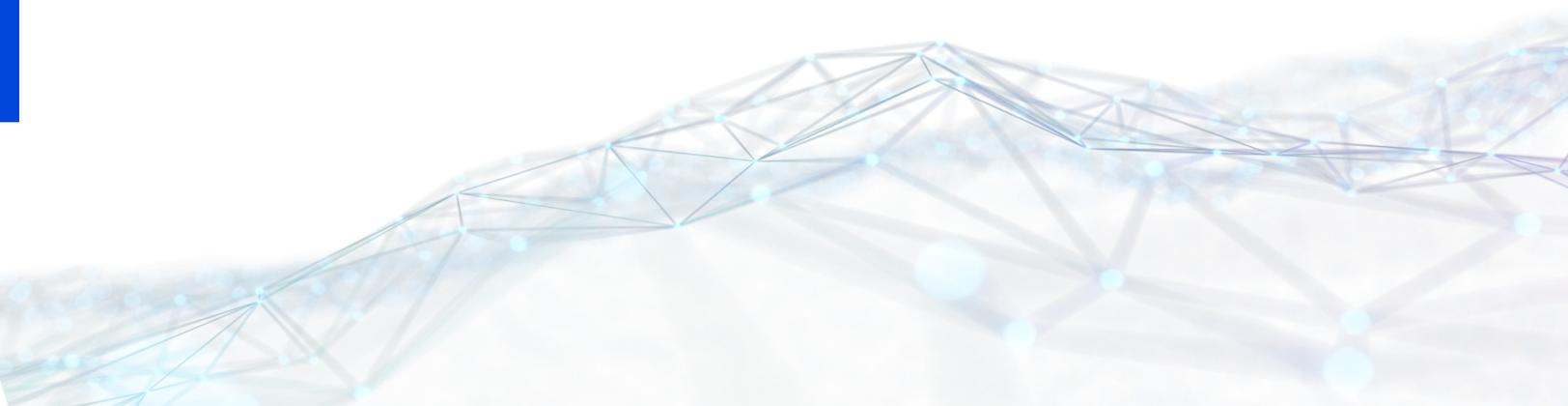




# **TIBCO BusinessConnect™ Container Edition - ebXML Protocol**

## **User Guide - ebMS3/AS4 Standard**

Version 1.0.0 | October 2023



# Contents

---

<b>Contents</b> .....	<b>2</b>
<b>Overview</b> .....	<b>6</b>
BusinessConnect Container Edition - ebXML Protocol Architecture .....	7
Features .....	8
<b>Process Flows and Functionality</b> .....	<b>10</b>
Public and Private Messages .....	10
Private Messages and Processes .....	10
Public Messages and Processes .....	11
Processing ebMS3 Transactions .....	12
One-Way Push Transactions .....	12
One-Way Pull Transactions .....	13
Two-Way Asynchronous/Synchronous Transactions .....	15
Two-Way Push and Pull Transactions .....	19
Processing Signals .....	22
Message Partition Channels .....	23
Reception Awareness .....	25
Receipt Reply Pattern .....	26
Duplicate Elimination .....	28
Security Module .....	28
Digital Signature .....	29
Encryption .....	30
UsernameToken for Authentication .....	30
PullRequest Authorization .....	31
Transport Level Security .....	31
Error Handling .....	32
Error Types .....	33

Consistency of Message Settings .....	34
<b>Preparing to Use BusinessConnect Container Edition - ebXML Protocol .....</b>	<b>35</b>
Exchanging Information with Your Trading Partners .....	35
Configuring Your BusinessConnect Container Edition Server .....	36
Transports .....	36
Copying Large Files to NFS (Sharing) .....	37
Domains .....	38
External Users .....	39
<b>Managing ebMS3 Operation Definitions .....</b>	<b>40</b>
Adding Transactions .....	40
Transaction Properties .....	43
One-Way Transaction Properties .....	43
Pull Transaction Properties .....	49
Two-Way Transaction Properties .....	52
Push and Pull Transaction Properties .....	59
Exporting Transactions .....	68
Importing Transactions .....	69
<b>Setting Up Trading Hosts and Partners .....</b>	<b>70</b>
Configuring ebMS3/AS4 Standard for Trading Hosts .....	70
General Tab .....	70
Configuring ebMS3/AS4 Standard for Trading Partners .....	72
General Tab .....	73
Transports Tab .....	75
<b>Configuring Agreement Protocol Bindings .....</b>	<b>76</b>
Configuring Business Agreements with ebMS3 Protocol ebMS3/AS4 Standard .....	76
Configuring Operation Bindings .....	76
Binding Operations .....	77
Editing Operation Bindings .....	77
Operation Settings Tab .....	78

Action Settings Tab .....	79
Transports Tab .....	79
Setting Document Security Properties .....	80
Configuring Transports .....	81
Outbound Transports for Host .....	82
Inbound Transport for Partner .....	82
Overriding Outbound Transport Settings .....	83
Overriding Participant Settings .....	83
<b>Private Processes .....</b>	<b>85</b>
Configuring Private Processes .....	85
Standalone Private Processes .....	85
TIBCO ActiveMatrix BusinessWorks Processes .....	86
Private Process Message Formats .....	86
Outbound Request Format .....	87
Request Acknowledgment Format .....	96
Inbound Response Format .....	98
Inbound Request Format .....	102
Outbound Response Format .....	106
Response Acknowledgment Format .....	112
Advisory Signal Message Format .....	114
Error Message Format .....	118
Additional Data Classes .....	121
Configuring Private Processes with TIBCO ActiveMatrix BusinessWorks .....	123
Configuring BCServerConfig Resource .....	123
Configuring Send and Receive Activities .....	124
<b>Viewing Logs .....</b>	<b>127</b>
Transactions Screen .....	127
Customizing Transactions Screen .....	127
Sort Transactions .....	129
Search for Transactions .....	129

Using Search .....	130
Transaction Details .....	131
<b>Tutorial .....</b>	<b>132</b>
Prerequisites .....	132
Setting Up the Tutorial .....	133
Setting Up Buyer on Machine 1 .....	133
Setting Up Seller on Machine 2 .....	141
Configuring Private Processes in TIBCO Business Studio .....	149
Running the Tutorial .....	150
Creating a Process Invoker .....	150
One-Way Push Transactions .....	151
One-Way Pull Transactions .....	154
Two-Way Asynchronous/Synchronous Transactions .....	157
Two-Way Push and Pull Transactions .....	160
<b>Status Codes .....</b>	<b>165</b>
<b>Property Reference .....</b>	<b>179</b>
<b>Smart Routing .....</b>	<b>181</b>
<b>Public Messages .....</b>	<b>184</b>
ebMS3 Public Message Structure .....	184
SOAP Envelope .....	186
UserMessage Element .....	187
SignalMessage Element .....	192
<b>TIBCO Documentation and Support Services .....</b>	<b>195</b>
<b>Legal and Third-Party Notices .....</b>	<b>197</b>

# Overview

---

This section provides an overview of BusinessConnect Container Edition - ebXML Protocol and the product features.

**i Note:** This document focuses on the ebMS3/AS4 standard of the protocol.

ebXML (electronic business XML) is a business protocol that buyers, sellers, and intermediaries use to securely and reliably share XML or non-XML business documents and messages over the internet. ebXML introduces a set of internet messaging services standards that you can follow to exchange business documents in a secure and reliable way.

BusinessConnect Container Edition - ebXML Protocol implements ebXML messaging services standard ebMS 3.0 and OASIS AS4 Profile.

## OASIS ebMS 3.0 Standard and AS4 Profile

The emergence of the OASIS ebMS 3.0 standard represents a leap forward in Web Services business-to-business messaging services. The OASIS ebMS 3.0 standard uses many Web Services standards and forms a single comprehensive specification for defining the secure and reliable exchange of documents by using Web Services.

However, the ebMS 3.0 specification still contains numerous options and comprehensive alternatives for addressing a variety of scenarios for exchanging data over a Web Services platform, therefore the AS4 Profile is developed by OASIS, which defines a subset of ebMS 3.0 functionality based on the “just-enough” design principles.

BusinessConnect Container Edition - ebXML Protocol implements the AS4 ebHandler Conformance Profile, which supports both Sending and Receiving roles, and for each role both message pushing and message pulling. The implementation also uses the fundamental Web Services standards such as SOAP, SOAP with Attachments and WS-Security, among others.

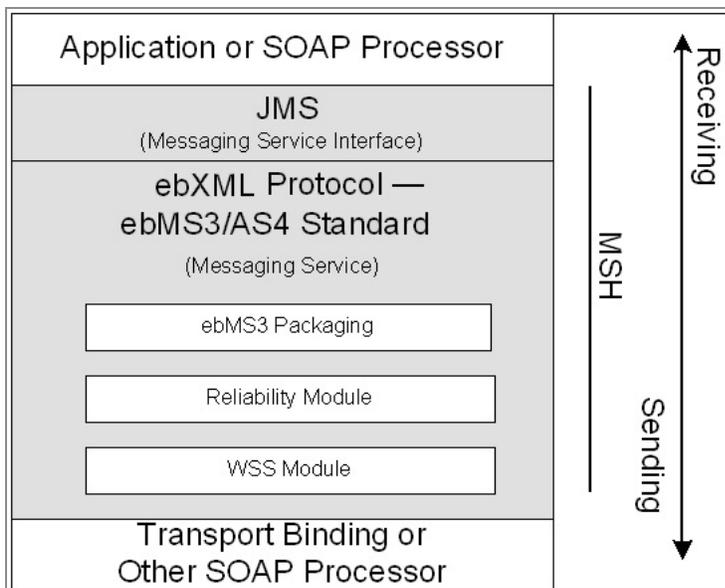
Go to <http://docs.oasis-open.org/ebxml-msg/ebms/v3.0> to access the ebMS3 specifications and AS4 profile.

# BusinessConnect Container Edition - ebXML Protocol Architecture

BusinessConnect Container Edition - ebXML Protocol — ebMS3/AS4 Standard is the TIBCO implementation of ebXML Message Service 3.0. As a plug-in to BusinessConnect Container Edition, it can handle one-way and two-way transactions.

The following figure is a graphical representation of the BusinessConnect Container Edition - ebXML Protocol implementation of ebMS3 components.

Figure 1: ebMS3 Components in BusinessConnect Container Edition - ebXML Protocol



The previous figure shows the following conceptual parts:

- [Messaging Service Interface](#)
- [Messaging Service Layer](#)
- [Transport Services](#)

## Messaging Service Interface

The applications compliant with ebMS3 can interact with the messaging service layer through the messaging service interface.

## Messaging Service Layer

The message service layer maps the abstract interface to the underlying transport services. It includes the following conceptual parts:

- **ebMS3 Packaging** Contains ebMS3 header elements and one or more payloads. See [Public Messages](#).
- **Reliability Module** Includes the following functions:
  - Receipt awareness
  - Duplicate elimination
  - Message retrySee [Reception Awareness](#) and [Duplicate Elimination](#).
- **WSS Module** Performs all security-related functions including:
  - Generation and verification of any digital signatures
  - Encryption and decryption of any SOAP body and attachments
  - Username/token authentication and authorizationSee [Security Module](#).

## Transport Services

This part of the MSH uses transport protocols, which deliver public messages.

The transport protocols supported by BusinessConnect Container Edition - ebXML Protocol are HTTP, HTTPS, and HTTPSCA.

## Features

The following are the major features in BusinessConnect Container Edition - ebXML Protocol ebMS3/AS4 Standard:

- Conforms to the ebMS 3.0 core specification and AS4 ebHandler profile.
- One-way push, one-way pull, two-way asynchronous/synchronous, and two-way push and pull transactions.
- Message Service Handler with receipt awareness and duplicate elimination.

- Receipt reply pattern specified by ebMS3/AS4: callback, response.
- Temporarily stores the outbound messages to be pulled later by the trading partner; Message Partition Channel (MPC) and selective message pulling for pull request.
- WSS Username/token as an alternative approach for authentication by following the WSS usernameToken profile.
- WSS Username/token used for authorization against a particular MPC.
- Payload signature and encryption using XML signature and encryption by following the WSS security standard.
- Digital signature and encryption of attachments by following the WSS SOAP Message with Attachments standard.
- Compressed attachments.
- Sets the ebMS3 message customer properties for payload and attachments.
- SOAP and MIME public message packaging.
- Generates standard errors defined in the ebMS3 core specification and AS4 profile.
- HTTP and HTTPS transport protocols.
- Support for integration with TIBCO ActiveMatrix BusinessWorks.
- Advisory signals to private processes for receipts or error signals received from the trading partner.
- The ability to specify certain timing constraints, such as how long to wait for a response from a private process.

Certain database tables use centralized storage of critical information. See *TIBCO BusinessConnect™ Container Edition Trading Partner Management* for more information. You can use the Audit Trail to view these records.

# Process Flows and Functionality

---

This section describes process flows and relevant functionality when ebMS3 is used by BusinessConnect Container Edition - ebXML Protocol for business transactions.

## Public and Private Messages

In an ebMS3 transaction, two partners exchange business documents over the internet based on the predefined rules of ebXML. These rules are defined by a set of processing modes (P-Mode). ebXML specifies what message formats and transport protocols that the partners have agreed to use, among other options. See *Anatomy of a Business Protocol in TIBCO BusinessConnect™ Container Edition Trading Partner Management* for more information on the partner agreement. The agreement reached between the two trading partners is represented by Agreement Reference ID or AgreementRef.

The exchange of business documents is known as the *process flow*. In any process flow implemented by BusinessConnect Container Edition - ebXML Protocol, two types of messages are exchanged:

- Private messages
- Public messages

## Private Messages and Processes

Private messages are exchanged between private processes and BusinessConnect Container Edition - ebXML Protocol. Private messages carry business documents, including request, response, or notification documents, or alternatively, advisory and error signals. For more information about BusinessConnect Container Edition - ebXML Protocol private messages, see [Private Process Message Formats](#).

Private processes are back-office systems of an enterprise that generates or processes business documents, and conducts business logic. TIBCO ActiveMatrix BusinessWorks™ Plug-in for BusinessConnect™, which works with TIBCO ActiveMatrix BusinessWorks, can act as the interface of the private processes to handle conversion of internal data to and from

the formats acceptable to the BusinessConnect Container Edition servers, which is the public process.

- **Outbound**

TIBCO ActiveMatrix BusinessWorks™ Plug-in for BusinessConnect™ working in private processes converts internal data into business documents in the formats acceptable to the public process. In this case, the private processes are the message producer in the terminology of ebMS3.

- **Inbound**

TIBCO ActiveMatrix BusinessWorks™ Plug-in for BusinessConnect™ working in private processes receives business documents in the formats defined by the public processes, converts them to internal company formats, and processes the data or forwards it to other internal resources for processing. In this case, the private processes are the message consumer in the terminology of ebMS3.

See [Private Processes](#).

## Public Messages and Processes

BusinessConnect Container Edition - ebXML Protocol works with the BusinessConnect Container Edition servers that act as public processes and exchanges documents with a trading partner over the internet by using the message formats and protocols defined by the ebXML Message Service specification, or called ebXML Messaging Service Handler (MSH), in terminology of ebXML. For more information about public processes, see [Public Messages](#).

An ebMS3 public message is a SOAP message that contains:

- **SOAP header**  
SOAP header contains ebMS message header and security header.
- **SOAP body**  
SOAP body encapsulates user payload and SOAP fault.
- **Attachments**  
Attachments encapsulate supplemental documents of arbitrary formats.

An ebMS3 message can be a user message or a signal message.

See [ebMS3 Public Message Structure](#).

## User Message

An ebMS3 user message is an ebMS message that contains a user message unit. In other words, it contains an eb:UserMessage element as a child of eb:Messaging. An ebMS user message carries business data. It is submitted by a producer and is subject to delivery to a consumer.

## Signal Message

An ebMS3 signal message is an ebMS message that contains a signal message unit. An ebMS signal message does not carry any business data. It is not intended to be delivered to a message consumer.

Three types of signal message are specified in ebMS 3.0: the error signal message, the MSH receipt signal message, and the pull request signal message.

# Processing ebMS3 Transactions

This section describes how the BusinessConnect Container Edition - ebXML Protocol ebMS3/AS4 standard processes transactions of various Message Exchange Patterns (MEPs).

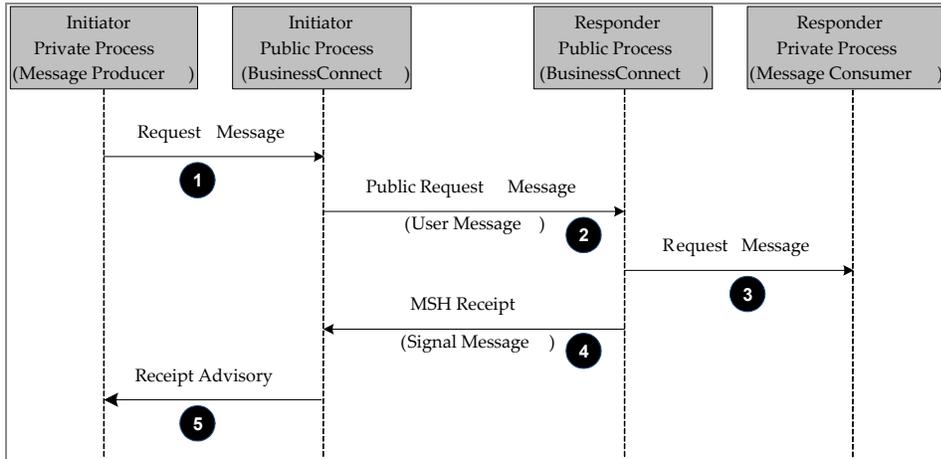
You can use BusinessConnect Container Edition - ebXML Protocol that conforming to the ebMS3 /AS4 standard to process transactions of the following MEPs:

- [One-Way Push Transactions](#)
- [One-Way Pull Transactions](#)
- [Two-Way Asynchronous/Synchronous Transactions](#)
- [Two-Way Push and Pull Transactions](#)

## One-Way Push Transactions

The following figure shows how BusinessConnect Container Edition - ebXML Protocol processes a one-way push transaction with MSH receipt required for the request message. In this figure, both the initiator MSH and responder MSH are implemented by BusinessConnect Container Edition for simplicity and convenience of description.

Figure 2: Processing One-Way Push Transactions



The following description is of the previous figure.

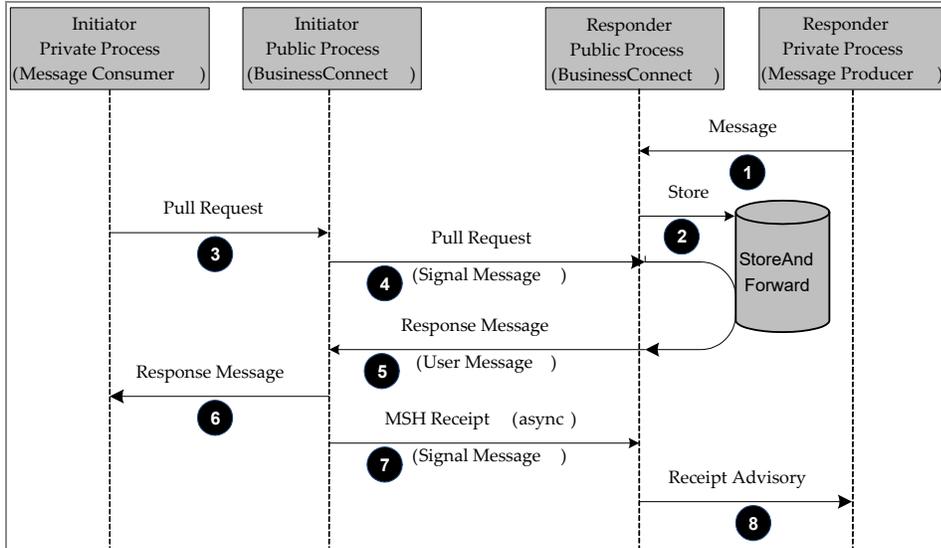
1. The initiator private process sends a request to the initiator BusinessConnect Container Edition server.
2. The initiator BusinessConnect Container Edition server encapsulates the business request into an ebMS3 user message that includes the necessary security handling, and sends the message to the responder BusinessConnect Container Edition server.
3. The responder BusinessConnect Container Edition server unpackages the received ebMS3 message, applies security handling, for example, signature verification, decryption, authentication, authorization, and sends the business documents, including the attachments, to the responder private process.
4. The responder BusinessConnect Container Edition server sends an MSH receipt back to the initiator BusinessConnect Container Edition server indicating that the message has been received and successfully processed. The MSH receipt can be sent synchronously or asynchronously.
5. The initiator BusinessConnect Container Edition server publishes an advisory message to the initiator private process indicating that the request has been accepted by the responder side.

## One-Way Pull Transactions

The following figure shows how BusinessConnect Container Edition - ebXML Protocol processes a one-way pull transaction with MSH Receipt required for the response message.

In this figure, both the initiator MSH and responder MSH are implemented by BusinessConnect Container Edition for simplicity and convenience of description.

Figure 3: Processing One-Way Pull Transactions



The following description is of the previous figure.

1. The responder private process (message producer) sends the business document in the format of a private process message to the responder BusinessConnect Container Edition server.
2. The responder BusinessConnect Container Edition server receives the business document and encapsulates it into an ebMS3 user message after applying the necessary security handling, and stores it to a temporary storage named Store-and-Forward.
3. The initiator private process (message consumer) sends a pull request in the format of a private process message to the initiator BusinessConnect Container Edition server.
4. The initiator BusinessConnect Container Edition server generates an ebMS3 pull request signal message, and then relays the pull request to the responder BusinessConnect Container Edition server.
5. The responder BusinessConnect Container Edition server retrieves a qualified user message from the Store-and-Forward storage, and then sends the message in a pull response to the initiator BusinessConnect Container Edition server.

6. The initiator BusinessConnect Container Edition server unpackages the user message, including applying the necessary security handling, for example, signature verification, decryption, and authentication, and sends the extracted business document to the initiator private process.
7. The initiator BusinessConnect Container Edition server sends an MSH receipt to the responder BusinessConnect Container Edition server asynchronously indicating that the message has been processed successfully.
8. On receiving the MSH receipt, the responder BusinessConnect Container Edition server publishes an advisory message to the responder private process indicating that the response has been accepted by the initiator side.

