li> </ul>
getManyBusinessAgreement	<p>Search is based on the name attribute value of the root element GETMANYBusinessAgreementREQ. The 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, Company* returns all business agreements containing this string, such as Company1-Company2 businessagreement, where Company1 is the Host and Company2 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. The 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 *y* would return all the participants containing *y*, such as Company1.</li> <li>• Sorting is not supported.</li> </ul> <p>Searching for a participant based on a domain ID or AS2 ID yields results only when this participant has a domain ID or AS2 ID assigned to it. This search works for protocols that support domain IDs or AS2 IDs. For these situations, only search by name is applicable.</p>

Read Operation	Rules and Descriptions
getManyProtocolOperation	<p>This operation returns all operations for a particular protocol. The 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 does 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 [Write Operation Rules](#).

### 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.

Write Operation	Rules and Descriptions
updateBusinessAgreement	<p>This operation allows the user to update an existing business agreement.</p>
updateEnabledProtocol	<p>This operation allows you to set enabled protocol properties for a participant. The input schema for this operation is constructed when you click <b>Update from Configuration Store</b> on the BusinessConnect connection shared resource. The schema considers all the 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 considers all the 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</li></ul>

Write Operation	Rules and Descriptions
updateOperationTree	<p>request schema selected for the protocol and level.</p> <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 considers all the 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	<p>This operation allows the user to update an existing participant.</p>
upsertBusinessAgreement	<p>This operation allows the user to update an existing business agreement or insert a new one.</p>
upsertParticipant	<p>This operation allows the user to update an existing participant or insert a new one.</p>

## Transaction Overview

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

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

For more information, see [Sessions](#).
- Batch transactions:** When batch transactions are used, the CMI Protocol engine stores all the data access requests. Later, the batch can be executed or canceled by sending a special request from the private process or through the message queue logs from the Log Viewer.

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

## Sessions

There are two types of sessions:

- **Explicit sessions:** During an explicit session, `login` or `logout` requests are initiated to start or to terminate a session. Each `login` request contains the client's identity information. Once this identity is accepted and the 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, performs operations, and terminates the session once the response is sent.
- **Session Timeout:** For security reasons, each user session times out, and each `sessionID` becomes invalid after a configurable time period.

If a session times out, the 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` triggers an error message.

### Edit Plug-in Properties - Session Expiry Settings

To define the time interval after which a session expires:

1. Select **BusinessConnect > System Settings > Activated Protocol Plug-ins and Properties**.
2. Click 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 interprets a transaction request to be of the type `batch` if a property `batchID` is defined in `AESchema`. All requests with the same `batchID` is executed one by one,

according to the sequence in which they have been received. Batch transactions adhere to the following 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 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](#).

# 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, 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 [Private Process Smart Routing Rules](#).

## 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

---

You can perform import operations using the operations editor.

## Import Operations



### Note

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.

## 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, is displayed as shown in the table below.

Detail schemas for each of the operations is provided in *BC\_HOME\protocols\cmi\examples\SampleDocs\cmipayload.xsd*.

The following table explains the fields of **CMI Protocol Transactions**. For more information, see [Operation Categories](#).

**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.
executeBatch	Execute a batch.
<b>Category: Read</b>	
getManyAuditLog	Get a set of audit logs.
getManyBusinessAgreement	Get a set of business agreements.
getManyParticipant	Get a set of participants.
getManyProtocolOperation	Get a set of operations for a particular protocol.
getOneAuditLog	Get one audit log.
getOneBusinessAgreement	Get one business agreement.
getOneEnabledProtocol	Get one protocol that has been enabled.
getOneOperationBinding	Get one operation binding.
getOneOperationTree	Get one operation tree.
getOneParticipant	Get one participant.
<b>Category: Session</b>	

<b>Field</b>	<b>Explanation</b>
login	Login and create a new session.
logout	Logout and destroy session.
<b>Category: Write</b>	
deleteBusinessAgreement	Delete the business agreement.
deleteParticipant	Delete the participant.
insertBusinessAgreement	Insert a new business agreement.
insertParticipant	Insert a new participant.
updateBusinessAgreement	Update the existing business agreement.
updateEnabledProtocol	Update the existing enabled protocol.
updateOperationBinding	Update the existing operation binding.
updateOperationTree	Update the existing operation tree.
updateParticipant	Update the existing participant.
upsertBusinessAgreement	Update or insert a business agreement.
upsertParticipant	Update or insert 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.

## Viewing Logs

---

This section explains how to view the transaction logs in Audit Trail of BusinessConnect Container Edition ConfigStore Management Interface Protocol.

TIBCO® AuditSafe stores information about the messages and documents processed by BusinessConnect Container Edition.

You can use the TIBCO AuditSafe to follow the processing states of inbound or outbound documents. Some of the types of information stored in the audit log include: sent and received documents; document originator; trading partner name; processing status; and validation errors.

To view the logs, on the Data Viewer tile, click **Audit Trail > CMI**.

All the transactions are displayed on the **Transactions** page.

## Transactions

On the Transactions page, you can perform the following actions:

- **Sort** transactions by columns.
- **Search** for transactions by using Filters
- View **Transaction Details**
- Change the columns you view by selecting the categories from the **Settings** dialog

The page displays 20 records with the newest transactions listed first by default. Page through the display to see older records. You can sort the transactions by category, or search for records of a specific type. The number of records returned is limited to 10,000 audit events or a maximum of 500 pages. Attempting to select additional pages after 500, will return a "No records found" error.

## Sort Transactions

Sort transactions by clicking the arrows on a column (category) heading.

To re-order the columns, click the category name to select it, and drag it to a different location on the page.

The transactions are sorted by date and time by default, so the newest transactions appear at the top of the list.

## Search for Transactions

The search feature in AuditSafe is powerful and easy to use.

You can search for values within multiple categories. The search finds matching records even if a category (or column) is not displayed. You can hide or show categories by selecting them from the **Settings** dialog box.

Use the following options to initiate a search action:

- Use the **Keyword** option to search for a term in all of the categories.
- Searching for values or keywords within the same category is an **OR** search.  
For example: Search **Status** for In Progress or Completed to find all transactions that contain those values.
- Searching for values between 2 or more categories is an **AND** search. You can combine the two search-types to narrow your results.  
For example: Search **Status** for PROCESSING**OR**COMPLETED, **AND** search **Audit Event** for Request Received.

The system automatically segments your query into OR and AND searches as seen in the following example:

### Using Search

Search contains the following options:

- Selecting **Time** allows you to define a Start and End range for your search.
- Selecting **Keyword** searches for the term in all categories.

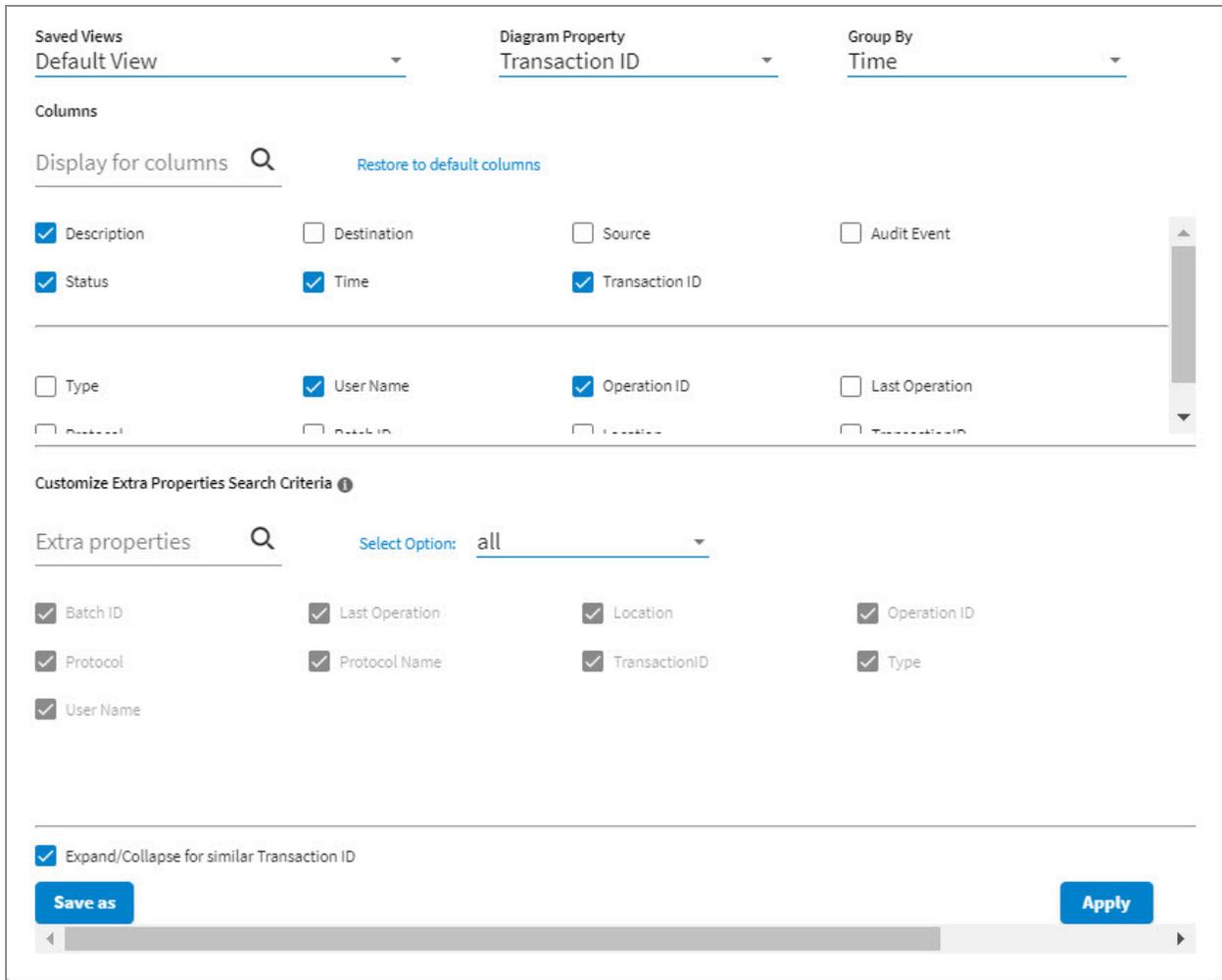
Search Option	Description
=	<b>Equals:</b> Finds an exact match for a term. Entering the first few letters will list matching options. Select one and press ENTER. If an exact match is not available, no items are listed.
!=	<b>Not Equal to:</b> Use this option to exclude terms from your search.
~	<b>Like:</b> Enter a few letters to list items containing a matching string.