**i Note:** For the signed Pull Request signal message, the initiator sends a signed MSH Receipt signal message.

## Two-Way Asynchronous/Synchronous Transactions

You can use BusinessConnect Container Edition - ebXML Protocol to process a two-way asynchronous transaction or a two-way synchronous transaction.

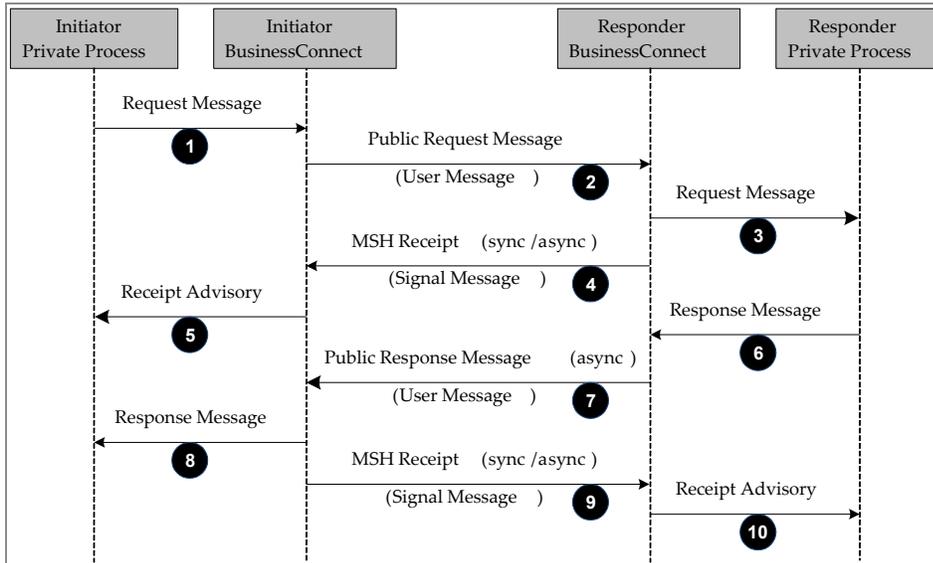
### Two-Way Asynchronous Transactions

A two-way asynchronous transaction can be considered a combination of two one-way push transactions in opposite directions. The second one, which is a response message, refers to the first one using a special message header field `eb:RefToMessageId`. In a two-way asynchronous transaction, the response message is sent asynchronously, and if the MSH receipt is required for the request message and response message, the MSH receipt can be sent synchronously or asynchronously.

The following figure shows how BusinessConnect Container Edition - ebXML Protocol processes a two-way asynchronous transaction in which MSH receipt required for both the request message and the response message can be sent synchronously or asynchronously.

In this figure, both the initiator MSH and responder MSH are implemented by BusinessConnect Container Edition for simplicity and convenience of description.

Figure 4: Processing Two-Way Asynchronous Transactions



The following description is of the previous figure.

1. The initiator private process sends the request business document in the format of a private process message to the initiator BusinessConnect Container Edition server.
2. The initiator BusinessConnect Container Edition server encapsulates the business request into an ebMS3 request user message and sends the message to the responder BusinessConnect Container Edition server. The necessary security handling can be applied to the user message.
3. The responder BusinessConnect Container Edition server unpackages the request user message with the necessary security handling being applied, and then sends the extracted request business documents to the responder private process.
4. The responder BusinessConnect Container Edition server sends an MSH receipt back to the initiator BusinessConnect Container Edition server synchronously or asynchronously. When the receipt is sent back synchronously or asynchronously, the original HTTP or HTTPS transport channel connection is closed after this receipt.
5. The initiator BusinessConnect Container Edition server sends a receipt advisory message to the initiator private process indicating that the request message has been received and processed successfully by the responder.
6. The responder private process sends the response business document in the format of a private process message to the responder BusinessConnect Container Edition

server.

7. The responder BusinessConnect Container Edition server packages the business message into another ebMS3 user message with the necessary security handling being applied, and then sends a response user message to the initiator BusinessConnect Container Edition server asynchronously because the original transport channel initiated by the initiator BusinessConnect Container Edition server was already closed.
8. The initiator BusinessConnect Container Edition server unpackages the response user message with the necessary security handling being applied, and then sends the extracted response business documents to the initiator private process.
9. The initiator BusinessConnect Container Edition server sends an MSH receipt to the responder BusinessConnect Container Edition server synchronously or asynchronously.
10. The responder BusinessConnect Container Edition server sends a receipt advisory message to the responder private process indicating the response message has been received and processed by the initiator successfully.



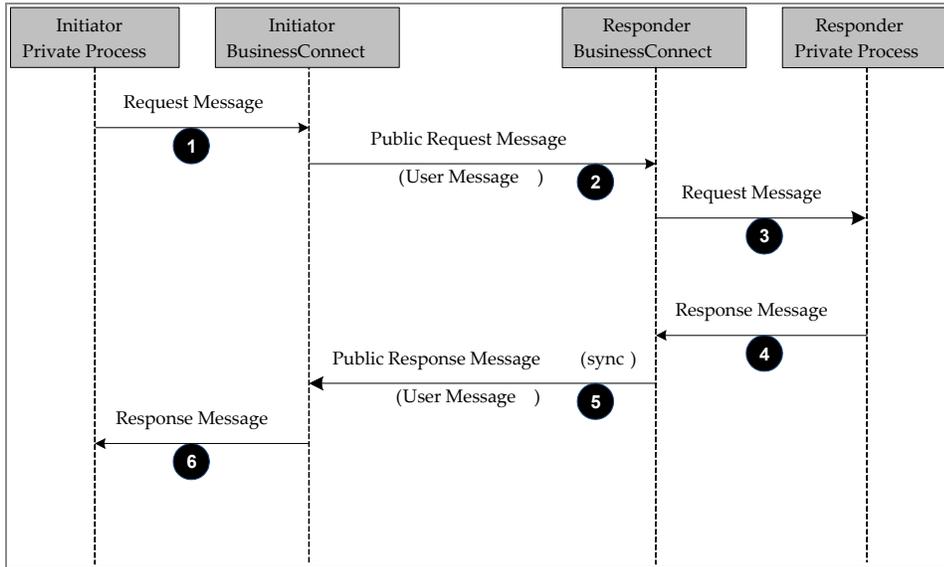
**Note:** You can set BusinessConnect Container Edition - ebXML Protocol to send an MSH receipt for a push request synchronously and send the MSH receipt for a push response asynchronously, and vice versa.

## Two-Way Synchronous Transactions

In a two-way synchronous transaction, the response message is sent synchronously through the back channel of the original transport. The reply must refer to the request by the `RefToMessageId` element. For two-way synchronous transactions, MSH receipt is not implemented for request messages and response messages.

The following figure shows how BusinessConnect Container Edition - ebXML Protocol processes a two-way synchronous transaction.

Figure 5: Processing Two-Way Synchronous Transactions



The following description is of the previous figure.

1. The initiator private process sends the request business document in the format of a private process message to the initiator BusinessConnect Container Edition server.
2. The initiator BusinessConnect Container Edition server encapsulates the business request into an ebMS3 request user message and sends the message to the responder BusinessConnect Container Edition server. The necessary security handling can be applied to the user message.
3. The responder BusinessConnect Container Edition server unpackages the request user message with the necessary security handling being applied, and then sends the extracted request business documents to the responder private process.
4. The responder private process sends the response business document in the format of private process message to the responder BusinessConnect Container Edition server.
5. The responder BusinessConnect Container Edition server packages the business message into an ebMS3 user message with the necessary security handling being applied, and then sends a response user message to the initiator BusinessConnect Container Edition server synchronously through the back channel of the original transport.
6. The initiator BusinessConnect Container Edition server unpackages the response user message with the necessary security handling being applied, and then sends the

extracted response business documents to the initiator private process.

**i** **Note:** When the synchronous pattern is selected, ensure that you select **never** from the **Require MSH Receipt** list on the **Receipt** tab of the request and response.

## Two-Way Push and Pull Transactions

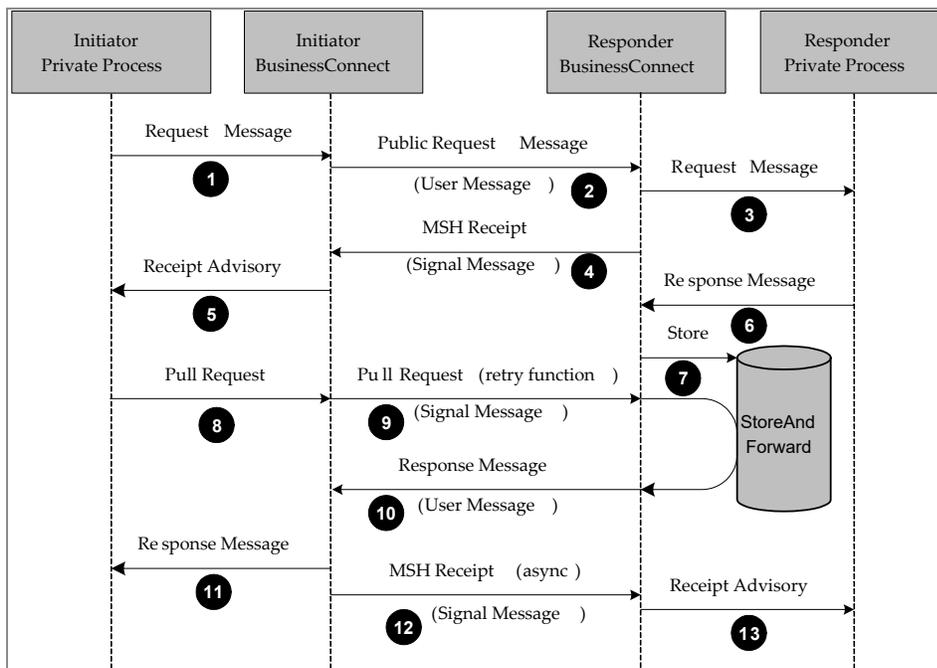
A two-way push and pull transaction can be considered a combination of a one-way push transaction and a one-way pull transaction.

An initiator sends a request message to a responder in the push transaction process, and then pulls the response message back from the responder. The pull request is automatically generated by the BusinessConnect Container Edition engine, and retries are supported with flexible time intervals. This pull request also carries the `RefToMessageId` element that refers to the original request and can be used for the responder for selective pulling. This `RefToMessageId` element is also returned back in the user message pulled from the responder.

The pull request must carry a WSS Username/token that is used to identify from which trading partner the pull request is initiated, and also to authorize the accessibility to a particular MPC. In the pull transaction process, the response message is sent synchronously through the back channel of the original transport of the pull request.

The following figure shows how BusinessConnect Container Edition - ebXML Protocol processes a two-way push and pull transaction.

Figure 6: Processing Two-Way Push and Pull Transactions



The following description is of the previous figure.

1. The initiator private process sends the request business document in the format of a private process message to the initiator BusinessConnect Container Edition server.
2. The initiator BusinessConnect Container Edition server encapsulates the business request into an ebMS3 request user message and sends the message to the responder BusinessConnect Container Edition server. The necessary security handling can be applied to the user message.
3. The responder BusinessConnect Container Edition server unpackages the request user message with the necessary security handling being applied, and then sends the extracted request business documents to the responder private process.
4. The responder BusinessConnect Container Edition server sends an MSH receipt back to the initiator BusinessConnect Container Edition server synchronously or asynchronously. After the receipt is sent back, the original HTTP or HTTPS transport channel connection is closed after this receipt.
5. The initiator BusinessConnect Container Edition server sends a receipt advisory message to the initiator private process indicating that the request message has been received and processed successfully by the responder.
6. The responder private process sends the response message in the format of private

process message to the responder BusinessConnect Container Edition server.

7. The responder BusinessConnect Container Edition server receives the response message and encapsulates it into an ebMS3 user message after applying the necessary security handling, and stores it to a temporary storage named Store-and-Forward.
8. The initiator BusinessConnect Container Edition server sends a pull request when a trigger starts after a configured time. The pull request must carry a WSS Username/token that is used to identify from which trading partner the pull request is initiated, and also to authorize the accessibility to a particular MPC. A RefToMessageId element, which refers to the original request message and is used to pull the corresponding response user message, is also indicated in the pull request message. The initiator BusinessConnect Container Edition server retries to send the request if the message is not received by the responder BusinessConnect Container Edition server or the response message is not ready to be pulled.
9. The responder BusinessConnect Container Edition server retrieves the qualified response that is referred in the pull request by the RefToMessageId element from the Store-and-Forward storage, and then sends the message in the back channel of the pull request to the initiator BusinessConnect Container Edition server.
10. The initiator BusinessConnect Container Edition server unpackages the user message, including applying the necessary security handling, for example, signature verification, decryption, and authentication, and sends the extracted business document to the initiator private process.
11. The initiator BusinessConnect Container Edition server sends an MSH receipt to the responder BusinessConnect Container Edition server asynchronously indicating that the message has been processed successfully.
12. On receiving the MSH receipt, the responder BusinessConnect Container Edition server publishes an advisory message to the responder private process indicating that the response has been accepted by the initiator side.



**Note:** For the signed Pull Request signal message, the initiator sends a signed MSH Receipt signal message.

## Processing Signals

This section describes how BusinessConnect Container Edition - ebXML Protocol processes signal messages. Three types of signal messages are specified in ebMS3:

- [Error Signal](#)
- [Receipt Signal](#)
- [Pull Signal](#)

### Error Signal

BusinessConnect Container Edition - ebXML Protocol sends error signals to trading partners if an error occurs when processing inbound messages.

See [Error Handling](#).

### Receipt Signal

You can set BusinessConnect Container Edition - ebXML Protocol to require MSH receipt for a request or response message sent to a trading partner, or generate an MSH receipt for a received request or response message. If required, the MSH receipt message is used to inform a message sender that the message has been received to achieve the reliability function called [Reception Awareness](#) defined in the ebMS3 specification. Note that in the processing flow of BusinessConnect Container Edition - ebXML Protocol, BusinessConnect Container Edition - ebXML Protocol sends out the MSH receipt message after it successfully receives and processes the user message.

### Non-repudiation of Receipt

If the original user message is digitally signed, it can be configured so that the receipt can contain a well formed `ebbpsig:NonRepudiationInformation` element. This element includes the digital digest of the original user message for the purpose of non-repudiation.

See [Receipt Reply Pattern](#) for more information about the reply pattern for ebMS3 transactions.

### Pull Signal

You can use BusinessConnect Container Edition - ebXML Protocol to send a pull request out to pull a user message from a message producer on a trading partner side, or process a

pull request from a trading partner.

The pull signal message can contain the MPC information to indicate from which message partition channel the message is pulled, and the authorization token that is used by the responder side to authorize the request for the particular MPC. An optional `RefToMessageId` element can be used, and this element is necessary if this is a selective pulling that only pulls the user message with the same `RefToMessageId` element. This is a scenario in two-way push and pull message exchange pattern.

For more information about pull signal messages, see [Message Partition Channels](#) and [One-Way Pull Transactions](#).

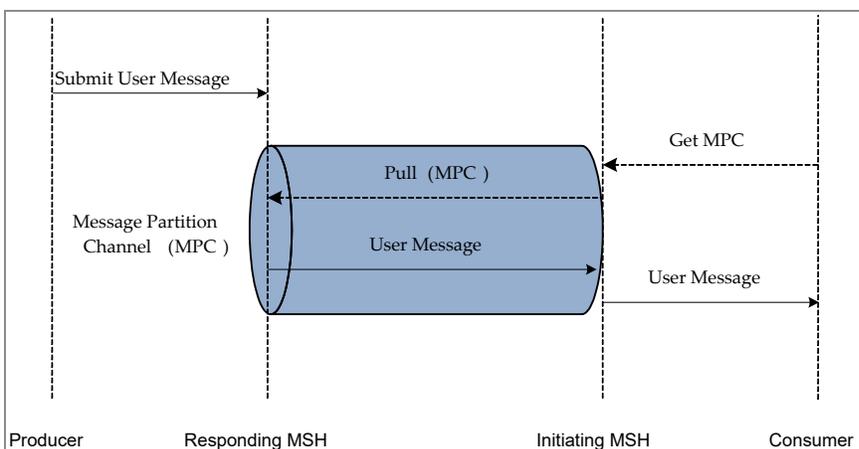
## Message Partition Channels

You can use Message Partition Channels (MPCs) for partitioning the flow of messages from a sending MSH to a receiving MSH into several flows, and then the flows can be controlled separately and consumed differently in terms of transfer priorities.

In BusinessConnect Container Edition - ebXML Protocol ebMS3/AS4 standard, MPCs are used in Pull transactions, so messages can be pulled from different queues identified by MPCs.

The following figure shows an example of the MPC workflow for one-way pull transactions.

*Figure 7: MPC for One-Way Pull Transactions*



1. The producer submits a user message to the responding MSH, intended for the

consumer on the initiator side. MPC value can be specified for this user message. If no MPC is specified, the user message is considered belonging to the default MPC. In the current implementation, a username associated with a particular trading partner has to be submitted together with the user message to be stored for pulling, so that this message can only be pulled by this user. MPC and the username are stored in Store-and-Forward together with the message.

A `RefToMessageId` element can also be submitted to indicate that only a pull request with the `RefToMessageId` element can pull this message. This is a scenario in two-way push and pull MEP, where the response user message stored in Store-and-Forward is associated with an earlier request user message. The response contains a `RefToMessageId` element that refers to the request message, and only a pull request with the same `RefToMessageId` element can pull this response message.

2. The consumer sends a pull signal message by the initiating MSH. The pull request must carry a WSS Username/token that is used to identify from which trading partner the pull request is initiated, and also to authorize the accessibility to a particular MPC.

If MPC is specified in the pull signal message, the message is supposed to be pulled from the specified MPC channel. If no MPC is specified in the pull signal message, the message is supposed to be pulled from the default MPC channel. If the optional `RefToMessageId` element is specified, the message to be pulled must have this `RefToMessageId` element refer to the original request message.

3. According to the pull request signal received, the responding MSH selects a previously submitted message from Store-and-Forward, and sends it over the response to the initiating MSH. In this process, the WSS Username/token and the MPC in the pull request message are used to authorize if the user has access to the messages in this particular MPC. If a `RefToMessageId` element is presented in the pull request, this element is also used to identify the message only with the same `RefToMessageId` element.

**Note:** If more than one message in the MPC conforms to the conditions specified by the pull request, a First-in-First-out (FIFO) policy is applied to select the message. If no user message is available in the specified MPC to be pulled, an error signal message with warning error code is sent back instead.

4. The initiating MSH receives the returned ebMS3 message, unpackages the message,

including the necessary security handling, and sends the extracted business documents to the initiator private process, the message consumer.

5. The initiating MSH can choose to send a receipt asynchronously to the responder MSH, indicating that the message has been received and processed. The receipt can also be a non-repudiation receipt, which contains the signed digest of the user message just being pulled.

## Reception Awareness

With BusinessConnect Container Edition - ebXML Protocol, you can customize the level of reliability in message delivery by requiring *MSH receipt* messages and trying to send the messages again if the receipt messages are not received in a given time. MSH receipts, when required, are used to confirm the successful delivery of messages.

To configure MSH receipts, go to **BusinessConnect Container Edition > B2B Administration > Operations Editor > ebMS3 > transaction > Action tab > Receipt**, and select one option from the **Require MSH Receipt** list.

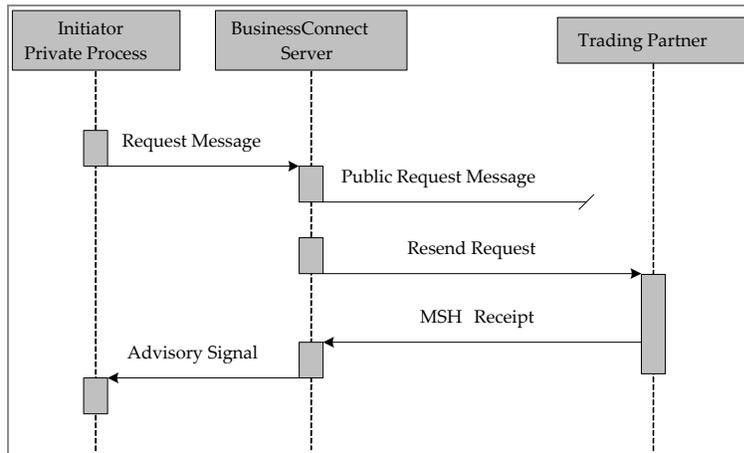
When MSH receipts are required, you can set the following parameters:

- **The time to wait for an MSH receipt**  
The length of time to wait for an MSH receipt from a trading partner. After this time expires, the action message is resent to the trading partner as many times as the maximum retries times. If the last retry fails, an error message is sent back to the private process (the message producer) indicating the failure of the delivery.
- **Maximum number of retries**  
The maximum number of times to retry sending the action message before receiving an MSH receipt.

 **Note:** This property is not available for one-way pull transaction.

The following figure shows how BusinessConnect Container Edition - ebXML Protocol processes a one-way push transaction with MSH receipt required for the request message.

Figure 8: Message Delivery with MSH Receipt



If the initiator does not receive the receipt for a public message before the waiting time expires, it resends the message until the maximum number of retries has been reached.

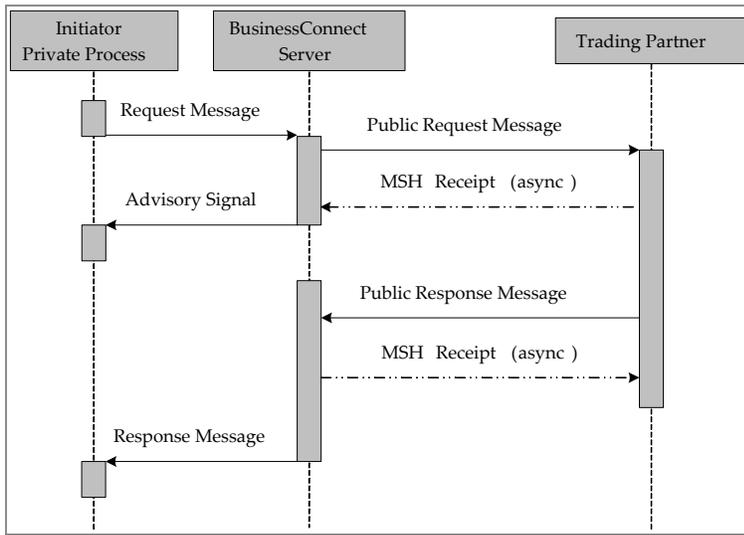
## Receipt Reply Pattern

You can use BusinessConnect Container Edition - ebXML Protocol to define two receipt reply patterns for ebMS3 transactions: callback and response. You can select one of them from the **Receipt Reply Pattern** list on **BusinessConnect Container Edition > B2B Administration > Operations Editor > ebMS3 > transaction > Action** tab.

- **callback**

Sends MSH receipts through a separate transport channel. This mode is called asynchronous mode. The following figure depicts the behavior of this mode when BusinessConnect Container Edition - ebXML Protocol works as an initiator in a two-way asynchronous transaction.

Figure 9: Receipt Reply Pattern: callback

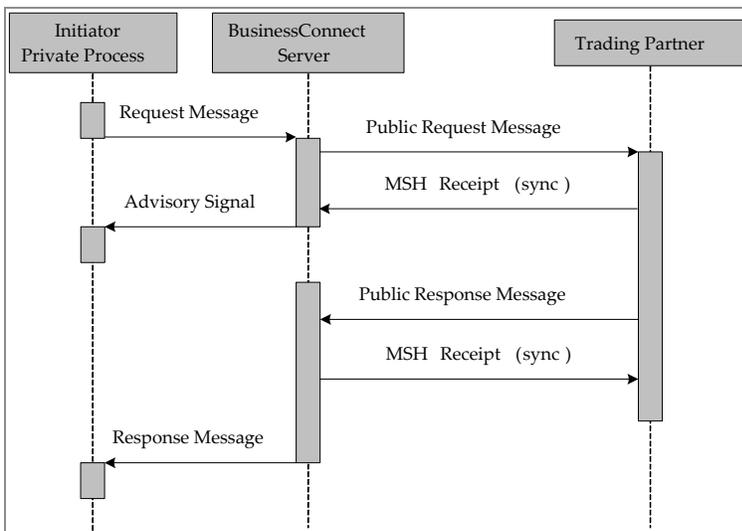


The dotted line in the previous figure indicates that the receipts are replied through a separate transport channel.

- **response**

Sends the MSH receipt by the back-channel of the original transport. This mode is called synchronous mode. The following figure depicts the behavior of BusinessConnect Container Edition - ebXML Protocol when it works as the initiator in a two-way asynchronous transaction.

Figure 10: Receipt Reply Pattern: response



The solid line in the previous figure indicates that the receipts are replied through the back channel of the original message transport.

## Duplicate Elimination

With BusinessConnect Container Edition - ebXML Protocol, you can customize the level of reliability in message delivery by detecting and eliminating duplicate messages.

To set BusinessConnect Container Edition - ebXML Protocol to eliminate duplicate messages or not, go to **BusinessConnect Container Edition > B2B Administration > Operations Editor > ebMS3 > transaction > Action > General** and select one value from the **Eliminate Duplicate Message** list.

Duplicate detection and elimination are always supported by the BusinessConnect Container Edition - ebXML Protocol ebMS3/AS4 standard both for inbound messages and outbound messages. A duplicate message can occur if the sending side retries to send the same message because of the loss of the receipt used to acknowledge an original user message.

The receiving MSH keeps the record of all recently received messages, so for a newly received user message, the `eb:MessageInfo/eb:MessageId` element in the message is checked against the record. If this message ID exists in the record, duplication is detected and audit logged. This duplication status is reported to the private process of the receiver side (the message consumer). If duplicate elimination is configured, the duplicate user message is not sent to the private process, but rather an error advisory is sent to the private process.

The sending MSH also keeps the record of all business messages received from its private process. If the business message received from the private process contains a message ID and this ID matches one in the record, duplication is detected. If duplication elimination is configured, this message is rejected.

## Security Module

BusinessConnect Container Edition - ebXML Protocol ebMS3/AS4 Standard supports the security module defined in the ebMS 3.0 core specification and the AS4 ebHandler Profile that follows the WSS specifications.

BusinessConnect Container Edition - ebXML Protocol ebMS3/AS4 Standard supports the following security functions:

- [Digital Signature](#)
- [Encryption](#)
- [UsernameToken for Authentication](#)
- [PullRequest Authorization](#)
- [Transport Level Security](#)

## Digital Signature

Digital signatures are used to bind information to the identity of its originator. They can be used to provide data origin authentication, data integrity, and non-repudiation.

BusinessConnect Container Edition - ebXML Protocol ebMS3/AS4 Standard supports the following digital signature functions:

- Web Services Security X.509 Certificate Token Profile.
- Digital signature for SOAP header and SOAP body:
  - Detached signatures format as defined by the XML Signature Specification.
  - The entire `eb:Messaging` container element and the SOAP body must be included in the WSS signature when signing messages.
- Digital signature for SOAP attachments:
  - The Attachment-Content-Only transform is used for the attachment signature, which follows the AS4 specification.
  - All MIME body parts of the to-be-signed attachments are included in the WSS signature, together with the entire `eb:Messaging` container element and the SOAP body when signing attachment.
- Digital signature for receipt (non-repudiation of receipt) to acknowledge signed messages.

You can set BusinessConnect Container Edition - ebXML Protocol to use the signature of request and response, go to **BusinessConnect Container Edition > B2B Administration > Operations Editor > ebMS3 > *transaction* > Action > General > Require Non-repudiation of Request/Response.**

You can set BusinessConnect Container Edition - ebXML Protocol to use the signature of receipt. Go to **BusinessConnect Container Edition > B2B Administration > Operations Editor > ebMS3 > transaction > Action > Receipt > Require Non-Repudiation of MSH Receipt**.

## Encryption

BusinessConnect Container Edition - ebXML Protocol ebMS3/AS4 Standard supports the following encryption functions:

- Web Services Security X.509 Certificate Token Profile
- Encryption for SOAP body
  - Supports the WSS Encryption [WSS11] specification over SOAP body when it is used to carry payloads.
- Encryption for SOAP attachments
  - Supports the SOAP Messages with Attachments specification [SOAPATTACH] to encrypt the MIME body parts of the included payloads.
  - Processes Content-Transfer-Encoding in MIME header:  
When sending encrypted attachments, the value set for the Content-Transfer-Encoding header field is "binary". When receiving encrypted attachments, the acceptable value for the Content-Transfer-Encoding header field can be "binary" or "base64".

You can set BusinessConnect Container Edition - ebXML Protocol to use the document encryption. Go to **BusinessConnect Container Edition > B2B Administration > Operations Editor > ebMS3 > transaction > Document > Encrypt Document**.

## UsernameToken for Authentication

As an authentication alternative, WSS Username/Password Token ([WSS11-UT]) can also be used, without the exclusiveness of Digital Signature Authentication, for inbound message authentication.

The password type of Username/token is the only digest password in the current implementation for outbound messages. For inbound message, either the digest or plain text password that is normally used together with SSL transport is accepted.

When Username/token is used for authentication, the `wsse:UsernameToken` element is in the “primary” security header along with other security entities such as digital signature and encryption element if they are present. This “primary” security header is supposed to be handled by the security module of the receiving MSH, so it does not have the `@role="ebms"` attribute.

You can set BusinessConnect Container Edition - ebXML Protocol to use the Username/token for authentication. Go to **BusinessConnect Container Edition > B2B Administration > Operations Editor > ebMS3 > transaction > Action > General > Require UsernameToken Authentication of Request/Response.**

## PullRequest Authorization

WSS Username/Password Token ([WSS11-UT]) can also be used with Pull transaction, to authorize a pull request for the accessibility to a specific MPC. If the MPC is missing, the request is supposed to be pulled from the default channel.

The username and password in the PullRequest are used for authorizing whether an external user, who is associated with a trading partner, has the access to pull messages from a particular MPC. If the authorization succeeds, the message is pulled from this particular channel belonging to this partner in message storage.

When the Username/token is used for MPC authorization, it is put in a "secondary" security header, and is supposed to be handled by the ebMS module on the receiving MSH, so this security header has the `@role="ebms"` attribute in the message.

See [Message Partition Channels](#).

## Transport Level Security

The transport level security such as SSL over HTTP/HTTPS is also supported by the protocol.

For more information about the transport level security functions, see the TIBCO BusinessConnect™ Container Edition documentation.

# Error Handling

BusinessConnect Container Edition - ebXML Protocol sends error signals to the trading partner whenever it encounters an error while processing the inbound ebMS3 action or signal messages. See [Processing Signals](#). Reasons for an error message include, but are not limited to the following situations:

- A delivery failure occurred.
- The ebMS3 message envelope is not well formed, cannot be validated on XML, SOAP, or ebXML layer, and fails to resolve external references.
- The transaction is not authorized.
- The ebMS3 message data does not match up with the settings in the BusinessConnect Container Edition - ebXML Protocol configuration store, such as settings for trading partners, digital signatures, encryption, non-repudiation of receipt, or the mismatch of agreement reference, P-mode ID, and so on.
- The message did not pass security checks, such as digital signature verification failure, failed to be decrypted, and so on.

When BusinessConnect Container Edition - ebXML Protocol receives an error message from a trading partner in response to an outbound message, it updates the private process that is the message producer by publishing the following private messages:

- **In case of trading partner error while processing request message:** an ae/ebMS3/InitiatorResponse message that packages the error message in the **response** field. See [Inbound Response Format](#).
- **In case of trading partner error while processing response message:** if an ae/ebMS3/ResponderAck message has not already been sent to the private process, an ae/ebMS3/ResponderAck message with an error code in the **statusCode** field. See [Response Acknowledgment Format](#).
- An ae/ebMS3/Advisory message on the error subject. See [Error Message Format](#).

 **Note:** All the error messages are sent synchronously.

## Error Types

An error generated by BusinessConnect Container Edition - ebXML Protocol can be one of the following types:

- [SOAP Faults](#)
- [AS4 Errors](#)
- [ebMS Errors](#)
- [Internal Errors](#)

### SOAP Faults

The SOAP faults generated by BusinessConnect Container Edition - ebXML Protocol are errors against the SOAP specification.

Go to [http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/core/os/ebms\\_core-3.0-spec-os.html](http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/core/os/ebms_core-3.0-spec-os.html).

### AS4 Errors

The AS4 Errors generated by BusinessConnect Container Edition - ebXML Protocol are errors against the AS4 specification.

Go to <http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/profiles/AS4-profile/v1.0/cs03/AS4-profile-v1.0-cs03.html>.

### ebMS Errors

The ebMS errors generated by BusinessConnect Container Edition - ebXML Protocol are errors occurred against the ebMS3 specification.

Go to [http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/core/os/ebms\\_core-3.0-spec-os.html](http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/core/os/ebms_core-3.0-spec-os.html).

### Internal Errors

The internal errors generated by BusinessConnect Container Edition - ebXML Protocol are errors occurred during the one-way push, one-way pull, two-way asynchronous/synchronous, and two-way push and pull transactions that are not defined by either SOAP, ebMS, or AS4 specifications.

See [Status Codes](#) for a list of all internal errors.

## Consistency of Message Settings

In ebXML, it is imperative that you and your trading partner use messages and message elements that are consistent with MSH configuration of each other. This is best achieved if you and your trading partner reach a detailed agreement for each of the business transactions you perform together. This business agreement must include the following details for every transaction: transaction name, role and service information, timeout settings, duplicate elimination, digital signatures, encryption, MSH acknowledgments, receipt acknowledgments, and synchronous reply mode.

When BusinessConnect Container Edition - ebXML Protocol receives an inbound message that is inconsistent with any of the settings, the transaction is interrupted and an MSH error is sent to the trading partner.

# Preparing to Use BusinessConnect Container Edition - ebXML Protocol

---

This section describes the preliminary information that you use to configure your BusinessConnect Container Edition server for your ebMS3 transactions. It also provides instructions for configuring some of this information in your BusinessConnect Container Edition server. Perform all the tasks in this section before configuring BusinessConnect Container Edition participants with the ebXML protocol.

## Exchanging Information with Your Trading Partners

Before you start conducting electronic business with your trading partner, you must exchange the following details:

- **Party ID and PartyID Type**

The Party IDs are how you and your trading partner identify each other. The identity belongs to a business domain, which is called PartyID Type in the ebMS3 standard. The PartyID and PartyID Type of both sides are present in the ebMS3 user messages exchanged between the two parties. The trading partner can choose not to use PartyID Type, in which case the PartyID must be the format of a valid URI.

- **Server URLs**

You and your trading partner must know the server URL of each other to conduct electronic business transactions.

Server URLs for BusinessConnect Container Edition - ebXML Protocol ebMS3/AS4 Standard follow the formats: *protocol://hostName:portNumber/dmz/ebMS3*

The host name and port number are determined by your IT department. See [Transports](#) for how to set the inbound transport for receiving messages from your trading partner.

Example:

`http://www.myhost.com:6700/dmz/ebMS3`

<https://www.myhost.com:6705/dmz/ebMS3>

- **Public Certificates**

To use HTTPS for your ebMS3 transactions, and use digital signatures for document authentication, or encryption for data security you must also exchange public certificates. BusinessConnect Container Edition - ebXML Protocol ebMS3/AS4 Standard supports PKI X.509 Certificate Token for public certificate.

- **P-Mode and Agreement Reference**

You and your trading partner must agree on the processing mode (P-Mode ID) and the agreement reference (AgreementRef) that are used to guide the transactions.

- **MPC**

You and your trading partner must agree on the MPC if they support message prioritization by using Message Partition Channel, and the Username/tokens used to authorize the accessibility to each channel.

## Configuring Your BusinessConnect Container Edition Server

You must perform a few tasks in BusinessConnect Container Edition before you can successfully configure a BusinessConnect Container Edition participant with BusinessConnect Container Edition - ebXML Protocol.

### Transports

Before you can configure a BusinessConnect Container Edition transport protocol of participants, you must configure the transport protocols. BusinessConnect Container Edition - ebXML Protocol supports the following two types of transports: HTTP and HTTPS.

For more information about configuring the outbound transport to send ebMS3/AS4 messages to your trading partner, see HTTP, HTTPS, and HTTPSCA Transports in *TIBCO BusinessConnect™ Container Edition Trading Partner Management*.

For more information about configuring the transport services with your trading partner, see Gateway Services in *TIBCO BusinessConnect™ Container Edition Administration*.

## Copying Large Files to NFS (Sharing)

To copy files (greater than 5 MB as configured by default) to the NFS (sharing) folders, perform the following steps:

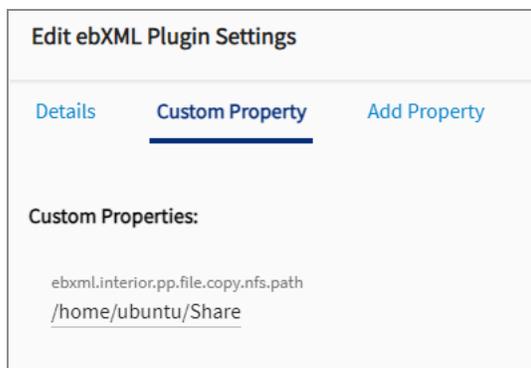
### Procedure

1. Go to **System Settings > Others > Activate Protocol Plugins**.
2. Click **ebXML**.
3. In the Edit ebXML Plugin Settings dialog, click **Add Property**.
4. Set the following fields.

Field	Value
Property Name	ebxml.interior.pp.file.copy.nfs.path
Property Type	String

5. Click **Add**.
6. On the **Custom Property** tab, configure the `ebxml.interior.pp.file.copy.nfs.path` field to NFS path.

Example: `/home/ubuntu/Share`



**Note:** You must set the `ebxml.fileref.threshold` value to 5000000 (5 MB).

7. Click **Save**.

8. Similarly, create a custom property for BCCE with the following values.

Field	Value
Property Name	<code>audit.tas.logging.payload.limit</code>
Property Type	Integer

This limits the view or downloadable payload size on AuditSafe. The maximum limit is 5 MB.

**Note:** You must set the `audit.tas.logging.payload.limit` value to 5000000.

**Custom Properties:**

`audit.tas.logging.payload.limit`

5000000

---

`audit.tas.logging.payload.limit`

## Domains

If the Party IDs you and your trading partners used to identify each other belong to a business domain, which is also called PartyID Type in ebMS3 standard, you have to define this domain in the BusinessConnect Container Edition server. This domain, when chosen to be used for a trading partner, presents as the `eb:PartyID/@type` attribute in the ebMS3 message sent to the partner.

To add a domain, perform the following steps:

### Procedure

1. Go to **System Settings > Others > Metadata Type Configuration**.
2. On the **Domains** tab, click the **Add** icon .

3. On the **Create Domain Type** dialog, in the **Name** field enter the name of the domain.
4. Click **Add**.

## External Users

If your trading partners use WSS:UsernameToken to authenticate the message they send to you, or use WSS:UsernameToken to authorize a pull request issued by them, you must define the external users associated with this trading partner.

See the "External Users" section in *TIBCO BusinessConnect™ Container Edition Trading Partner Management* to create an external user and associate this user with a particular trading partner.

# Managing ebMS3 Operation Definitions

---

This section explains about managing ebMS3 operation definitions.

The Operations Editor panel organizes ebMS3 operation definitions, which are called transactions in containers of various levels. Each of the operation definitions consists of many property definitions, which define the process mode of a particular business transaction. A process mode is called P-Mode in ebMS3 terminology and is referenced by P-Mode ID. In Operations Editor, transactions defined by the same organization are grouped into the same *organization* container, for example, RosettaNet.org. Within an organization container, they are organized into *business process* containers, for example, 3A4. Then, transactions are further organized by their document version into *version* containers, for example, 1.1. A *transaction* is the most basic unit for an ebMS3 transaction in the Operations Editor panel, for example, Create Purchase Order.

**i Note:** When you give a container any name you want, follow the naming convention to ensure that both you, and your trading partner are using the same naming conventions for your ebMS3 transactions.

Before you can use ebMS3 in business transactions, you must add the transactions in the Operations Editor panel and configure each of them accordingly. See [Adding Transactions](#).

After you have added the transactions into the Operations Editor panel, you can export them to a file. When you upgrade or reinstall BusinessConnect Container Edition - ebXML Protocol, you can simply import the data into the Operations Editor panel. See [Exporting Transactions](#) and [Importing Transactions](#).

## Adding Transactions

The section provides the instructions on adding transactions in the Operations Editor panel, which consists of the following tasks:

- [Adding Organization Containers](#)
- [Adding Business Process Containers](#)

- [Adding Version Containers](#)
- [Adding Transactions](#)

## Adding Organization Containers

To add an organization container, perform the following steps:

### Procedure

1. On the **B2B Administration** tile, click **Operations Editor**.
2. Click **ebMS3** from the **Protocol** list.
3. Click the **Add** icon  .
4. In the **New Organization** dialog:
  - a. In the **Name** field, enter the organization name.
  - b. Optional: In the **Description** field, enter a description.
  - c. Optional: In the **Business Process Agency ID** field, enter the business process agency ID.
5. Click **Add**.

## Adding Business Process Containers

To add a business process container, perform the following steps:

### Procedure

1. Hover over the organization container that you created in [Adding Organization Containers](#) and click the **three-dot** menu.
2. Click **New Business Process**.
3. In the **New Business Process** dialog:
  - a. In the **Name** field, enter the name of the business process as defined by the organization you specified in [Adding Organization Containers](#).
  - b. Optional: In the **Description** field, enter a description.
4. Click **Add**.

## Adding Version Containers

To add a version container, perform the following steps:

### Procedure

1. Hover over the version container that you created in [Adding Business Process Containers](#) and click the **three-dot** menu.
2. Click **New Version**.
3. In the **New Version** dialog:
  - a. In the **Name** field, enter the version of the business transaction.
  - b. Optional: In the **Description** field, enter a description.
  - c. Optional: To add a schema file for this version container, click **Upload file** and browse to locate the schema.
4. Click **Add**.

## Adding Transactions

To add a transaction, perform the following steps:

### Procedure

1. Hover over the version container that you created in [Adding Version Containers](#) click the **three-dot** menu.
2. Select either **New One-Way Transaction**, **New Two-Way Transaction**, **New Push and Pull Transaction**, or **New Pull Transaction**.
3. Configure the transaction on each tab.  
For more information about how to configure transactions, see [Transaction Properties](#).
4. Click **Add**.

# Transaction Properties

This section describes each field in one-way transactions, two-way transactions, pull transactions, and push and pull transactions.

- [One-Way Transaction Properties](#)
- [Pull Transaction Properties](#)
- [Two-Way Transaction Properties](#)
- [Push and Pull Transaction Properties](#)

## One-Way Transaction Properties

If you select **New One-Way Transaction** in [Adding Transactions](#), you see the **One-Way Transaction** and **Request Action** tabs when you configure the transaction in the **New One-Way Transaction** dialog. A one-way operation type can be used for guiding a one-way push transaction, or can be used for guiding how a message is prepared and stored in Store-and-Forward, so it can be pulled by another one-way pull transaction.

 **Tip:** You can override certain transaction settings when configuring protocol bindings for a business agreement. See [Operation Settings Tab](#).

The following table describes each field on the **One-Way Transaction** tab.