1. Click the **Select** dropdown to choose a category.
2. In the **Search Transactions** field, select a search option and enter a word.
3. As you begin to enter a word, the system displays the available options.
4. Select an option and press **ENTER** to add it to your query. Continue adding options to narrow your returns.
5. Once you have created your query, click **Find** to list the results.
6. Click the **X** (Delete) icon to remove a search term.

## Customize Views

The **Settings** icon allows you to select which columns to display.

Click the gear icon to view the list and check the boxes for the columns that you want to view.



Click **Apply** to save your choices.

Customize your view using the following options:

Option	Description
Group By	Use this option to group results by category. To do this, select a category from the drop-down.
Columns	Select the check box next to the category to display the columns you want to view. You can also enter a partial word to display all the categories containing that string as shown in the following image:

Option	Description
<div data-bbox="391 289 1183 562"><p>Columns</p><p>Display for columns <input type="text"/> <a href="#">Restore to default columns</a></p><p><input checked="" type="checkbox"/> Description      <input type="checkbox"/> Destination      <input type="checkbox"/> Source</p><p><input checked="" type="checkbox"/> Status      <input checked="" type="checkbox"/> Time      <input checked="" type="checkbox"/> Transaction ID</p></div>	

# Private Process Communications

---

This section describes private message formats in CMI Protocol transactions.

## 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](#).

- **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](#).

- **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](#)). 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](#).

- **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](#).

## Private Messages in Sessions

When real-time transactions are used, all requests are 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](#).

When you use private processes in sessions, the following 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 sends 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](#).

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

1. A Private Process sends an `INITIATOR.REQUEST` message to CMI Protocol.

2. After performing a message validation, CMI Protocol stores 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.

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 starts 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 is 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 is 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 is 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.

## JMS Queue Name

*prefix.installation*.INITIATOR.REQUEST

Example: AX.BC.BC-ACME.INITIATOR.REQUEST

## Message Name

InitiatorRequest

### 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 is 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: <i>1.0/Write/UpdateParticipant</i> .
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.

Field	Type	Required	Description
passWord	base64Binary	no	Internal Application User's password, which should be base64-encoded outside before it is set.
sessionID	String	no	<p>A sessionID to identify the current session. Either UserID/passwd or sessionID must be supplied.</p> <p>If both UserID/passwd and sessionID are not empty, sessionID takes precedence.</p>
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	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 contains the same ID in the closure field.

# Initiator Response

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

## JMS Queue Name

`prefix.installation.INITIATOR.RESPONSE`

Example: `AX.BC.BC-ACME.INITIATOR.RESPONSE`

## Message Name

`InitiatorResponse`

### InitiatorResponse

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

Field	Type	Required	Description
			<pre> &lt;UPDATEParticipantRESP&gt;   &lt;Status code="100" message="OK" /&gt; &lt;/UPDATEParticipantRESP &gt; or &lt;UPDATEParticipantRESP&gt;   &lt;ERRORMsg code="403" message="Data Violation" desc="Exception ..."/&gt; &lt;/UPDATEParticipantRESP&gt; </pre>
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 contains 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.