### One-Way Operation Type: One-Way Transaction Tab

Field	Description
<b>General Tab</b>	
<b>Name</b>	The displayed name of the transaction.
<b>Description</b>	Optional. A short description of the transaction.
<b>Roles Tab</b>	
<b>Initiating Role</b>	The role of the initiator in this transaction. This field is mapped to the

Field	Description
	respective Role element in the eb:PartyId/eb:From element of the public user message.
<b>Responding Role</b>	The role of the responder in this transaction. This field is mapped to the respective Role element in the eb:PartyId/eb:To element of the public user message.
<b>Service Information Tab</b>	
<b>Service</b>	The service that acts on this message, as defined by ebMS. This field is mapped to the Service element in the public message SOAP header.  Specify a URI in this field if the <b>Service Type</b> field is empty.
<b>Service Type</b>	The service type, as defined by ebMS. This field is mapped to the Type attribute in the public message Service element.  Specify a URI in the <b>Service</b> field if this field is empty.
<b>AgreementRef</b>	The reference of the agreement between the host and the trading partner. Such an agreement is the set of P-Mode operations that are essential for the two parties for their MSHs to interoperate. This reference is present in the eb:CollaborationInfo/eb:AgreementRef element. In inbound messages sent from this trading partner, the eb:AgreementRef element must have the same value to be accepted and processed.  This field takes precedence over the setting on trading partners level, and you can override this value on the business agreements level.
<b>Type</b>	The type of the AgreementRef value. If it is empty, the AgreementRef value must be a valid URI.  This field takes precedence over the setting on trading partners level, and you can override this value on the business agreements level.
<b>P-Mode ID</b>	The identifier for the processing mode defined by this transaction. It is also called operation definition in BusinessConnect Container Edition. This is the default P-Mode ID for this P-Mode. You can override the P-

Field	Description
	<p>Mode ID by using a special P-Mode ID agreed between you and your trading partner when configuring protocol bindings for a business agreement to identify the specific P-Mode used between you and your trading partner for guiding this particular transaction.</p> <p><b>Note:</b> The combination of service, service type, the request action name, and P-Mode ID are the identification of a business transaction. These values are present in user messages, which are used by the receiver to recognize which P-Mode or operation type is used to process a received user message.</p>

## Document Tab

<b>Encrypt Document</b>	Select this checkbox to encrypt the outbound document. When this is selected, it is also required that the inbound message of this operation is encrypted.
<b>Compress Document</b>	Selecting this checkbox requires that the outbound documents of this operation are compressed. When this is selected, the contents of each attachment in the message are compressed.
	<p><b>Note:</b> This option is used only to compress attachments, not to compress messages. When a message does not include attachments, the message is not compressed even if you select this checkbox.</p>

The following table describes each field on the **Request Action** tab.

### One-Way Operation Type: Request Action Tab

Field	Description
<b>General Tab</b>	
<b>Name</b>	The name of the action.
<b>Description</b>	Optional. A description of the action.

Field	Description
<b>Direction</b>	The direction of this action.
<b>Require Non-repudiation of Request</b>	<p>Require the action message to be digitally signed for non-repudiation. If this checkbox is selected, BusinessConnect Container Edition - ebXML Protocol works as follows:</p> <ul style="list-style-type: none"> <li>• Signs the outbound action message and stores the message in the non-repudiation log.</li> <li>• Requires that the inbound action message of this operation is also digitally signed; if it is not signed, the message is considered invalid and an error message is sent back.</li> </ul>
<b>Require UsernameToken Authentication of Request</b>	<p>Require the action message to contain the <code>WSS:UsernameToken</code> element for authentication. If this checkbox is selected, BusinessConnect Container Edition - ebXML Protocol works as follows:</p> <ul style="list-style-type: none"> <li>• Sends the outbound action message with <code>WSS:UsernameToken</code> where the username and password are provided by the private process (the message producer).</li> <li>• Requires that the inbound message of the same operation also carries the <code>WSS:UsernameToken</code> element to be used for inbound message authentication. If the inbound message does not have the <code>WSS:UsernameToken</code> element, the message is rejected with an error message being sent to the trading partner.</li> </ul>
<b>Receipt Reply Pattern</b>	<p>The mode in which the receipt signals are to be replied. The options are as follows:</p> <ul style="list-style-type: none"> <li>• <b>callback</b> The MSH receipt signals are sent back in a separate transport channel, which is also called asynchronous mode. The original transport channel is closed after the message is received by the receiving server.</li> <li>• <b>response</b> The MSH receipt signals are sent back through the back channel of the original transport, which is also called synchronous mode.</li> </ul> <p>See <a href="#">Receipt Reply Pattern</a>.</p>

Field	Description
<b>Eliminate Duplicate Message</b>	<p>Messages that are considered duplicated are eliminated.</p> <p>BusinessConnect Container Edition - ebXML Protocol keeps the record of recent sent and received messages in database table. A message is considered duplicated if its message ID matches one of the messages in record. BusinessConnect Container Edition - ebXML Protocol checks the messages received from private processes on the sending side, or the messages received from trading partners on the receiving side for duplications. The options are as follows:</p> <ul style="list-style-type: none"> <li>• <b>never</b> Do not eliminate duplicate messages. If this option is selected, messages that are found to be duplicated are not eliminated. This applies to both the messages received from private processes on the outbound side, or the messages received from trading partners on the inbound side.</li> <li>• <b>always</b> Always eliminate duplicate messages. If this option is selected, the messages that are found to be duplicated are eliminated, and the messages cannot be sent out, or delivered to its destination. This applies to both the inbound and outbound side.</li> <li>• <b>perMessage</b> This only applies to the outbound side. Once it is selected, the <code>dupElimination</code> property in the messages sent from private processes determines whether the message is to be eliminated if it is considered duplicated. This provides you more flexibility for controlling this behavior on each message.</li> </ul>

## Receipt Tab

<b>Require MSH Receipt</b>	<p>Require the MSH receipt for the action message. This applies on the message receiving side to determine whether a receipt is to be created and sent to the message sending side.</p> <p>On the message sending side, a receipt signal is sent back if this is configured by the sending MSH, so the Reception Awareness function is achieved. If the receipt is not received within a period of time, the original message can be resent. The options are as follows:</p> <ul style="list-style-type: none"> <li>• <b>never</b> Do not generate receipt signal for the received action</li> </ul>
----------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Field	Description
	<p>message.</p> <ul style="list-style-type: none"> <li>• <b>always</b> Always generate a receipt signal for the received action message.</li> </ul>
<b>Time to Wait for MSH Receipt (min.)</b>	<p>The length of time to wait for an MSH receipt signal from a trading partner after an action message has been sent out. After this time expires, the action message is resent to the trading partner as many times as specified in the <b>Maximum Number of Retries</b> field. After the last retry attempt, BusinessConnect Container Edition - ebXML Protocol sends an error to the private process.</p> <p>The resending of an action message is different from the HTTP connection retries defined in outbound HTTP transport, which is used to communicate with a trading partner. HTTP connection retries, defined in the HTTP transport configuration, are always performed when the initiator fails to connect to the HTTP server of the responder to send an action message. Ensure that the HTTP connection retries do not interfere with the resending of an action message. The value in this field must be greater than the value based on the formula of HTTP and HTTPS transports: <math>\text{Retry Interval} \times \text{Retry Count}</math>.</p> <p>When an action message is sent by the HTTP transport, if no response is received within the time defined in the <b>Socket Timeout (seconds)</b> field in the outbound HTTP transport configuration, the initiator stops the connection. This is considered as a scenario in which the resending of an action message is necessary. The value in this field must be greater than the value for HTTP and HTTPS transport socket timeout.</p> <p>See HTTP, HTTPS, and HTTPSCA transports in <i>TIBCO BusinessConnect™ Container Edition Trading Partner Management</i>.</p>
<b>Maximum Number of Retries</b>	<p>The number of times to retry sending the action message before receiving an MSH receipt.</p>
<b>Require Non-repudiation of MSH Receipt</b>	<p>Require the MSH receipt to be digitally signed and the receipt contains the digital digest of the received message. This applies on the message receiving side.</p>

Field	Description
	On the message sending side, the replied receipt signal is checked if it is signed properly, and the digital digest contained in the receipt is matched with the preserved digest of the original message for data integrity.

## Pull Transaction Properties

If you select **New Pull Transaction** in [Adding Transactions](#), you see the **Pull Transaction**, **Pull Request Action**, and **Pull Response Action** tabs when you configure the transaction in the **New Pull Transaction** dialog.



**Tip:** You can override certain transaction settings when configuring protocol bindings for a business agreement. See [Operation Settings Tab](#).  
The `tibco.com/EBMS Pull Request/1.0/Pull Request` pull request operation is predefined with the installation of the product and activation of BusinessConnect Container Edition - ebXML Protocol. This is a fixed operation that is used for guiding the processing of an inbound pull request and cannot be removed. However, you can override it in the protocol bindings of a business agreement.

The following table describes each field on the **Pull Transaction** tab.

### Pull Operation Type: Pull Transaction Tab

Field	Description
<b>General Tab</b>	
<b>Name</b>	The displayed name of the transaction.
<b>Description</b>	Optional. A short description of the transaction.
<b>Roles Tab</b>	

Field	Description
<b>Initiating Role</b>	The role of the initiator in this transaction.
<b>Responding Role</b>	The role of the responder in this transaction. Unlike the one-way push transaction, the initiator role and responding role properties are not present in the pull request and response messages.

## Service Information Tab

<b>Service</b>	The service that acts on this message. Unlike the one-way push transaction, the service and service type properties are present in the pull request messages.
<b>Service Type</b>	The service type.
<b>P-Mode ID</b>	The identifier for the processing mode defined by the transaction. Unlike the one-way push transaction, the P-Mode ID of the one-way pull transaction is not present in the pull request messages.

The following table describes each field on the **Pull Request Action** tab.

### Pull Operation Type: Pull Request Action Tab

Field	Description
<b>General Tab</b>	
<b>Name</b>	The name of the action.
<b>Description</b>	Optional. A description of the action.
<b>Direction</b>	The direction of this action.
<b>Require Non-repudiation of Request</b>	<p>Require the action message to be digitally signed for non-repudiation. If this checkbox is selected, BusinessConnect Container Edition - ebXML Protocol works as follows:</p> <ul style="list-style-type: none"> <li>Signs the outbound action message and stores the message in the non-repudiation log.</li> </ul>

Field	Description
<b>Require UsernameToken Authentication of Request</b>	<ul style="list-style-type: none"> <li>Requires that the inbound action message of this operation is also digitally signed; if it is not signed, the message is considered invalid and an error message is sent back.</li> </ul> <p>Require the action message to contain the WSS:UsernameToken element for authentication. If this checkbox is selected, BusinessConnect Container Edition - ebXML Protocol works as follows:</p> <ul style="list-style-type: none"> <li>Sends the outbound action message with WSS:UsernameToken where the username and password are provided by the private process (the message producer).</li> <li>Requires that the inbound message of the same operation also carries the WSS:UsernameToken element to be used for inbound message authentication. If the inbound message does not have the WSS:UsernameToken element, the message is rejected with an error message being sent to the trading partner.</li> </ul>
<b>Eliminate Duplicate Message</b>	<p>The receiver of the action message must eliminate duplicate messages. BusinessConnect Container Edition - ebXML Protocol can remove duplicates of action messages that are kept in persistent storage. For a pull request, this property only applies to the outbound side, because no message ID information is in an inbound pull request message. The options are as follows:</p> <ul style="list-style-type: none"> <li><b>never</b> Do not eliminate duplicate messages.</li> <li><b>always</b> Always eliminate duplicate messages.</li> <li><b>perMessage</b> If this option is selected, BusinessConnect Container Edition - ebXML Protocol handles duplicate elimination according to the definition of the message.</li> </ul>

The following table describes each field on the **Pull Response Action** tab.

**Pull Operation Type: Pull Response Action Tab**

Field	Description
<b>General Tab</b>	
<b>Name</b>	The name of the action.
<b>Description</b>	Optional. A description of the action.
<b>Direction</b>	The direction of this action.

## Two-Way Transaction Properties

If you select **New Two-Way Transaction** in [Adding Transactions](#), you see the **Two-Way Transaction**, **Request Action**, and **Response Action** tabs when you configure the transaction in the **New Two-Way Transaction** dialog. A two-way transaction can be considered two associated one-way transaction, with the second transaction being the response of the first one. So the properties of the two-way transaction are almost the same as the one-way transaction.

 **Tip:** You can override certain transaction settings when configuring protocol bindings for a business agreement. See [Operation Settings Tab](#).

The following table describes each field on the **Two-Way Transaction** tab.

**Two-Way Operation Type: Two-Way Transaction Tab**

Field	Description
<b>General Tab</b>	
<b>Name</b>	The displayed name of the transaction.
<b>Description</b>	Optional. A short description of the transaction.

Field	Description
<b>Roles Tab</b>	
<b>Initiating Role</b>	The role of the initiator in this transaction. This field is mapped to the respective Role element in the eb:PartyId/eb:From element of the public user message.
<b>Responding Role</b>	The role of the responder in this transaction. This field is mapped to the respective Role element in the eb:PartyId/eb:To element of the public user message.
<b>Service Information Tab</b>	
<b>Service</b>	The service that acts on this message, as defined by ebMS. This field is mapped to the Service element in the public message SOAP header.  You must specify a URI in this field if the <b>Service Type</b> field is empty.
<b>Service Type</b>	The service type, as defined by ebMS. This field is mapped to the Type attribute in the Service element of the public message.  You must specify a URI in the <b>Service</b> field if this field is empty.
<b>AgreementRef</b>	The reference of the agreement between the host and the trading partner. Such an agreement is the set of P-Mode operations that are essential for the two parties for their MSHs to interoperate. This reference is present in the eb:CollaborationInfo/eb:AgreementRef element. In inbound messages sent from this trading partner, the eb:AgreementRef element must have the same value to be accepted and processed.  This field takes precedence over the setting on trading partners level, and you can override this value on the business agreements level.
<b>Type</b>	The type of the AgreementRef value. If it is empty, the AgreementRef value must be a valid URI.  This field takes precedence over the setting on trading partners level, and you can override this value on the business agreements level.

Field	Description
<b>P-Mode ID</b>	<p>The identifier for the processing mode defined by the transaction. It is also called operation definition in BusinessConnect Container Edition. This is the default P-Mode ID for this P-Mode. You can override the P-Mode ID by using a special P-Mode ID agreed between you and your trading partner when configuring protocol bindings for a business agreement to identify the specific P-Mode used between you and your trading partner for guiding this particular transaction.</p> <p><b>Note:</b> The combination of service, service type, the request action name, and P-Mode ID is the identification of a business transaction. These values are present in user messages, which are used by the receiver to recognize which P-Mode or operation type is used to process a received user message.</p>

## Document Tab

<b>Encrypt Document</b>	Select this checkbox to encrypt the outbound document. When this is selected, it is also required that the inbound message of this operation must be encrypted.
<b>Compress Document</b>	<p>Selecting this checkbox requires that the outbound documents of this operation are compressed. When this is selected, the contents of each attachment in the message are compressed.</p> <p><b>Note:</b> This option is used only to compress attachments, not to compress messages. When a message does not include attachments, the message is not compressed even if you select this checkbox.</p>

The following table describes each field on the **Request/Response Action** tab.

### Two-Way Operation Type: Request/Response Action Tab

Field	Description
<b>General Tab</b>	

Field	Description
<b>Name</b>	The name of the action.
<b>Description</b>	Optional. A description of the action.
<b>Direction</b>	The direction of this action.
<b>Max.Response Wait Time (min.)</b>	<p>Configure only on the <b>Request Action</b> tab.</p> <p>The maximum length of time to wait for the response message from a trading partner, counting from the time the action message is sent. If this time expires, BusinessConnect Container Edition - ebXML Protocol sends an ae/ebMS3/Advisory message on the error subject.</p> <div style="border: 1px solid #ccc; background-color: #f9f9f9; padding: 5px; margin-top: 10px;"> <p><b>Note:</b> This function does not take effect in two-way synchronous transactions.</p> </div>
<b>Max. Private Process Response Wait Time (min.)</b>	<p>Configure only on the <b>Response Action</b> tab.</p> <p>The maximum length of time to wait for the response message from a private process, counting from the time the action message is sent to the private process. If this time expires, BusinessConnect Container Edition - ebXML Protocol sends an MSH error to the trading partner.</p>
<b>Require Non-repudiation of Request/Response</b>	<p>Require the action message to be digitally signed for non-repudiation. If this checkbox is selected, BusinessConnect Container Edition - ebXML Protocol works as follows:</p> <ul style="list-style-type: none"> <li>• Signs the outbound action message and stores the message in the non-repudiation log.</li> <li>• Requires that the inbound action message of this operation is also digitally signed; if it is not signed, the message is considered invalid and an error message is sent back.</li> </ul>
<b>Require UsernameToken Authentication of Request/Response</b>	<p>Require the action message to contain the WSS:UsernameToken element for authentication. If this checkbox is selected, BusinessConnect Container Edition - ebXML Protocol works as follows:</p>

Field	Description
<b>Response Reply Pattern</b>	<p data-bbox="524 674 1312 741">Configure only on the <b>Request Action</b> tab. The options are as follows:</p> <ul data-bbox="573 779 1414 1066" style="list-style-type: none"> <li data-bbox="573 779 1325 888">• <b>synchronous</b> Waits for the response message from your trading partner through the back channel of the original transport.</li> <li data-bbox="573 919 1414 1066">• <b>asynchronous</b> Waits for the response message from your trading partner in a separate transport channel. The original transport channel is closed after the message is received by the receiving server.</li> </ul> <p data-bbox="524 1098 1081 1125">The default option is asynchronous pattern.</p> <div data-bbox="524 1157 1414 1297" style="background-color: #f0f0f0; padding: 10px;"> <p data-bbox="540 1171 1398 1276"><b>Note:</b> When the synchronous pattern is selected, ensure that you select <b>never</b> from the <b>Require MSH Receipt</b> list in the <b>Receipt</b> tab of the request and response.</p> </div>
<b>Receipt Reply Pattern</b>	<p data-bbox="524 1346 1398 1413">The mode in which the receipt signals are to be replied. The options are as follows:</p> <ul data-bbox="573 1444 1382 1738" style="list-style-type: none"> <li data-bbox="573 1444 1382 1598">• <b>callback</b> The MSH receipt signals are sent back in a separate transport channel, which is also called asynchronous mode. The original transport channel is closed after the message is received by the receiving server.</li> <li data-bbox="573 1629 1382 1738">• <b>response</b> The MSH receipt signals are sent back through the back channel of the original transport, which is also called synchronous mode.</li> </ul>

Field	Description
	See <a href="#">Receipt Reply Pattern</a> .
<b>Eliminate Duplicate Message</b>	<p>Messages that are considered duplicated are eliminated. BusinessConnect Container Edition - ebXML Protocol keeps the record of recent sent and received messages in database table. A message is considered duplicated if its message ID matches one of the messages in record. BusinessConnect Container Edition - ebXML Protocol checks the messages received from private processes on the sending side, or the messages received from trading partners on the receiving side for duplications. The options are as follows:</p> <ul style="list-style-type: none"> <li>• <b>never</b> Do not eliminate duplicate messages. If this option is selected, messages that are found to be duplicated are not eliminated. This applies to both the messages received from private processes on the outbound side, or the messages received from trading partners on the inbound side.</li> <li>• <b>always</b> Always eliminate duplicate messages. If this option is selected, the messages that are found to be duplicated are eliminated, and the messages cannot be sent out, or delivered to its destination. This applies to both the inbound and outbound side.</li> <li>• <b>perMessage</b> This only applies to the outbound side. Once it is selected, the <code>dupElimination</code> property in the messages sent from private processes determines whether the message is to be eliminated if it is considered duplicated. This provides you more flexibility for controlling this behavior on each message.</li> </ul>

## Receipt Tab

<b>Require MSH Receipt</b>	<p>Require the MSH receipt for the action message. This applies on the message receiving side to determine whether a receipt is to be created and sent to the message sending side.</p> <p>On the message sending side, a receipt signal is sent back if this is configured by the sending MSH, so the Reception Awareness function is achieved. If the receipt is not received within a period of time, the</p>
----------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Field	Description
<b>Time to Wait for MSH Receipt (min.)</b>	<p>original message can be resent. The options are as follows:</p> <ul style="list-style-type: none"> <li>• <b>never</b> Do not generate receipt signal for the received action message.</li> <li>• <b>always</b> Always generate a receipt signal for the received action message.</li> </ul> <p>The length of time to wait for an MSH receipt from a trading partner. After this time expires, the action message is resent to the trading partner as many times as specified in the <b>Maximum Number of Retries</b> field. After the last retry attempt, and another period of Time to Wait for MSH Receipt has passed, BusinessConnect Container Edition - ebXML Protocol sends an error to the private process.</p> <p>The resending of an action messages is different from the HTTP connection retries defined in outbound HTTP transport, which is used to communicate with a trading partner. HTTP connection retries, defined in the HTTP transport configuration, are always performed when the initiator fails to connect to the HTTP server of the responder to send an action message. Ensure that the HTTP connection retries do not interfere with the resending of an action message. The value in this field must be greater than the value based on the formula of HTTP and HTTPS transports: <math>\text{Retry Interval} \times \text{Retry Count}</math>.</p> <p>When an action message is sent by the HTTP transport, if no response is received within the time defined in the <b>Socket Timeout (seconds)</b> field in the outbound HTTP transport configuration, the initiator stops the connection. This is considered as a scenario in which the resending of an action message is necessary. The value in this field must be greater than the value for HTTP and HTTPS transport socket timeout.</p> <p>See HTTP, HTTPS, and HTTPSCA transports in <i>TIBCO BusinessConnect™ Container Edition Trading Partner Management</i>.</p>
<b>Maximum Number of Retries</b>	The number of times to retry sending the action message before receiving an MSH receipt.

Field	Description
<b>Require Non-repudiation of MSH Receipt</b>	<p>Require the MSH receipt to be digitally signed and the receipt contains the digital digest of the received message. This applies on the message receiving side.</p> <p>On the message sending side, the replied receipt signal is checked if it is signed properly, and the digital digest contained in the receipt is matched with the preserved digest of the original message for data integrity.</p>

## Push and Pull Transaction Properties

If you specify Push and Pull in [Adding Transactions](#), you see the **Push and Pull Transaction**, **Push Action**, and **Pull Action** tabs when you configure the transaction in the Edit Transaction panel. A push and pull transaction can be considered a combination of a one-way push transaction and a one-way pull transaction.

An initiator uses a push action to send a request user message to the responder, and then uses a pull action to pull the response user message. The pull action is a selective pulling, because this action only pulls the response message that corresponds to the original request.

**i Note:** You can override certain transaction settings when configuring protocol bindings for a business agreement. See [Operation Settings Tab](#).

The following table describes each field on the **Push and Pull Transaction** tab.

### Push and Pull Operation Type: Push and Pull Transaction Tab

Field	Description
<b>General Tab</b>	
<b>Name</b>	The displayed name of the transaction.
<b>Description</b>	Optional. A short description of the transaction.

Field	Description
<b>Roles Tab</b>	
<b>Initiating Role</b>	The role of the initiator in this transaction. This field is mapped to the respective <code>Role</code> element in the <code>eb:PartyId/eb:From</code> element of the public user message.
<b>Responding Role</b>	The role of the responder in this transaction. This field is mapped to the respective <code>Role</code> element in the <code>eb:PartyId/eb:To</code> element of the public user message.
<b>Service Information Tab</b>	
<b>Service</b>	The service that acts on this message, as defined by ebMS. This field is mapped to the <code>Service</code> element in the public message SOAP header.  You must specify a URI in this field if the <b>Service Type</b> field is empty.
<b>Service Type</b>	The service type, as defined by ebMS. This field is mapped to the <code>Type</code> attribute in the <code>Service</code> element of the public message.  You must specify a URI in the <b>Service</b> field if this field is empty.
<b>AgreementRef</b>	The reference of the agreement between the host and the trading partner. Such an agreement is the set of P-Mode operations that are essential for the two parties for their MSHs to interoperate. This reference is present in the <code>eb:CollaborationInfo/eb:AgreementRef</code> element. In inbound messages sent from this trading partner, the <code>eb:AgreementRef</code> element must have the same value to be accepted and processed.  This field takes precedence over the setting on trading partners level, and you can override this value on the business agreements level.
<b>Type</b>	The type of the <code>AgreementRef</code> value. If it is empty, the <code>AgreementRef</code> value must be a valid URI.  This field takes precedence over the setting on trading partners level, and you can override this value on the business agreements level.

Field	Description
<b>P-Mode ID</b>	<p>The identifier for the processing mode defined by this transaction. It is also called operation definition in BusinessConnect Container Edition. This is the default P-Mode ID for this P-Mode. You can override the P-Mode ID by using a special P-Mode ID agreed between you and your trading partner when configuring protocol bindings for a business agreement to identify the specific P-Mode used between you and your trading partner for guiding this particular transaction.</p> <p><b>Note:</b> The combination of service, service type, the request action name, and P-Mode ID is the identification of a business transaction. These values are present in user messages, which are used by the receiver to recognize which P-Mode or operation type is used to process a received user message.</p>

## Document Tab

<b>Encrypt Document</b>	Select this checkbox to encrypt the outbound document. When this checkbox is selected, it is also required that the inbound message of this operation is encrypted.
<b>Compress Document</b>	Selecting this checkbox requires that the outbound documents of this operation are compressed. When this checkbox is selected, the contents of each attachment in the message are compressed.

**Note:** This option is used only to compress attachments, not to compress messages. When a message does not include attachments, the message is not compressed even if you select this checkbox.

The following table describes each field on the **Push Action** tab.

### Push and Pull Operation Type: Push Action Tab

Field	Description
<b>General Tab</b>	

Field	Description
<b>Name</b>	The name of the action.
<b>Description</b>	Optional. A description of the action.
<b>Direction</b>	The direction of this action.
<b>Require Non-repudiation of Request</b>	<p>Require the action message to be digitally signed for non-repudiation. If this checkbox is selected, BusinessConnect Container Edition - ebXML Protocol works as follows:</p> <ul style="list-style-type: none"> <li>• Signs the outbound action message and stores the message in the non-repudiation log.</li> <li>• Requires that the inbound action message of this operation is also digitally signed; if it is not signed, the message is considered invalid and an error message is sent back.</li> </ul>
<b>Require UsernameToken Authentication of Request</b>	<p>Require the action message to contain the <code>WSS:UsernameToken</code> element for authentication. If this checkbox is selected, BusinessConnect Container Edition - ebXML Protocol works as follows:</p> <ul style="list-style-type: none"> <li>• Sends the outbound action message with <code>WSS:UsernameToken</code> where the username and password are provided by the private process (the message producer).</li> <li>• Requires that the inbound message of the same operation also carries the <code>WSS:UsernameToken</code> element to be used for inbound message authentication. If the inbound message does not have the <code>WSS:UsernameToken</code> element, the message is rejected with an error message being sent to the trading partner.</li> </ul>
<b>Receipt Reply Pattern</b>	<p>The mode in which the receipt signals are to be replied. The options are as follows:</p> <ul style="list-style-type: none"> <li>• <b>callback</b> The MSH receipt signals are sent back in a separate transport channel, which is also called asynchronous mode. The original transport channel is closed after the message is received by the receiving server.</li> <li>• <b>response</b> The MSH receipt signals are sent back through the</li> </ul>

Field	Description
	<p>back channel of the original transport, which is also called synchronous mode.</p> <p>See <a href="#">Receipt Reply Pattern</a>.</p>
<b>Eliminate Duplicate Message</b>	<p>Messages that are considered duplicated are eliminated. BusinessConnect Container Edition - ebXML Protocol keeps the record of recent sent and received messages in database table. A message is considered duplicated if its message ID matches one of the messages in record. BusinessConnect Container Edition - ebXML Protocol checks the messages received from private processes on the sending side, or the messages received from trading partners on the receiving side for duplications. The options are as follows:</p> <ul style="list-style-type: none"> <li>• <b>never</b> Do not eliminate duplicate messages. If this option is selected, messages that are found to be duplicated are not eliminated. This applies to both the messages received from private processes on the outbound side, or the messages received from trading partners on the inbound side.</li> <li>• <b>always</b> Always eliminate duplicate messages. If this option is selected, the messages that are found to be duplicated are eliminated, and the messages cannot be sent out, or delivered to its destination. This applies to both the inbound and outbound side.</li> <li>• <b>perMessage</b> This only applies to the outbound side. Once it is selected, the <code>dupElimination</code> property in the messages sent from private processes determines whether the message is to be eliminated if it is considered duplicated. This provides you more flexibility for controlling this behavior on each message.</li> </ul>
<h2>Receipt Tab</h2>	
<b>Require MSH Receipt</b>	<p>Require the MSH receipt for the action message. This applies on the message receiving side to determine whether a receipt is to be created and sent to the message sending side.</p> <p>On the message sending side, a receipt signal is sent back if this is</p>

Field	Description
<b>Time to Wait for MSH Receipt (min.)</b>	<p>configured by the sending MSH, so the Reception Awareness function is achieved. If the receipt is not received within a period of time, the original message can be resent. The options are as follows:</p> <ul style="list-style-type: none"> <li>• <b>never</b> Do not generate a receipt signal for the received action message.</li> <li>• <b>always</b> Always generate a receipt signal for the received action message.</li> </ul> <p>The length of time to wait for an MSH receipt from a trading partner. After this time expires, the action message is resent to the trading partner as many times as specified in the <b>Maximum Number of Retries</b> field. After the last retry attempt, and another period of Time to Wait for MSH Receipt has passed, BusinessConnect Container Edition - ebXML Protocol sends an error to the private process.</p> <p>The resending of an action messages is different from the HTTP connection retries defined in outbound HTTP transport, which is used to communicate with a trading partner. HTTP connection retries, defined in the HTTP transport configuration, are always performed when the initiator fails to connect to the HTTP server of the responder to send an action message. Ensure that the HTTP connection retries do not interfere with the resending of an action message. The value in this field must be greater than the value based on the formula of HTTP and HTTPS transports: <math>\text{Retry Interval} \times \text{Retry Count}</math>.</p> <p>When an action message is sent by the HTTP transport, if no response is received within the time defined in the <b>Socket Timeout (seconds)</b> field in the outbound HTTP transport configuration, the initiator stops the connection. This is considered as a scenario in which the resending of an action message is necessary. The value in this field must be greater than the value for HTTP and HTTPS transport socket timeout.</p> <p>See HTTP, HTTPS, and HTTPSCA transports in <i>TIBCO BusinessConnect™ Container Edition Trading Partner Management</i>.</p>
<b>Maximum Number of Retries</b>	The number of times to retry sending the action message before receiving an MSH receipt.

Field	Description
<b>Require Non-repudiation of MSH Receipt</b>	<p>Require the MSH receipt to be digitally signed and the receipt contains the digital digest of the received message. This applies on the message receiving side.</p> <p>On the message sending side, the replied receipt signal is checked if it is signed properly, and the digital digest contained in the receipt is matched with the preserved digest of the original message for data integrity.</p>

The following table describes each field on the **Pull Action** tab.

#### Push and Pull Operation Type: Pull Action Tab

Field	Description
<b>General Tab</b>	
<b>Name</b>	The name of the action.
<b>Description</b>	Optional. A description of the action.
<b>Direction</b>	The direction of this action.

#### Pull Request Tab

<b>Require Non-repudiation of Request</b>	<p>Require the action message to be digitally signed for non-repudiation. If this checkbox is selected, BusinessConnect Container Edition - ebXML Protocol works as follows:</p> <ul style="list-style-type: none"> <li>• Signs the outbound action message and stores the message in the non-repudiation log.</li> <li>• Requires that the inbound action message of this operation is also digitally signed; if it is not signed, the message is considered invalid and an error message is sent back.</li> </ul>
<b>Require UsernameToken</b>	Require the action message to contain the <code>WSS:UsernameToken</code> element for authentication. If this checkbox is selected, BusinessConnect

Field	Description
<b>Authentication of Request</b>	<p>Container Edition - ebXML Protocol works as follows:</p> <ul style="list-style-type: none"> <li>• Sends the outbound action message with WSS:UsernameToken where the username and password are provided by the private process (the message producer).</li> <li>• Requires that the inbound message of the same operation also carries the WSS:UsernameToken element to be used for inbound message authentication. If the inbound message does not have the WSS:UsernameToken element, the message is rejected with an error message being sent to the trading partner.</li> </ul>
<b>Message Partition Channel</b>	<p>The URI that identifies the message partitioning channel to which the message is assigned. This can be present to both a push message and a message to be pulled.</p> <p><b>Note:</b> For a pull request by an initiator, if a value is provided by a TIBCO ActiveMatrix BusinessWorks private process, and another value is also configured in this field in TIBCO Administrator, BusinessConnect Container Edition uses the value in the private process.</p>
<b>Initial Polling Interval (sec.) (Polling Interval Doubled for each Subsequent Poll)</b>	<p>Specify the initial polling interval of the pull request.</p> <p>This polling interval doubles each time for the subsequent pull. The default value is 15. For example, if you use the default value 15, the first polling starts after 15 seconds, and then the next polling starts after 30 seconds.</p> <p><b>Note:</b></p> <ul style="list-style-type: none"> <li>• If the initiator does not require an MSH receipt of the push message, counting starts after the initiator sends a push message to a trading partner.</li> <li>• If the initiator requires an MSH receipt, counting starts after the initiator received the MSH receipt of the push message.</li> </ul>
<b>Maximum Polling Interval (sec.)</b>	<p>The maximum polling interval. The polling interval doubles each time for the subsequent pull. However, polling interval stops to increase</p>

Field	Description
	<p>after it reaches this maximum number.</p> <p>The default value is 120.</p>
<b>Maximum Polling Times</b>	<p>The number of times to retry pulling a user message from a trading partner.</p> <p>The default value is 3.</p>
Service Endpoint URL for Pull Request	<p>The service URL used as the outbound HTTP or HTTPS transport URL. The value in this field can override the URL configured in the outbound HTTP or HTTPS transport of your trading partner.</p> <p>If you do not provide a value in this field, the default HTTP or HTTPS transport URL of your trading partner is used.</p>

## Pull Response Tab

<b>Require Non-repudiation of Response</b>	<p>Require the action message to be digitally signed for non-repudiation. If this checkbox is selected, BusinessConnect Container Edition - ebXML Protocol works as follows:</p> <p>Signs the outbound action message and stores the message in the non-repudiation log.</p> <p>Requires that the inbound action message of this operation is also digital signed; if it is not signed, the message is considered invalid and an error message is sent back.</p>
<b>Require UsernameToken Authentication of Response</b>	<p>Require the action message to contain the WSS:UsernameToken element for authentication. If this checkbox is selected, BusinessConnect Container Edition - ebXML Protocol works as follows:</p> <ul style="list-style-type: none"> <li>• Sends the outbound action message with WSS:UsernameToken where the username and password are provided by the private process (the message producer).</li> <li>• Requires that the inbound message of the same operation also carries the WSS:UsernameToken element to be used for inbound message authentication. If the inbound message does not have</li> </ul>

Field	Description
	the WSS:UsernameToken element, the message is rejected with an error message being sent to the trading partner.
<b>Max. Private Process Response Wait Time (min.)</b>	The maximum length of time to wait for the response message from a private process, counting from the time the action message is sent to the private process. If this time expires, BusinessConnect Container Edition - ebXML Protocol sends an MSH error to the trading partner.

## Exporting Transactions

You can export all the ebMS3 transaction configurations. You can also export a subset of the transactions: by individual transaction (Export Transaction), by version container (Export Version), by business process container (Export Business Process), or by organization container (Export Organization).

To export ebMS3 transactions, perform the following steps:

### Procedure

1. On the **B2B Administration** tile, click **Operations Editor**.
2. To export all ebMS3 transactions:
  - a. Select the **ebMS3** checkbox.
  - b. Click the **Export** icon.
  - c. On the **Export Operations** dialog, click **Export**.  
You can now see the File Download dialog with `operations.bcce` as the suggested filename.
3. Click **Save** in the File Download dialog.
4. Alternatively, to export a subset of the ebMS3 transactions:
  - a. On the Operations Editor | ebMS3 page, select the organization, business process, version, or transaction you want to export.
  - b. Click the **Export** icon.

- c. On the **Export Operations** dialog, click **Export**.

You can now see the File Download dialog with `operations.bcce` as the suggested filename.

5. Change to a desired filename and click **Save**.

## Importing Transactions

To import ebMS3 transactions, perform the following steps:

### Procedure

1. On the **B2B Administration** tile, click **Operations Editor**.
2. From the **Protocol** list, click **ebMS3**.
3. Click the **Import** icon.
4. In the **Import Operations Data** dialog, click **Upload file**.  
Enter the password, if required.
5. Locate the exported `.bcce` file and click **Open**.
6. Click **Import**.

# Setting Up Trading Hosts and Partners

---

This section explains how to set up trading hosts and partners in BusinessConnect Container Edition - ebXML Protocol.

## Configuring ebMS3/AS4 Standard for Trading Hosts

This section explains how to configure the ebMS3/AS4 Standard for a trading host in BusinessConnect Container Edition. It assumes that you have already set up a host participant. For more information about setting up a trading host, see *TIBCO BusinessConnect™ Container Edition Trading Partner Management*.

To configure the ebXML protocol for a trading host, perform the following steps:

### Procedure

1. On the **Partner Management** tile, click **Hosts**.
2. Click the name of a host participant.
3. On the **Protocols** tab select the **ebMS3** checkbox.
4. Click **Edit Configurations** next to the **ebMS3** checkbox.
5. Configure [General Tab](#).
6. Click **Save**.

## General Tab

To manage general information for the ebMS3 trading host, configure the **General** tab.

Field	Description
<b>Default Domain Identity</b>	The default domain identity to use for this host. Select from the list of domain-identity pairs that you created in the Domain Identities dialog. See <a href="#">Adding Domain Identities</a> to add or edit a domain identity. This domain and identity are present in the eb:PartyInfo/eb:From/eb:PartyId element on outbound message sent to trading partners, or in the eb:PartyInfo/eb:To/eb:PartyId element in an inbound message received from trading partners.
<b>Valid Email Address List</b>	Currently not used.

## Adding Domain Identities

You must configure at least one domain-identity pair for your host participant. The domain value in the domain-identity pair corresponds to the `type` attribute of the `<PartyId>` element in the public message, and the identity value corresponds to the `<PartyId>` element itself.

To add a domain-identity pair for the ebXML protocol, perform the following steps:

### Procedure

1. On the **General** tab, click the **Add** icon  next to the **Default Domain Identity** field.
2. On the **Add New** tab, from the **Domain Type** dropdown list, select a domain.  
This is mapped to the domain string in the host domain-identity pair. If the domain you want to use does not exist, see [Domains](#) for instructions on adding domains.
3. In the **ID** field, enter the identity of the host participant.
4. Click **Add**.

**i Note:** If the domain you specify is URI, you must specify a valid URI in the **ID** field, for example, **urn:123456789**. However, the domain URI normally does not have to be present in outbound or inbound messages, as this is the default party ID type for ebMS3 standard. For example, if URI is used as the domain type of the trading partner, in outbound messages sent to trading partners, the domain URI is not present as the `eb:PartyInfo/eb:To/eb:PartyId/@type` attribute. Similarly, in an inbound message if the `eb:PartyInfo/eb:From/ebPartyId/@type` attribute is absent, the URI is considered the domain type.

## Configuring ebMS3/AS4 Standard for Trading Partners

This section explains how to configure the ebMS3/AS4 Standard protocol for a trading partner in BusinessConnect Container Edition. It assumes that you have already set up a partner participant. For more information about setting up a trading partner, see *TIBCO BusinessConnect™ Container Edition Trading Partner Management*.

To configure the ebMS3/AS4 Standard for a trading partner, perform the following steps:

### Procedure

1. On the **Partner Management** tile, click **Partners**.
2. Click the name of a trading partner.
3. On the **Protocols** tab select the **ebMS3** checkbox.
4. Click **Edit Configurations** next to the **ebMS3** checkbox.
5. Configure the following tabs:
  - **General**  
See [General Tab](#) for more information on configuring this tab.
  - **Transports**  
See [Transports Tab](#) for more information on configuring transports.
6. Click **Save**.

## General Tab

To manage general information for the ebMS3 trading partner, configure the **General** tab.

### Trading Partner General Properties

Field	Description
<b>Default Domain Identity</b>	The default domain identity to use for this partner. Select from the list of domain identities created in the Domain Identity dialog. See <a href="#">Adding Domain Identities</a> to add or edit a domain identity. This domain and identity are present in the eb:PartyInfo/eb:To/eb:PartyId element on outbound message sent to trading partners, or in the eb:PartyInfo/eb:From/eb:PartyId element in an inbound message received from trading partners.
<b>Valid Email Address List</b>	Currently not used.
<b>Security Token Reference</b>	<p>The Web Services Security security token reference mechanism. Configure this field so that a compatible X509 token type can be referenced by the wsse:SecurityTokenReference element in the signature and encryption elements of the message.</p> <p>The following supported mechanisms can be used to reference a security token and other key bearing elements:</p> <ul style="list-style-type: none"> <li>• <b>BinarySecurityToken:</b> Reference to a Binary Security Token.</li> <li>• <b>X509IssuerSerial:</b> Reference to an Issuer and Serial Number. The default value is X509IssuerSerial.</li> <li>• <b>X509SubjectKeyIdentifier:</b> Reference to a Subject Key Identifier.</li> </ul> <p><b>Note:</b> For more details on Security Token References, see <a href="http://docs.oasis-open.org/wss/v1.1/wss-v1.1-spec-errata-os-SOAPMessageSecurity.pdf">http://docs.oasis-open.org/wss/v1.1/wss-v1.1-spec-errata-os-SOAPMessageSecurity.pdf</a></p>

### Agreement Reference

<b>AgreementRef</b>	The reference of the agreement between the host and this particular
---------------------	---------------------------------------------------------------------

Field	Description
	<p>trading partner. Such an agreement is the set of P-Mode operations that are essential for the two parties for their MSHs to interoperate. This reference is present in the eb:CollaborationInfo/eb:AgreementRef element. In inbound messages sent from this trading partner, the eb:AgreementRef must have the same value to be accepted and processed.</p> <p><b>Note:</b> If you enter NULL in this field, this parameter is ignored, and no AgreementRef element is contained in the outgoing ebMS3 message.</p>
Type	<p>The type of the AgreementRef value. If it is empty, the AgreementRef value must be a valid URI.</p> <p><b>Note:</b> If you enter NULL in this field, this parameter is ignored.</p>

## Adding Domain Identities

You must configure at least one domain-identity pair for your partner participant. The domain value in the domain-identity pair corresponds to the type attribute of the PartyId element of the public message, and the identity value corresponds to the PartyId element itself.

To add a domain-identity pair for the ebXML protocol, perform the following steps:

### Procedure

1. On the **General** tab, click the **Add** icon  next to the **Default Domain Identity** field.
2. On the **Add New** tab, from the **Domain Type** dropdown list, select a domain.
 

This is mapped to the domain string in the host domain-identity pair. If the domain you want to use does not exist, see [Domains](#) for instructions on adding domains.
3. In the **ID** field, enter the identity of the host participant.
4. Click **Add**.

**i Note:** If the domain you specify is URI, you must specify a valid URI in the **Identity** field, for example, **urn:123456789**. However, the domain URI normally does not have to be present in outbound or inbound messages, as this is the default party ID type for ebMS3 standard. For example, if URI is used as the domain type of the trading partner, in outbound messages sent to trading partners, the domain URI is not present as the `eb:PartyInfo/eb:To/eb:PartyId/@type` attribute. Similarly, in an inbound message if the `eb:PartyInfo/eb:From/ebPartyId/@type` attribute is absent, the URI is considered the domain type.

## Transports Tab

On the **Transports** tab, configure outbound transport settings for sending ebXML messages to the trading partner.

To add an outbound transport, perform the following steps:

### Procedure

1. On the **Transports** tab, click **Add Outbound Transport**.
2. In the **Add Transport** dialog:
  - a. In the **Transport Name** field, enter the transport name.
  - b. From the **Transport Type** dropdown list, select the transport type.