### JMS Queue Name

*prefix.installation.ADVISORY*

Example: AX.BC.BC-ACME.ADVISORY

## Message Name

Advisory

### 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 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.

### JMS Queue Name

*prefix.installation.ERROR*

Example: AX.BC.BC-ACME.ERROR

### Message Name

ERROR

#### 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 contains 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

<b>Field</b>	<b>Type</b>	<b>Required</b>	<b>Description</b>
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 Messages

---

This appendix lists error messages related to CMI Protocol.

## 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 [Error Codes and Messages](#).

### 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 username/password 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 aeschema 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.

Code	Message	Description
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">
```

# TIBCO Documentation and Support Services

---

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

## How to Access TIBCO Documentation

Documentation for TIBCO products is available on the [Product Documentation website](#), mainly in HTML and PDF formats.

The [Product Documentation website](#) is updated frequently and is more current than any other documentation included with the product.

## Product-Specific Documentation

The documentation for this product is available on the [TIBCO BusinessConnect™ Container Edition ConfigStore Management Interface Protocol Documentation](#) page.

## How to Contact Support for TIBCO Products

You can contact the Support team in the following ways:

- To access the Support Knowledge Base and getting personalized content about products you are interested in, visit our [product Support website](#).
- To create a Support case, you must have a valid maintenance or support contract with a Cloud Software Group entity. You also need a username and password to log in to the [product Support website](#). If you do not have a username, 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 [TIBCO Community](#).

# Legal and Third-Party Notices

---

SOME CLOUD SOFTWARE GROUP, INC. (“CLOUD SG”) SOFTWARE AND CLOUD SERVICES EMBED, BUNDLE, OR OTHERWISE INCLUDE OTHER SOFTWARE, INCLUDING OTHER CLOUD SG SOFTWARE (COLLECTIVELY, “INCLUDED SOFTWARE”). USE OF INCLUDED SOFTWARE IS SOLELY TO ENABLE THE FUNCTIONALITY (OR PROVIDE LIMITED ADD-ON FUNCTIONALITY) OF THE LICENSED CLOUD SG SOFTWARE AND/OR CLOUD SERVICES. THE INCLUDED SOFTWARE IS NOT LICENSED TO BE USED OR ACCESSED BY ANY OTHER CLOUD SG SOFTWARE AND/OR CLOUD SERVICES OR FOR ANY OTHER PURPOSE.

USE OF CLOUD SG SOFTWARE AND CLOUD SERVICES IS SUBJECT TO THE TERMS AND CONDITIONS OF AN AGREEMENT FOUND IN EITHER A SEPARATELY EXECUTED AGREEMENT, OR, IF THERE IS NO SUCH SEPARATE AGREEMENT, THE CLICKWRAP END USER AGREEMENT WHICH IS DISPLAYED WHEN ACCESSING, DOWNLOADING, OR INSTALLING THE SOFTWARE OR CLOUD SERVICES (AND WHICH IS DUPLICATED IN THE LICENSE FILE) OR IF THERE IS NO SUCH LICENSE AGREEMENT OR CLICKWRAP END USER AGREEMENT, THE LICENSE(S) LOCATED IN THE “LICENSE” FILE(S) OF THE SOFTWARE. USE OF THIS DOCUMENT IS SUBJECT TO THOSE SAME TERMS AND CONDITIONS, AND YOUR USE HEREOF SHALL CONSTITUTE ACCEPTANCE OF AND AN AGREEMENT TO BE BOUND BY THE SAME.

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 Cloud Software Group, Inc.

TIBCO, the TIBCO logo, the TIBCO O logo, TIB, and Enterprise Message Service are either registered trademarks or trademarks of Cloud Software Group, Inc. in the United States and/or other countries.

All other product and company names and marks mentioned in this document are the property of their respective owners and are mentioned for identification purposes only. You acknowledge that all rights to these third party marks are the exclusive property of their respective owners. Please refer to Cloud SG’s Third Party Trademark Notices (<https://www.cloud.com/legal>) for more information.

This document includes fonts that are licensed under the SIL Open Font License, Version 1.1, which is available at: <https://scripts.sil.org/OFL>

Copyright (c) Paul D. Hunt, with Reserved Font Name Source Sans Pro and Source Code Pro.

Cloud SG 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 a specific version of Cloud SG software 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. CLOUD SG MAY MAKE IMPROVEMENTS AND/OR CHANGES IN THE PRODUCT(S), THE PROGRAM(S), AND/OR THE SERVICES DESCRIBED IN THIS DOCUMENT AT ANY TIME WITHOUT NOTICE.

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 "README" FILES.

This and other products of Cloud SG may be covered by registered patents. For details, please refer to the Virtual Patent Marking document located at <https://www.cloud.com/legal>.

Copyright © 1999-2024. Cloud Software Group, Inc. All Rights Reserved.