**i Note:** The following transports are available to use with BusinessConnect Container Edition - ebXML Protocol:

- HTTP
- HTTPS

The steps required for configuring transports are the same for all protocols. For more information, see the transports section in *TIBCO BusinessConnect™ Container Edition Trading Partner Management*.

# Configuring Agreement Protocol Bindings

---

This section describes setting up business agreements to use BusinessConnect Container Edition - ebXML Protocol.

## Configuring Business Agreements with ebMS3 Protocol ebMS3/AS4 Standard

To configure a business agreement with the ebMS3 protocol binding, perform the following steps:

### Procedure

1. On the **Partner Management** tile, click **Business Agreements**.
2. Click the **<host-partner>** agreement that you want to edit.
3. On the **Bind Protocol** tab, select the **ebMS3** checkbox.
4. Click **Edit Configurations** next to the **ebMS3** checkbox.
5. Configure the following tabs:
  - **Operations** See [Configuring Operation Bindings](#).
  - **Document Exchange** [Setting Document Security Properties](#).
  - **Transports** See [Configuring Transports](#).
  - **Override Configuration** See [Overriding Participant Settings](#).
6. Click **Save**.

## Configuring Operation Bindings

Use the **Operations** tab to configure the ebMS3 transactions that each party in a business agreement can initiate and respond to.

- **Allow All Operations**

All transactions configured in the Operations Editor between participants can be used. If you select this option, you can still modify the behavior of one or more transactions by binding the transactions in the **Initiating Operations** and **Responding Operations** sections.

If you do not select this option, you must explicitly bind each transaction for each party in the **Initiating Operations** and **Responding Operations** sections.

## Binding Operations

The **Initiating Operations** section lists the operations that the host can initiate in this agreement. The **Responding Operations** section lists the operations that the partner can initiate.

To bind operations, perform the following steps:

### Procedure

1. Click **Add Operations**.
2. Click the **Caret** icon  to expand the browser tree.
3. Select the ebMS3 transactions that you want to bind.
4. Click **Done**.

## Editing Operation Bindings

To edit an ebMS3 operation binding, perform the following steps:

### Procedure

1. Click the name of an operation in the **Initiating Operations** or **Responding Operations** section.
2. Configure this particular operation binding on the following tabs:
  - a. **Operation Settings**

Override the default settings for this transaction. See [Operation Settings Tab](#).

**b. Action Settings**

Override settings chosen in [Action Settings Tab](#).

**c. Transports**

Override the default transport settings for this transaction. See [Transports Tab](#).

3. Click **Save**.

## Operation Settings Tab

You can override the default operation settings. For information about configuring the default settings of a transaction, see [Adding Transactions](#).

To override the default operation settings, perform the following steps:

### Procedure

1. On the **Override Configuration** tab, enable the **Override Settings** toggle.
2. Select one of the following items from the list to override operation settings for this transaction:

- **Roles**

Selecting this item to override the fields on the **Roles** tab for this transaction. See the Roles Tab section in [One-Way Operation Type: One-Way Transaction Tab](#) for a description of each field.

- **Service Information**

Selecting this item to override the **P-Mode ID** field on the **Service Information** tab for this transaction. The P-Mode ID that you provide here identifies the specific settings in this operation binding, and overrides the default P-Mode ID configured for this transaction. The service and type values cannot be overridden. See the Service Information Tab section in [One-Way Operation Type: One-Way Transaction Tab](#) for a description of the **P-Mode ID** field.

- **Document**

Selecting this item to override the fields on the **Document** tab for this transaction. See the Document Tab section in [One-Way Operation Type: One-Way Transaction Tab](#) for a description of each field.

## Action Settings Tab

You can also override the default settings of a specific action. You can find the default action settings of a transaction in the Operations Editor.

**i** **Note:** Operations you select in the **Initiating Operations** section can only have their **Request Action** tab settings overridden, while operations you select in the **Responding Operations** section can only have their **Response Action** tab settings overridden.

To override action settings, perform the following steps:

### Procedure

1. Click the **Action Settings** tab.
2. Select the **Override Action Settings** checkbox.
3. Select one of the following items from the list to override operation settings for this transaction:
  - a. **General** Select this item to override the fields on the **General** tab for this action.
  - b. **Receipt** Select this item to override the fields on the **Receipt** tab for this action.

## Transports Tab

After you define transport settings for a business agreement (see [Configuring Transports](#)), you can override the transport settings for a specific transaction.

To override outbound transport settings, perform the following steps:

### Procedure

1. Click the **Transports** tab.
2. Select the **Override Transports** checkbox.
3. Select the transport that you want to use from the **Primary Transport** list.

# Setting Document Security Properties

On the **Document Exchange** tab, specify security information for the transacted ebXML documents between the participants in this business agreement. The keys and certificates selected on the tab are configured on the **Credentials** tab of a participant. For information about configuring participant credentials, see *TIBCO BusinessConnect™ Container Edition Trading Partner Management*.

Follow the instructions in [Configuring Business Agreements with ebMS3 Protocol ebMS3/AS4 Standard](#) to get to the **Document Exchange** tab. See the following table to configure the document security properties.

## Document Security Properties

Field	Description
Outbound Document Exchange	
<b>Signing Key</b>	<p>The private key of the selected host. BusinessConnect Container Edition - ebXML Protocol uses this key to sign outbound documents in this business agreement.</p> <p>You must have already installed a private key for this host during participant configuration. See <i>TIBCO BusinessConnect™ Container Edition Trading Partner Management</i> for more information on installing private keys for host participants.</p> <div style="border: 1px solid #ccc; padding: 5px; background-color: #f9f9f9;"> <p><b>Note:</b> In special cases, the key is automatically retrieved from a specific keystore conforming to SBR. The keystore is configured on the Host layer.</p> </div>
<b>Digest Algorithm</b>	<p>This list has multiple digest algorithm options, among them either SHA1 or SHA256 can be selected for the digest algorithm used by the digital signature of ebMS3/AS4 Standard.</p>
<b>Encryption Certificate</b>	<p>The public certificate of the selected partner. BusinessConnect Container Edition - ebXML Protocol uses this certificate to encrypt outbound messages in this business agreement.</p> <p>You must have already installed a certificate for this partner during participant configuration. See <i>TIBCO BusinessConnect™ Container Edition Trading Partner Management</i> for more information on installing certificates</p>

Field	Description
	for partner participants.
<b>Encryption Algorithm</b>	The algorithm used to encrypt documents. The available options are DES3, AES-128, AES-192, AES-256, AES-128-GCM, AES-192-GCM, AES-256-GCM.
<b>Inbound Document Exchange</b>	
<b>Verification Certificate</b>	<p>The public key of the selected partner. BusinessConnect Container Edition - ebXML Protocol uses this key to verify the signed inbound documents in this business agreement.</p> <p>You must have already installed a certificate for this partner during participant configuration. See <i>TIBCO BusinessConnect™ Container Edition Trading Partner Management</i> for more information on installing certificates for partner participants.</p>
<b>Decryption Key</b>	<p>The private key of the selected host. BusinessConnect Container Edition - ebXML Protocol uses this key to decrypt inbound messages in this business agreement.</p> <p>You must have already installed a private key for this host during participant configuration. See <i>TIBCO BusinessConnect™ Container Edition Trading Partner Management</i> for more information on installing private keys for host participants.</p>

## Configuring Transports

Specify the transport for the host and the partner in a business agreement of the ebXML protocol binding.

Follow the instructions in [Configuring Business Agreements with ebMS3 Protocol ebMS3/AS4 Standard](#) to get to the **Transports** tab, and read this section for more information on how to configure in this tab.

## Outbound Transports for Host

On this section of the **Transports** tab, assign a transport for sending business transaction messages to the trading partner. The transport methods are configured as a part of the partner configuration: **BusinessConnect Container Edition > Partner Management > Partners > Name > Protocols > ebMS3 > Edit Configurations > Transports**. Click **Add Outbound Transport** and add.

For more information about configuring transports, see *TIBCO BusinessConnect™ Container Edition Trading Partner Management*.

The following table lists and describes the options available for transport assignment with ebMS3.

### Outbound Transports for Host

Field	Description
<b>Primary Transport</b>	The transport for sending outbound messages among the multiple outbound transports you might have defined for this trading partner.
<b>Client Authentication Identity for HTTPS, FTPS, HTTPSCA</b>	The key identity used when the remote server requires client authentication for an HTTPS connection.

## Inbound Transport for Partner

On this section of the **Transports** tab, specify the transport types your partner can use to send ebMS3 messages to your BusinessConnect Container Edition. The transport types available for selection here reflect the transports configured for this BusinessConnect Container Edition deployment. Transports for the deployment are configured in BusinessConnect Container Edition: **BusinessConnect Container Edition > System Settings > Transport Protocols > Inbound Protocols**.

For more information about configuring transports on the **Inbound Protocols** tab, see *TIBCO BusinessConnect™ Container Edition Trading Partner Management*.

Depending on your deployment configuration, the following transports can be available for selection:

**Inbound Transport Fields**

Field	Description
<b>HTTPSCA</b>	Client Authentication can be used when you select HTTPS as the inbound transport.
<b>HTTPS</b>	HTTPS connections can be used from this partner directly.
<b>HTTP</b>	HTTP connections can be used from this partner directly.

## Overriding Outbound Transport Settings

The selected outbound transport is the default transport of this Business Agreement. However, you can override this with another transport for a particular operation.

To override the outbound transport settings, perform the following steps:

**Procedure**

1. In the Edit Protocol Bindings: ebMS3 panel, click the **Operation Bindings** tab.
2. Click an operation in the **Initiating Operations** or **Responding Operations** section.
3. Continue by following the instructions in [Transports Tab](#).

## Overriding Participant Settings

You can use the ebMS3 panel to override the general protocol settings of each participant, which are configured on the **General** tab of the participant ebMS3 protocol configuration panel. For more information about configuring these default settings, see [Configuring ebMS3/AS4 Standard for Trading Hosts](#) and [Configuring ebMS3/AS4 Standard for Trading Partners](#).

To override the participant settings for a business agreement, perform the following steps:

**Procedure**

1. Follow the instructions in [Configuring Business Agreements with ebMS3 Protocol](#)

[ebMS3/AS4 Standard](#) to get to the Edit Protocol Bindings: ebMS3 panel.

2. Go to the **Override Configuration** tab.

Two configuration sections are displayed, one for each participant in this business agreement.

3. Enable the **Override Settings** toggle in the configuration section for the desired participant.
4. Configure the settings in the section. See [General Tab](#) for a host or [Trading Partner General Properties](#) for a partner.
5. Click **Save**.

# Private Processes

---

This section describes the interaction between BusinessConnect Container Edition - ebXML Protocol and its private processes and provides specifications for private process messages.

## Configuring Private Processes

BusinessConnect Container Edition Private Processes are enterprise back-office system or connections to the back-office system. They can be implemented by technology or any language, with the only requirement that they use JMS transport to communicate with the BusinessConnect Container Edition Interior Servers with certain format. They can also leverage TIBCO ActiveMatrix BusinessWorks to implement the private processes.

- **Standalone**

Standalone private processes are implemented by any technology or language, which use JMS to communicate with BusinessConnect Container Edition.

- **TIBCO ActiveMatrix BusinessWorks**

You can also use TIBCO ActiveMatrix BusinessWorks to implement the private processes, which leverage BusinessConnect palette that work inside TIBCO ActiveMatrix BusinessWorks to communicate with BusinessConnect Container Edition. You can also use TIBCO Business Studio to test your private processes in debug mode.

## Standalone Private Processes

If you want to design standalone private processes for use with BusinessConnect Container Edition - ebXML Protocol, you must understand JMS message exchange. BusinessConnect Container Edition - ebXML Protocol uses TIBCO ActiveExchange™ messages to exchange ebMS3 messages with the private processes. See [Private Process Message Formats](#) for formatting information of ebMS3 messages.

## TIBCO ActiveMatrix BusinessWorks Processes

The BusinessConnect palette, which is installed into TIBCO ActiveMatrix BusinessWorks during the TIBCO ActiveMatrix BusinessWorks™ Plug-in for BusinessConnect™ installation, encapsulates the message exchange configurations. You can also use a variety of TIBCO ActiveMatrix BusinessWorks palettes to define elaborate private processes to suit the needs of your business transactions.

For more information about using TIBCO ActiveMatrix BusinessWorks private processes, see [Configuring Private Processes with TIBCO ActiveMatrix BusinessWorks](#).

See [Tutorial](#) for configuring TIBCO ActiveMatrix BusinessWorks private processes to send and receive messages with BusinessConnect Container Edition - ebXML Protocol.

## Private Process Message Formats

BusinessConnect Container Edition - ebXML Protocol uses ActiveExchange™ messages to exchange ebMS3 messages with private processes through JMS. This section describes the ebXML-specific data classes that are used in the messages. You must format messages and include all required information as prescribed by the respective data classes before they are sent from private processes. By using TIBCO ActiveMatrix BusinessWorks and leveraging BusinessConnect palettes that work with TIBCO ActiveMatrix BusinessWorks, the message formatting is automatic.

For JMS transport, it transmits message data in the message body of JMS ObjectMessage messages. See the "Private Processes" section in *TIBCO BusinessConnect™ Container Edition Concepts*.

**i Note:** The JMS topics and queues that BusinessConnect Container Edition - ebXML Protocol uses do not contain the protocol name. Ensure that the JMSType attribute in the header element contains the string "ebMS3" when exchanging JMS messages with BusinessConnect Container Edition - ebXML Protocol.

See the following sections for formatting details of the ebMS3 message type:

- [Outbound Request Format](#)
- [Request Acknowledgment Format](#)
- [Inbound Response Format](#)

- [Inbound Request Format](#)
- [Outbound Response Format](#)
- [Response Acknowledgment Format](#)
- [Advisory Signal Message Format](#)
- [Error Message Format](#)

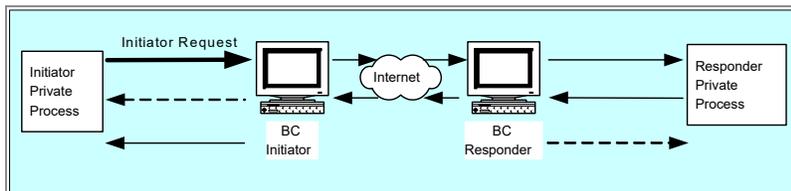
## Data Types and Data Fields

See the following section for formats of specific message fields and other data objects:

- [Additional Data Classes](#)

## Outbound Request Format

The initiator private process uses the following message class to send outbound request messages to BusinessConnect Container Edition - ebXML Protocol.



### Class

ae/ebMS3/InitiatorRequest

Example: AX.BC.ACME\_SERVER.ebMS3.INITIATOR.REQUEST

*prefix.installation*.INITIATOR.REQUEST (JMS queue)

### Message fields

See the following table for a description of each message field.

**ae/ebMS3/InitiatorRequest Message Fields**

Field	Type	Required	Description
<b>standardID</b>	String	Yes	The string "ebMS3".
<b>hostDomain</b>	String	No	<p>The domain name of the trading host, which is actually the PartyID Type of the host in ebMS3 transactions. BusinessConnect Container Edition - ebXML Protocol uses the <b>hostDomain</b> field and the <b>hostID</b> field as follows:</p> <ul style="list-style-type: none"> <li>• If <b>hostDomain</b> and <b>hostID</b> together match a domain-identity pair of a trading host, and then that host is used along with the matching domain-identity pair.</li> <li>• If only <b>hostDomain</b> is specified, and then the first domain-identity pair of any trading host that has a matching domain value is used.</li> <li>• If neither <b>hostDomain</b> nor <b>hostID</b> is specified, or if a match cannot be found for any host in the configuration store, and then the default domain identity of the default trading host is used.</li> </ul>
<b>hostID</b>	String	No	The identity value of the trading host domain-identity pair, which is

Field	Type	Required	Description
			actually the PartyID of the host in ebMS3 transactions. See the description for <b>hostDomain</b> on how this field is used.
<b>tpName</b>	String	Yes	The name of the trading partner in this transaction, as defined in the Partners panel in the BusinessConnect Container Edition console.
<b>tpDomain</b>	String	No	The domain name of the trading partner, which is actually the PartyID Type of the trading partner in ebMS3 transactions. Specify a domain here if you want to use a domain identity other than the one configured in the <b>Default Domain Identity</b> field of the trading partner Participants panel.
<b>operationID</b>	String	No	<p>The identifier for the transaction defined in ebMS3 operation definitions in the configuration store. Example: rosettanet.org/3A4/2.0/Request Purchase Order.</p> <p>This field is ignored when values are in all the following fields:</p> <ul style="list-style-type: none"> <li>• <b>organization</b></li> <li>• <b>businessProcess</b></li> <li>• <b>businessProcessVersion</b></li> <li>• <b>transactionName</b></li> </ul>

Field	Type	Required	Description
<b>organization</b>	String	No	<p>The organization name for the transaction. Example: rosettanet.org.</p> <p>When a value is in this field, there must be values in the <b>businessProcess</b>, <b>businessProcessVersion</b>, and <b>transactionName</b> fields. When this combination does not exist, the <b>operationID</b> field is used.</p>
<b>businessProcess</b>	String	No	<p>The name of the business process as defined by the organization you use. Example: 3A4.</p> <p>When a value is in this field, there must be values in the <b>organization</b>, <b>businessProcessVersion</b> and <b>transactionName</b> fields. When this combination does not exist, the <b>operationID</b> field is used.</p>
<b>businessProcess Version</b>	String	No	<p>The business process version. Example: 2.0.</p> <p>When a value is in this field, there must be values in the <b>organization</b>, <b>businessProcess</b> and <b>transactionName</b> fields. When this combination does not exist, the <b>operationID</b> field is used.</p>
<b>transactionName</b>	String	No	<p>The name of the transaction. Example: Request Purchase Order.</p>

Field	Type	Required	Description
			When a value is in this field, there must be values in the <b>organization</b> , <b>businessProcess</b> , and <b>businessProcessVersion</b> fields. When this combination does not exist, the <b>operationID</b> field is used.
<b>request</b>	String	Yes	<p>The request payload, which is usually an XML string. In the ebMS3 outbound request message, this is the content of the SOAP body.</p> <p><b>Note:</b> Text-based payloads and attachments must be encoded in UTF-8. In TIBCO ActiveMatrix BusinessWorks™ Plug-in for BusinessConnect™, the <b>body</b> field is used to specify the content of the <b>request</b> field. This field is ignored when the <b>requestFile</b> field is not empty.</p>
<b>requestFile</b>	String	No	<p>The file reference of the payload content. This can be used to send a large payload.</p> <p><b>Note:</b> This field takes precedence over the <b>request</b> field.</p>
<b>Message Properties</b>	N/A	No	<p>Specific message properties that can be used for handling the messages efficiently.</p> <p>The property names must be agreed by trading partners.</p>

Field	Type	Required	Description
<b>PayloadInfo</b>	N/A	No	Information of the payload of the request message. This is the payload to be sent in the SOAP body.  For more information about the subfields, see <a href="#">PayloadInfo</a> .
<b>o RSAKeyExchangeInfo</b>	sequence of ebMS3 //RSAKeyExchangeInfo	No	An element, which overrides the following algorithms from the private process: <ol style="list-style-type: none"> <li><b>KeyTransportAlgorithm:</b> Supported Encryption algorithm applied to the encrypted data. It supports the following algorithms: <ul style="list-style-type: none"> <li>rsa-1_5</li> <li>rsa-oaep</li> <li>rsa-oaep-mgf1p</li> </ul> </li> <li><b>RSAES-OAEP-Params:</b> Supports two parameters of the RSA OAEP algorithm in the private process: <ul style="list-style-type: none"> <li>MessageDigestAlgorithm (SHA-1, SHA-256, SHA-384, and SHA-512)</li> <li>MaskGenerationAlgorithm (SHA-1, SHA-224, SHA-256, SHA-384, and SHA-512)</li> </ul> </li> </ol>

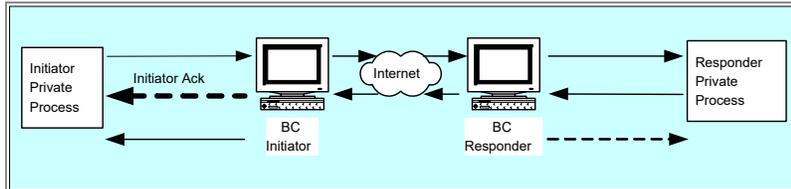
Field	Type	Required	Description
			<p><b>Note:</b> The type of the <b>KeyTransportAlgorithm</b> specified in the private process overrides the value of <code>ebms3.ob.keyExchange.algo</code> property defined in the ebXML plug-in properties.</p>
<b>attachment</b>	sequence of ebMS3/Info/Attachment	No	<p>A sequence of attachments. Each of the attachments is sent out in one of the SOAP Attachments in the public message.</p> <p>See <a href="#">ae/BC/Attachment Fields</a> for formatting information.</p>
<b>conversationID</b>	String	No	<p>A unique identifier unique to this conversation. If you do not supply a value here, it is automatically generated for the public message.</p>
<b>messageID</b>	String	No	<p>A unique identifier for the outbound ebMS3 message. It must conform to the format described in RFC 2822. Otherwise, your trading partner cannot process the message properly.</p> <p>If you do not supply a value here, it is automatically generated by the BusinessConnect Container Edition server for the public message.</p>
<b>toBePulled</b>	Boolean	No	<p>Set this field to be <code>true</code> only when the message is not intended to be sent out directly, but rather be stored in Store-and-Forward, to be</p>

Field	Type	Required	Description
			pulled by the trading partner.
<b>mpc</b>	String	No	The URI that identifies the message partitioning channel to which the message is assigned. This value can be present to both a push message and a message to be pulled depending on the <code>true</code> or <code>false</code> value in the <b>toBePulled</b> field. This value can be also used for a pull request, where it is used to indicate from which message partitioning channel the message is pulled.
<b>RefToMessageID</b>	String	No	The message ID of a previous message that this current message is related to. This property is typically used when the current message is a response message to a previous request message. This property can also be used for a selective pull request that pulls the user message with the same <code>RefToMessageID</code> value.
<b>authorization Token</b>	N/A	No	Include username and password information that is used for pull request authorization. For a pull request message, both the username and password must be specified. For a message to be stored in Store-and-Forward, only the username must be specified. For a message to be pushed out directly, this is not a required field.

Field	Type	Required	Description
<b>authenticationToken</b>	N/A	No	Include username and password information that is used for WSS UsernameToken authentication, which is an alternative other than the WSS signature authentication. This can be used for all types of messages.
<b>dupElimination</b>	Boolean	No	Request that the duplicate message sent by the private process is eliminated by the BusinessConnect Container Edition outbound process.
<b>closure</b>	String	No	<p>A unique string for BusinessConnect Container Edition - ebXML Protocol to keep track of this transaction. It is copied to all response messages of this request.</p> <p>If you do not supply a value here, it is automatically generated by BusinessConnect Container Edition - ebXML Protocol.</p>
<b>trackingID</b>	String	No	<p>An identifier that the private process can keep track of messages that are otherwise related to each other. It is copied to all response messages to this request.</p> <p>Contrary to the <b>closure</b> and <b>conversationID</b> fields, there is no requirement on the value of this field, nor is it used to process public messages.</p>

## Request Acknowledgment Format

BusinessConnect Container Edition - ebXML Protocol uses the following message class to indicate to the private process whether the request is sent to the trading partner successfully.



### Class

ae/ebMS3/InitiatorAck

### Subject

Example: AX.BC.ACME\_SERVER.ebMS3.INITIATOR.ACK

*prefix.installation*.INITIATOR.ACK (JMS queue)

### Message fields

See the following table for description of each message field.

#### ae/ebMS3/InitiatorAck Message Fields

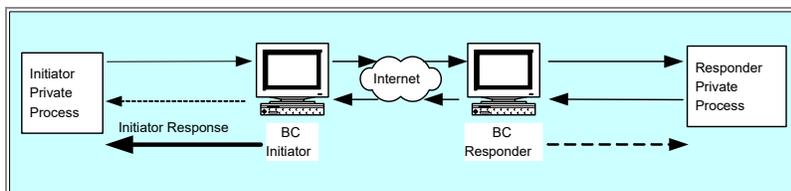
	Type	Required	Description
<b>standardID</b>	String	Yes	The string "ebMS3".
<b>tpName</b>	String	Yes	The name of the trading partner, as specified in the <b>tpName</b> field in the corresponding ae/ebMS3/InitiatorRequest message.
<b>tradingPartner</b>	trading partner	Yes	The trading partner. See <a href="#">ae/BC/TradingPartner Fields</a> for formatting information.

	Type	Required	Description
<b>host</b>	trading partner	Yes	The trading host. See <a href="#">ae/BC/TradingPartner Fields</a> for formatting information.
<b>operationID</b>	String	Yes	The identifier of the operation.Example: rosettanet.org/3A4/2.0/Request Purchase Order.
<b>organization</b>	String	Yes	The organization name for the transaction.
<b>businessProcess</b>	String	Yes	The name of the business process for the transaction.
<b>businessProcess Version</b>	String	Yes	The business process version for the transaction.
<b>transactionName</b>	String	Yes	The name of the transaction.
<b>agreementRef</b>	String	No	The value of the AgreementRef element in the ebMS3 Messaging header of the public message. This is configured in the <b>AgreementRef</b> field of the Participants panel for the trading partner. See <a href="#">Setting Up Trading Hosts and Partners</a> .
<b>conversationID</b>	String	Yes	The value of the ConversationId element in the ebMS3 messaging header of the public message.
<b>messageID</b>	String	Yes	The value of the MessageData/MessageId element in the ebMS3 messaging header of the public message.
<b>closure</b>	String	No	A unique string for BusinessConnect Container Edition - ebXML Protocol to keep track of this transaction. It is either generated by BusinessConnect Container

	Type	Required	Description
			Edition - ebXML Protocol or copied from the <b>closure</b> field in the corresponding ae/ebMS3/InitiatorRequest message.
<b>trackingID</b>	String	No	The value of the <b>trackingID</b> field in the corresponding ae/ebMS3/InitiatorRequest message, which can be used by the private process to keep track of messages that are related to each other.

## Inbound Response Format

BusinessConnect Container Edition - ebXML Protocol uses the following message class to forward inbound response messages to the initiator private process. This is in the scenario of a two-way transaction, where the response is replied by the responder in the second leg of the two-way transaction.



### Class

ae/ebMS3/InitiatorResponse

### Subject

Example: AX.BC.ACME\_SERVER.ebMS3.INITIATOR.RESPONSE

*prefix.installation*.INITIATOR.RESPONSE (JMS queue)

### Message fields

See the following table for a description of each message field.

**ae/ebMS3/InitiatorResponse Message Fields**

	Type	Required	Description
<b>standardID</b>	String	Yes	The string "ebMS3".
<b>fromTP</b>	trading partner	Yes	The trading partner. See <a href="#">ae/BC/TradingPartner Fields</a> for formatting information.
<b>host</b>	trading partner	Yes	The trading host. See <a href="#">ae/BC/TradingPartner Fields</a> for formatting information.
<b>fromRole</b>	String	No	The value of the <code>From/Role</code> element in the ebMS3 messaging header of the inbound public message.
<b>toRole</b>	String	No	The value of the <code>To/Role</code> element in the ebMS3 messaging header of the inbound public message.
<b>operationID</b>	String	Yes	The identifier of the transaction as defined in the operation definition in local configuration. Example: <code>rosettanet.org/3A4/2.0/Request Purchase Order</code> .
<b>organization</b>	String	Yes	The organization name for the transaction.
<b>businessProcess</b>	String	Yes	The name of the business process for the transaction.
<b>businessProcess Version</b>	String	Yes	The business process version for the transaction.
<b>transactionName</b>	String	Yes	The name of the transaction.
<b>statusCode</b>	Integer	Yes	A code indicating the status of the

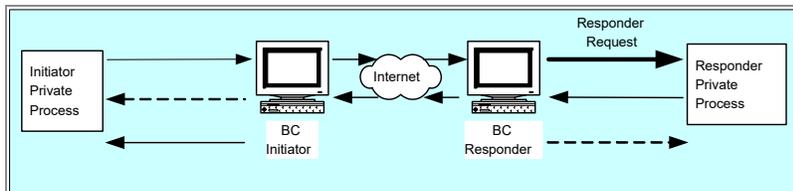
	Type	Required	Description
			transaction. See <a href="#">Status Codes</a> for more information.
<b>statusMsg</b>	String	Yes	A brief description of the code in the <b>statusCode</b> field.
<b>mpc</b>	String	No	The URI that identifies the message partitioning channel to which the message is assigned.
<b>timestamp</b>	String	No	The timestamp when the ebMS3 message is generated.
<b>response</b>	String	No	The response payload, which is usually an XML string. The content of this field comes from the content of the SOAP body in the inbound response ebMS3 message.  <div style="border: 1px solid #ccc; background-color: #f0f0f0; padding: 10px; margin-top: 10px;"> <p><b>Note:</b> In TIBCO ActiveMatrix BusinessWorks™ Plug-in for BusinessConnect™, the <b>body</b> field is used to specify the content of the <b>response</b> field. Also the field is ignored when the <b>responseFile</b> field is not empty</p> </div>
<b>responseFile</b>	String	No	The file reference when the size of the payload exceeds the private process file size threshold.  You can configure this by using the <code>ebxml.fileref.threshold</code> property from <b>BusinessConnect Container Edition &gt; System Settings &gt; Others &gt; Activate Protocol Plugins &gt; ebXML</b> .

	Type	Required	Description
			<p><b>Note:</b> This field takes precedence over the <b>response</b> field.</p>
<b>MessageProperties</b>	N/A	No	<p>Specific message properties as defined in the ebMS3 standard.</p> <p>The property names must be agreed by trading partners.</p>
<b>PayloadInfo</b>	N/A	No	<p>Information of the payload of the response message. This is the payload encapsulated as the content of the SOAP Body.</p> <p>See <a href="#">PayloadInfo</a> for more information about the subfields.</p>
<b>attachment</b>	sequence of ebMS3/Info/Attachment	No	<p>A sequence of attachments that are sent from a trading partner as the SOAP Attachments.</p> <p>See <a href="#">ae/BC/Attachment Fields</a> for formatting information.</p>
<b>agreementRef</b>	String	No	<p>The value of the AgreementRef element in the ebMS3 messaging header of the inbound public message.</p>
<b>conversationID</b>	String	Yes	<p>The value of the ConversationId element in the ebMS3 messaging header of the inbound public message.</p>
<b>messageID</b>	String	Yes	<p>The value of the MessageData/MessageId element in the ebMS3 messaging header of the inbound public message.</p>

	Type	Required	Description
<b>closure</b>	String	No	A unique string for BusinessConnect Container Edition - ebXML Protocol to keep track of this transaction. It is either generated by BusinessConnect Container Edition - ebXML Protocol or copied from the <b>closure</b> field in the corresponding <code>ae/ebMS3/InitiatorRequest</code> message.
<b>trackingID</b>	String	No	The value of the <b>trackingID</b> field in the corresponding <code>ae/ebMS3/InitiatorRequest</code> message, which can be used by the private process to keep track of messages that are otherwise related to each other.

## Inbound Request Format

BusinessConnect Container Edition - ebXML Protocol uses the following message class to forward inbound request messages to the responder private process.



### Class

`ae/ebMS3/ResponderRequest`

### Subject

Example: `AX.BC.ACME_SERVER.ebMS3.RESPONDER.REQUEST`

`prefix.installation.RESPONDER.REQUEST` (JMS queue)

## Message fields

See the following table for a description of each message field.

### ae/ebMS3/ResponderRequest Message Fields

Field	Type	Required	Description
<b>standardID</b>	String	Yes	The string "ebMS3".
<b>fromTP</b>	trading partner	No	The trading partner. See <a href="#">ae/BC/TradingPartner Fields</a> for formatting information.
<b>host</b>	trading partner	Yes	The trading host. See <a href="#">ae/BC/TradingPartner Fields</a> for formatting information.
<b>fromRole</b>	String	No	The role of the trading partner, which is copied from the From/Role element in the SOAP header of the inbound public message.
<b>toRole</b>	String	No	The role of the trading host, which is copied from the To/Role element in the SOAP header of the inbound public message.
<b>operationID</b>	String	Yes	The identifier of the operation. For example: rosettanet.org/3A4/2.0/Request Purchase Order.
<b>organization</b>	String	Yes	The organization name for the transaction.
<b>businessProcess</b>	String	Yes	The name of the business process for the transaction.
<b>businessProcess Version</b>	String	Yes	The business process version for the transaction.

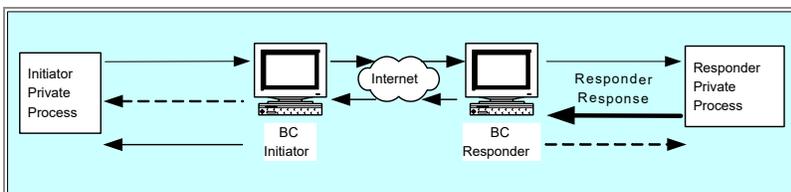
Field	Type	Required	Description
<b>transactionName</b>	String	Yes	The name of the transaction.
<b>service</b>	String	Yes	The service that acts on the message, which is copied from the <code>Service</code> element in the SOAP header of the inbound public message.
<b>serviceType</b>	String	No	The service type, which is copied from the <code>Service/@type</code> attribute in the SOAP header of the inbound public message.
<b>action</b>	String	Yes	The action name in the inbound message, which is copied from the <code>Action</code> element in the SOAP header of the inbound public message.
<b>mpc</b>	String	No	The message partitioning channel to which the message is assigned.
<b>timestamp</b>	String	No	The timestamp when the ebMS3 message is generated.
<b>request</b>	String	No	The request payload, which is usually an XML string. The content of this field comes from the SOAP body content of the inbound ebMS3 message.
			<p><b>Note:</b> In TIBCO ActiveMatrix BusinessWorks™ Plug-in for BusinessConnect™, the <b>body</b> field is used to specify the content of the <b>request</b> field. Also this field is ignored when the <b>requestFile</b> field is not empty.</p>
<b>requestFile</b>	String	No	The file reference when the size of the

Field	Type	Required	Description
			<p>payload exceeds the private process file size threshold.</p> <p>You can configure this by using the <code>ebxml.fileref.threshold</code> property from <b>BusinessConnect Container Edition &gt; System Settings &gt; Others &gt; Activate Protocol Plugins &gt; ebXML</b>.</p> <p><b>Note:</b> This field takes precedence over the <b>request</b> field.</p>
<b>Message Properties</b>	N/A	No	<p>Specific message properties that can be used for handling the messages efficiently.</p> <p>The property names must be agreed by trading partners.</p>
<b>PayloadInfo</b>	N/A	No	<p>Information of the payload of the request message. This is the payload encapsulated as the content of the SOAP Body.</p> <p>See <a href="#">PayloadInfo</a> for more information about the subfields.</p>
<b>attachment</b>	sequence of ebMS3/Info/Attachment	No	<p>A sequence of attachments received from the trading partner as the SOAP Attachments.</p> <p>See <a href="#">ae/BC/Attachment Fields</a> for formatting information.</p>
<b>agreementRef</b>	String	No	<p>The identifier of the collaboration protocol agreement, copied from the AgreementRef element in the ebMS3 messaging header of the inbound public message.</p>

Field	Type	Required	Description
<b>conversationID</b>	String	Yes	The identifier for the current conversation, copied from the ConversationId element in the SOAP header of the inbound public message.
<b>messageID</b>	String	Yes	The unique identifier for the inbound, request, copied from the MessageData/MessageId element in the SOAP header of the inbound public message.
<b>closure</b>	String	Yes	A unique string generated by BusinessConnect Container Edition - ebXML Protocol to keep track of this transaction. You must copy the value here to the <b>closure</b> field in the corresponding ae/ebMS3/ResponderResponse message.

## Outbound Response Format

The responder private process uses the following message class to send outbound response messages and business-level exceptions to BusinessConnect Container Edition - ebXML Protocol.



### Class

ae/ebMS3/ResponderResponse

## Subject name

Example: `AX.BC.ACME_SERVER.ebMS3.RESPONDER.RESPONSE`

`prefix.installation.RESPONDER.RESPONSE` (JMS queue)

## Message fields

See the following table for a description of each message field.

### ae/ebMS3/ResponderResponse Message Fields

Field	Type	Required	Description
<b>standardID</b>	String	Yes	The string "ebMS3".
<b>statusCode</b>	Integer	Yes	<p>The status of this transaction. Specify one of the following codes:</p> <ul style="list-style-type: none"> <li>• <b>200</b> The request is processed successfully.</li> <li>• <b>3652</b> The request is processed successfully but rejected. In this case, BusinessConnect Container Edition - ebXML Protocol sends a response message to the trading partner with <code>AcceptanceException</code> as the value of the <code>ExceptionType</code> element.</li> <li>• <b>3650</b> The request is not processed successfully. In this case, BusinessConnect Container Edition - ebXML Protocol sends a response</li> </ul>

Field	Type	Required	Description
			message to the trading partner with <code>GeneralException</code> as the value of the <code>ExceptionType</code> element.
<b>statusMsg</b>	String	Yes	A description of the code in the <b>statusCode</b> field. See <a href="#">Status Codes</a> .
<b>response</b>	String	No	<p>The response payload, which is usually an XML string. This is the payload to be encapsulated in the SOAP Body in the public response message.</p> <div style="border: 1px solid #ccc; padding: 10px; background-color: #f9f9f9;"> <p><b>Note:</b> Text-based payloads and attachments must be encoded in UTF-8. In TIBCO ActiveMatrix BusinessWorks™ Plug-in for BusinessConnect™, the <b>body</b> field is used to specify the content of the <b>response</b> field. Also the field is ignored when the <b>responseFile</b> field is not empty.</p> </div>
<b>responseFile</b>	String	No	The file reference used to specify the response payload file. You can use the file reference to send a large payload.

Field	Type	Required	Description
			<p><b>Note:</b> This field takes precedence over the <b>response</b> field.</p>
<b>Message Properties</b>	N/A	No	<p>Specific message properties that can be used for handling the messages efficiently.</p> <p>The property names must be agreed by trading partners.</p>
<b>PayloadInfo</b>	N/A	No	<p>Information of the payload of the response message, which is the content of the SOAP Body of the response message.</p> <p>See <a href="#">PayloadInfo</a> for more information about the subfields.</p>
<b>RSAKeyExchangeInfo</b>	sequence of ebMS3 //RSAKeyExchangeInfo	No	<p>An element that overrides the following algorithms from the private process:</p> <ol style="list-style-type: none"> <li><b>KeyTransportAlgorithm:</b> Supported Encryption algorithm applied to the encrypted data. It supports the following algorithms: <ul style="list-style-type: none"> <li>rsa-1_5</li> <li>rsa-oaep</li> <li>rsa-oaep-mgf1p</li> </ul> </li> <li><b>RSAES-OAEP-Params:</b> Supports two parameters</li> </ol>

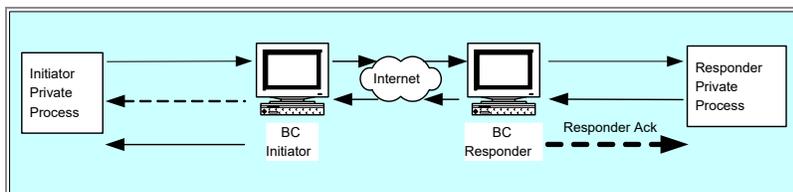
Field	Type	Required	Description
			<p>of the RSA OAEP algorithm in the private process:</p> <ul style="list-style-type: none"> <li>• MessageDigestAlgorithm (SHA-1, SHA-256, SHA-384, and SHA-512)</li> <li>• MaskGenerationAlgorithm (SHA-1, SHA-224, SHA-256, SHA-384, and SHA-512)</li> </ul> <div style="border: 1px solid #ccc; padding: 5px; background-color: #f9f9f9;"> <p><b>Note:</b> The type of the <b>KeyTransportAlgorithm</b> specified in the private process overrides the value of <code>ebms3.ob.keyExchange.algo</code> property defined in the ebXML Plug-in properties.</p> </div>
<b>attachment</b>	sequence of ebMS3/Info/Attachment	No	<p>A sequence of attachments to be sent out as SOAP Attachments in the public response message.</p> <p>See <a href="#">ae/BC/Attachment Fields</a> for formatting information.</p>
<b>messageID</b>	String	No	<p>A unique identifier for the outbound ebMS3 message. It must conform to the format described in RFC 2822. Otherwise, your trading partner cannot process the message</p>

Field	Type	Required	Description
			properly. If you do not supply a value here, it is automatically generated for the public message.
<b>closure</b>	String	No	A unique string for BusinessConnect Container Edition - ebXML Protocol to keep track of this transaction. You must copy the value from the <b>closure</b> field in the corresponding <code>ae/ebMS3/ResponderRequest</code> message.
<b>authorizationToken</b>	N/A	No	Not used for Responder .Response messages.
<b>authenticationToken</b>	N/A	No	Include username and password information that is used for WSS UsernameToken authentication, which is an alternative other than the WSS signature authentication.
<b>dupElimination</b>	Boolean	No	Request that the duplicate response messages sent by the private process are eliminated by the BusinessConnect Container Edition outbound process.
<b>trackingID</b>	String	No	An identifier that the private process can keep track of

Field	Type	Required	Description
			<p>messages that are otherwise related to each other. It is copied to the corresponding <code>ae/ebMS3/ResponderAck</code> message.</p> <p>Contrary to the <b>closure</b> and <b>conversationID</b> fields, there is no requirement on the value of this field, nor is it used to process public messages.</p>

## Response Acknowledgment Format

BusinessConnect Container Edition - ebXML Protocol uses the following message class to indicate to the private process whether the response message is sent to the trading partner successfully.



### Class

`ae/ebMS3/ResponderAck`

### Subject

Example: `AX.BC.ACME_SERVER.ebMS3.RESPONDER.ACK`

`prefix.installation.RESPONDER.ACK` (JMS queue)

### Message fields

See the following table for a description of each message field.

**ae/ebMS3/ResponderAck Message Fields**

	Type	Required	Description
<b>standardID</b>	String	Yes	The string "ebMS3".
<b>tradingPartner</b>	trading partner	Yes	The trading partner. See <a href="#">ae/BC/TradingPartner Fields</a> for formatting information.
<b>host</b>	trading partner	Yes	The trading host. See <a href="#">ae/BC/TradingPartner Fields</a> for formatting information.
<b>statusCode</b>	Integer	Yes	A code indicating the status of the transaction. See <a href="#">Status Codes</a> for more information.
<b>statusMsg</b>	String	Yes	A brief description of the code in the <b>statusCode</b> field.
<b>operationType</b>	String	No	The identifier of the transaction. Example: rosettanet.org/3A4/1.4/Purchase Order Request
<b>closure</b>	String	Yes	A unique string for BusinessConnect Container Edition - ebXML Protocol to keep track of this transaction. It is copied from the <b>closure</b> field in the corresponding ae/ebMS3/ResponderRequest message.
<b>operationID</b>	String	Yes	The identifier of the operation. Example: rosettanet.org/3A4/2.0/Request Purchase Order.
<b>organization</b>	String	Yes	The organization name for the transaction.
<b>businessProcess</b>	String	Yes	The name of the business process for the

	Type	Required	Description
			transaction.
<b>businessProcessVersion</b>	String	Yes	The business process version for the transaction.
<b>transactionName</b>	String	Yes	The name of the transaction.
<b>agreementRef</b>	String	No	The value of the AgreementRef element in the ebMS3 Messaging header of the public message. This is configured in the <b>AgreementRef</b> field of the Participants panel for the trading partner. See <a href="#">Setting Up Trading Hosts and Partners</a> .
<b>conversationID</b>	String	Yes	The value of the ConversationId element in the SOAP header of the public message.
<b>messageID</b>	String	No	The value of the MessageData/MessageId element in the SOAP header of the public message.
<b>trackingID</b>	String	No	The value of the <b>trackingID</b> field in the corresponding ae/ebMS3/ResponderResponse message, which can be used by the private process to keep track of messages that are otherwise related to each other.

## Advisory Signal Message Format

BusinessConnect Container Edition - ebXML Protocol uses the ae/ebMS3/Advisory class to update private processes on the status of ebMS3 transactions. BusinessConnect Container Edition - ebXML Protocol publishes ae/ebMS3/Advisory messages on the advisory signals subject when it receives signals from the trading partner.

See [Processing Signals](#).

The field names, which are not the same as column names, map to the column names in the audit database, and their values are the same as the values in the audit database.

**i Note:** The `ae/ebMS3/Advisory` message class is used for both advisory and error signals, but they are sent to separate private message subjects, and some of the message fields can hold different data for advisory and error signals.

## Class

`ae/ebMS3/Advisory`

## Subject

Example: `AX.BC.ACME_SERVER.ebMS3.ADVISORY.SIGNALS`

`prefix.installation.ADVISORY.SIGNALS` (JMS topic)

## Message fields

See the following table for description of each message field.

### **ae/ebMS3/Advisory Signal Fields**

	Type	Required	Description
<b>standardID</b>	String	Yes	The string "ebMS3".
<b>tpName</b>	String	No	The name of the trading partner in this transaction, as defined in the Partner Management panel in the BusinessConnect Container Edition console.
<b>tradingPartner</b>	trading partner	Yes	The trading partner. See <a href="#">ae/BC/TradingPartner Fields</a> for formatting information.
<b>host</b>	trading partner	Yes	The trading host. See <a href="#">ae/BC/TradingPartner Fields</a> for formatting information.

	Type	Required	Description
<b>operationID</b>	String	No	The identifier of the operation. For example: rosettnet.org/3A4/2.0/Request Purchase Order.
<b>transactionID</b>	String	No	The identifier of the transaction.
<b>statusCode</b>	Integer	No	<p>The status of this transaction. See the following possible codes:</p> <ul style="list-style-type: none"> <li>• <b>1651</b> Received a receipt acknowledgment from the trading partner.</li> <li>• <b>1653</b> Received an MSH receipt from the trading partner.</li> <li>• <b>3650</b> Received an exception message from the trading partner with an <code>ExceptionType/GeneralException</code> element in the message payload, indicating that the private process of the trading partner failed to process the request or response.</li> <li>• <b>3652</b> Received an exception message from the trading partner with an <code>ExceptionType/AcceptanceException</code> element in the message payload, indicating that the private process of the trading partner rejected the request or response.</li> </ul> <p>See <a href="#">Status Codes</a>.</p>
<b>statusMsg</b>	String	No	A brief description of the code in the <b>statusCode</b> field.
<b>details</b>	String	No	A more detailed description of the transaction status. This can be an analysis of

	Type	Required	Description
			the error that occurred.
<b>extraInfo</b>	String	No	The XML-formatted signal message that is received.
<b>closure</b>	String	No	A unique string for BusinessConnect Container Edition - ebXML Protocol to keep track of this transaction. It is copied from the <b>closure</b> field in the corresponding <code>ae/ebMS3/InitiatorRequest</code> or <code>ae/ebMS3/ResponderRequest</code> message.
<b>organization</b>	String	No	The organization name for the transaction.
<b>businessProcess</b>	String	No	The name of the business process for the transaction.
<b>businessProcess Version</b>	String	No	The business process version for the transaction.
<b>transactionName</b>	String	No	The name of the transaction.
<b>service</b>	String	No	The service that actions on the message, which is copied from the <code>Service</code> element in the SOAP header of the public message.
<b>serviceType</b>	String	No	The service type, which is copied from the <code>Service/type</code> attribute in the SOAP header of the public message.
<b>action</b>	String	No	The action used to process the inbound message, which is copied from the <code>Action</code> element in the SOAP header of the public message. If the value of the <code>Action</code> element is <code>Exception</code> , and then the value here is <code>GeneralException</code> or <code>AcceptanceException</code> , depending on the kind of exception.

	Type	Required	Description
<b>agreementRef</b>	String	No	The value of the AgreementRef element in the ebMS3 Messaging header of the public message. This is configured in the <b>AgreementRef</b> field of the Participants panel for the trading partner. See <a href="#">Setting Up Trading Hosts and Partners</a> .
<b>conversationID</b>	String	No	The value of the ConversationId element in the SOAP header of the public message.
<b>messageID</b>	String	No	The value of the MessageData/MessageId element in the SOAP header of the public message.
<b>trackingID</b>	String	No	The value of the <b>trackingID</b> field in the corresponding ae/ebMS3/InitiatorRequest or ae/ebMS3/ResponderResponse message, which can be used by the private process to keep track of messages that are otherwise related to each other.

## Error Message Format

BusinessConnect Container Edition - ebXML Protocol also uses the ae/ebMS3/Advisory class to send transaction errors to private processes.

The field names, which are not the same as column names, map to the column names in the audit database, and their values are the same as the values in the audit database.



**Note:** The ae/ebMS3/Advisory message class is used for both advisory and error signals, but they are sent to separate private message subjects, and some of the message fields can hold different data for advisory and error signals.

### Class

ae/ebMS3/Advisory

## Subject

Example: AX.BC.ACME\_SERVER.ebMS3.ERROR

*prefix.installation*.ERROR (JMS topic)

## Message fields

See the following table for a description of each message field.

### ae/ebMS3/Advisory Error Fields

	Type	Required	Description
<b>standardID</b>	String	Yes	The string "ebMS3".
<b>tpName</b>	String	No	The name of the trading partner in this transaction, as defined in the Partner Management panel in the BusinessConnect Container Edition console.
<b>tradingPartner</b>	trading partner	Yes	The trading partner. See <a href="#">ae/BC/TradingPartner Fields</a> for formatting information.
<b>host</b>	trading partner	Yes	The trading host. See <a href="#">ae/BC/TradingPartner Fields</a> for formatting information.
<b>transactionID</b>	String	No	The identifier of the transaction.
<b>operationID</b>	String	No	The identifier of the operation. Example: rosettanet.org/3A4/2.0/Request Purchase Order.
<b>statusCode</b>	Integer	No	A code indicating the status of the transaction. See <a href="#">Status Codes</a> .
<b>statusMsg</b>	String	No	A brief description of the code in the <b>statusCode</b> field.

	Type	Required	Description
<b>details</b>	String	No	A more detailed analysis of the error that occurred.
<b>extraInfo</b>	String	No	The XML-formatted signal message that is received.
<b>closure</b>	String	No	A unique string for BusinessConnect Container Edition - ebXML Protocol to keep track of this transaction. It is copied from the <b>closure</b> field in the corresponding <code>ae/ebMS3/InitiatorRequest</code> or <code>ae/ebMS3/ResponderRequest</code> message.
<b>organization</b>	String	No	The organization name for the transaction.
<b>businessProcess</b>	String	No	The name of the business process for the transaction.
<b>businessProcess Version</b>	String	No	The business process version for the transaction.
<b>transactionName</b>	String	No	The name of the transaction.
<b>service</b>	String	No	The service that actions on the message, which is copied from the <code>Service</code> element in the SOAP header of the public message.
<b>serviceType</b>	String	No	The service type, which is copied from the <code>Service/@type</code> attribute in the SOAP header of the inbound public message.
<b>action</b>	String	No	The action used to process the inbound message, which is copied from the <code>Action</code> element in the SOAP header of the public message. If the value of the <code>Action</code> element is <code>Exception</code> , and then the value

	Type	Required	Description
			here is GeneralException or AcceptanceException, depending on the kind of exception.
<b>agreementRef</b>	String	No	The value of the AgreementRef element in the ebMS3 Messaging header of the public message. This is configured in the <b>AgreementRef</b> field of the Participants panel for the trading partner. See <a href="#">Setting Up Trading Hosts and Partners</a> .
<b>conversationID</b>	String	No	The value of the ConversationId element in the SOAP header of the public message.
<b>messageID</b>	String	No	The value of the MessageData/MessageId element in the SOAP header of the public message.
<b>trackingID</b>	String	No	The value of the <b>trackingID</b> field in the corresponding ae/ebMS3/InitiatorRequest or ae/ebMS3/ResponderResponse message, which can be used by the private process to keep track of messages that are otherwise related to each other.

## Additional Data Classes

Two additional data classes are commonly used by some of the private process message classes:

- ae/BC/Attachment. See the following table for a description of each field.

**ae/BC/Attachment Fields**

Field	Type	Description
<b>Attachment</b>		
<b>name</b>	String	The name of the attached data.
<b>content-type</b>	String	The attached data type. Example: <code>application/binary</code> .
<b>content-id</b>	String	A unique identifier for the attachment.  For outbound attachments and also inbound attachments with an empty <b>name</b> field, BusinessConnect Container Edition - ebXML Protocol automatically generates a random string for the attachment content-id.
<b>content</b>	Binary	The content of the attachment. This field takes precedence over the <b>fileReference</b> field.  <b>Note:</b> All text-based attachments must be encoded in UTF-8.
<b>PartInfo</b>	N/A	The information of this particular attachment.
<b>fileReference</b>	String	The absolute path to the file that contains the attachment. This field is ignored if the <b>content</b> field is not empty.  For inbound messages, BusinessConnect Container Edition - ebXML Protocol uses this field to pass the reference of a large attachment. See <a href="#">Domains</a> .

- `ae/BC/TradingPartner`. See the following table for a description of each field.

**ae/BC/TradingPartner Fields**

Message Field	Type	Description
<b>TradingPartner</b>		

Message Field	Type	Description
<b>name</b>	String	The name of the trading participant.
<b>domain</b>	String	The domain of the trading participant.
<b>id</b>	String	The identity of the trading participant.

## Configuring Private Processes with TIBCO ActiveMatrix BusinessWorks

To define a TIBCO ActiveMatrix BusinessWorks private process to interact with BusinessConnect Container Edition requires the following resources:

- BusinessConnect Connection is a shared resource that connects to the BusinessConnect Container Edition configuration store for the settings in BusinessConnect Container Edition - ebXML Protocol.
- Activities of TIBCO ActiveMatrix BusinessWorks™ Plug-in for BusinessConnect™ are used to handle outbound and inbound BusinessConnect Container Edition messages.

### Configuring BCServerConfig Resource

For a private process to communicate with your BusinessConnect Container Edition server, you must include exactly one BCServerConfig resource in the root project view.

To configure the BCServerConfig resource, perform the following steps:

#### Procedure

1. Start TIBCO Business Studio.
2. Go to **File > Import**.
3. On the Import page, expand the **General** folder and select **Existing Studio Projects into Workspace**. Click **Next**.
4. Click **Browse** next to the **Select archive file** field to navigate to the `bcce-ebxml-`

1.0.0/samples/ebMS3/tutorial/bw directory, and select the ebMS3Demo\_for\_bw6.zip file. Click **Open** and then click **Finish**.

5. Expand **ebMS3 > Module Descriptors** in the Project Explorer view.
6. Double-click **Module Properties**.
7. Change the default value of the **BCHOME** property according to your environment.
8. In the Project Explorer view, expand **Resources** and double-click **BCConnection.bcResource**.
9. Click the **Server Access** tab.
10. Specify the values in the **JDBC Driver**, **JDBC URL**, **DB User**, and **DB Password** fields.

 **Note:** JDBC settings must be the same as the settings in your BusinessConnect Container Edition installation. To see the JDBC settings, review the deployment.properties file of BusinessConnect Container Edition.

11. Click the **Configuration** tab, and click **Update from Configuration Store**.
12. Select **ebMS3** from the **Protocol Name** list.
13. Click **Import Selected Business Protocol**.

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

14. Click **Save**.

## Configuring Send and Receive Activities

You can use BusinessConnect Container Edition Send activities for sending requests or receipts, responses, and miscellaneous messages. You can also use the BusinessConnect Container Edition Receive activities for receiving requests or receipts, response, and miscellaneous messages.

When configuring a Send activity, you must ensure that all required fields are supplied to the activity before the Send activity can be started. When a Send activity is started, a private process message of the selected type of the selected operation is sent to the BusinessConnect Container Edition Interior Server.

Unlike Send activities, Receive activities are *process starters*. Therefore, there can be only one Receive activity in a process and the Receive activity acts to start the process. A process with the Receive start activity is invoked when BusinessConnect Container Edition receives a message from a trading partner and sends it to TIBCO ActiveMatrix BusinessWorks, or when BusinessConnect Container Edition creates an Acknowledgment or Advisory message and sends it to the private process when certain event occurs. You can configure a process to use the output of the Receive activity to perform different tasks, such as formulating a response message and invoking a Send activity, or forwarding the received message to the downstream of your enterprise back-office system for further processing.

For more information about configuring each of the Send and Receive activities, see the *TIBCO ActiveMatrix BusinessWorks™ Plug-in for BusinessConnect™ documentation*. You can also work through the [Tutorial](#) to learn configuring a TIBCO ActiveMatrix BusinessWorks project to send and receive ebMS3 messages.

The following subsections give you a quick glance at the resources you must configure for different kinds of private processes.

## Configuring Initiators

For your TIBCO ActiveMatrix BusinessWorks processes to initiate ebMS3 transactions, use the following resources:

- BCServerConfig: for connecting to your BusinessConnect Container Edition server.
- Send Request/Notification: for sending request messages to trading partners.
- Receive Response: for receiving responses from trading partners. This is only for two-way transactions.
- Receive Misc. Msg: for receiving acknowledgments, advisory signals, or error signals from BusinessConnect Container Edition.

## Configuring Responders

For your TIBCO ActiveMatrix BusinessWorks processes to receive request messages, use the following resources:

- BCServerConfig: for connecting to your BusinessConnect Container Edition server.
- Receive Request/Notification: for receiving request messages from trading partners.
- Send Response: for sending response messages to BusinessConnect Container Edition

that forwards the response to the trading partner after processing and packaging. This is only for two-way transactions.

- Receive Misc. Msg: for receiving acknowledgments, advisory signals, or error signals from BusinessConnect Container Edition.

# Viewing Logs

---

This section explains how to view the transaction logs in Audit Trail of BusinessConnect Container Edition - ebXML Protocol.

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

To view the logs, on the Data Viewer tile, click **Audit Trail** > **ebMS3**.

All the transactions are displayed on [Transactions Screen](#).

## Transactions Screen

From the Transactions screen, you can do the following:

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

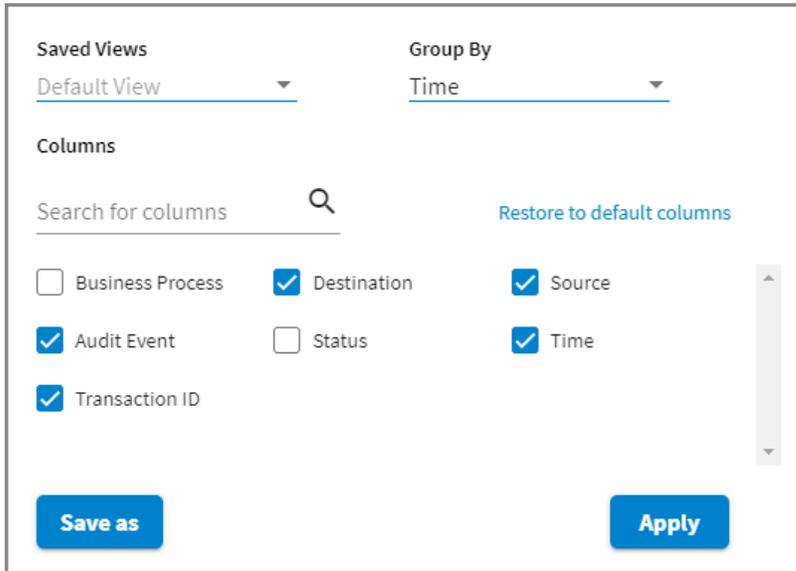
## Customizing Transactions Screen

### Procedure

1. Click the **Gear**  icon.
2. In the **Settings** dialog:
  - Select the checkboxes for the columns that you want to view on the Transaction screen.
  - Clear the checkboxes for the columns that you do not want to view on the Transaction screen.

3. Click **Apply** to save your transaction screen settings.

Figure 11: Settings



4. You can also use the following options:

Option	Description
<b>Group By</b>	Use this option to group results by category. To do this, select a category from the dropdown list.
<b>Columns</b>	Check the box next to the category to display the columns you want to view. You can also search for a category by entering a partial word to display all categories containing that string as shown in the following example.

**Columns**

Search for columns

Num    [Restore to default columns](#)

CO Number     DO Number     Invoice Number

PO Number     Tracking Number

5. Each page displays up to 20 records with the newest transactions listed first as default. Page through the display to see the older records. You can sort the

transactions by category, or search for records of a specific type. The number of records returned are limited to 10,000 audit events or a maximum of 500 pages.

**i Note:** A "No records found" error is generated if your selection exceeds 500 pages.

## Sort Transactions

Click the arrows on a column (category) heading to sort transactions.

- ↑ : Sort first to last.
- ↓ : Sort last to first.

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

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

## Search for Transactions

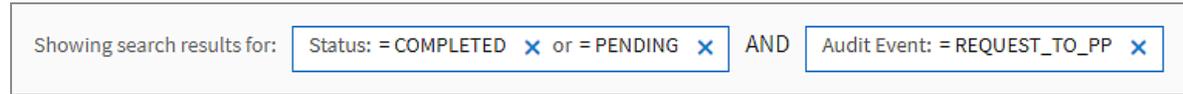
The search feature of TIBCO AuditSafe is powerful and easy-to-use.

You can search for values in multiple categories and find matching records even if a category (or column) is not displayed.

While searching for values, use the following options:

- **Keyword:** To search for a term in all categories.
- **Audit Event:** To search for an audit event in all categories.
- **OR:** To search for values or keywords within the same category.  
Example: Search **Status** for in PROCESSING **OR** COMPLETED to find all the transactions that contain those values.
- **AND:** To search for values between two or more categories.  
Example: Search **Status** for COMPLETED **OR** PENDING, **AND** search **Audit Event** for Request\_TO\_PP.

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



## Using Search

Search contains the following options:

- **Time:** To define the start and end range for your search.
- **Keyword:** To search for the term in all categories.

Search Option	Description
=	<b>Equals:</b> Finds an exact match for a term.  Entering the first few letters 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 Transaction** field, select a [Search Option](#) , and enter a word.
3. Select an option from the ones that are displayed when you enter a word, and add it to your query.
4. Continue adding options to narrow your search results.
5. After creating your query, click **Find** to list the results.
6. Click the **Remove** icon  to remove a search term.

## Transaction Details

Click a transaction in the list to see more details.

You can view the transactions using either a Diagram view or a List view.

Switch between the two views by clicking the icon on the top right.

The diagram view also displays related transactions.

# Tutorial

---

This tutorial guides you through the steps necessary for running the samples of BusinessConnect Container Edition - ebXML Protocol ebMS3/AS4 standard.

BusinessConnect Container Edition - ebXML Protocol provides the following sample projects. These sample projects contain processes of different transactions, and you can learn how to do transactions of various MEPs.

- ebMS3Demo\_for\_bw6.zip

You can find this file in the `bcce-ebxml-1.0.0\samples\ebMS3\tutorial\bw` directory. The `ebMS3Demo_for_bw6.zip` file is used with TIBCO ActiveMatrix BusinessWorks 6.x

- ebms3\_sample\_operations.csx

You can find this file in the `bcce-ebxml-1.0.0\samples\ebMS3\tutorial\interfaces` directory. This sample file contains operations used for BusinessConnect Container Edition - ebXML Protocol, and you can import the file into BusinessConnect Container Edition.

## Prerequisites

Before starting the tutorial, install the following TIBCO software packages:

1. TIBCO BusinessConnect™ Container Edition

For more information on setting up and running TIBCO BusinessConnect™ Container Edition, see the *TIBCO BusinessConnect™ Container Edition Administration* and *TIBCO BusinessConnect™ Container Edition Trading Partner Management* guides.

2. TIBCO ActiveMatrix BusinessWorks.
3. TIBCO ActiveMatrix BusinessWorks™ Plug-in for BusinessConnect™
4. TIBCO BusinessConnect™ Container Edition - ebXML Protocol

# Setting Up the Tutorial

The section describes how to configure the tutorial in BusinessConnect Container Edition.

Four samples are presented in this tutorial: one-way push transaction, one-way pull transaction, two-way asynchronous/synchronous transactions, and two-way push and pull transaction.

## Participant Information

See the following table for the participant information used in this tutorial.

	Machine 1	Machine 2
Trading Partner Name	Buyer	Seller
Key Identity File	bcpartner1_key.p12	bcpartner2_key.p12
Certificate file	bcpartner1_cert.p7b	bcpartner2_cert.p7b

## Setting Up Buyer on Machine 1

This section provides the instructions for setting up the BusinessConnect Container Edition server on Machine 1, which acts as the buyer in the transaction. The following tasks include:

- [Importing ebMS3 Operations](#)
- [Configuring Server Transport](#)
- [Configuring Trading Hosts](#)
- [Configuring Trading Partners](#)
- [Configuring Business Agreement](#)
- [Others](#)

## Importing ebMS3 Operations

To import the ebMS3 operations that are used in the tutorial, perform the following steps:

## Procedure

1. On Machine 1, log in to BusinessConnect Container Edition.
2. On the **B2B Administration** tile, click **Operations Editor**.
3. Click the **Import** icon.
4. On the Import Operations dialog:
  - a. Click **Upload file** and navigate to the `bcce-ebxml-1.0.0\samples\ebMS3\tutorial\interfaces` directory.
  - b. Select the `ebms3_sample_operations.csx` file, and then click **Open**.
  - c. Leave the **Password** field empty.
  - d. Click **Import**.

## Configuring Server Transport

To configure the BusinessConnect Container Edition server on Machine 1, perform the following steps:

## Procedure

1. Go to **System Settings > Transport Protocols**.
2. Click **Inbound Protocols**.
3. Select the **HTTP**, **HTTPS**, and **HTTPSCA** checkboxes.
4. Click **Save**.
5. Click **Configure Service** next to the **HTTP**, **HTTPS**, and **HTTPSCA** checkboxes.
6. Click the **Add** icon .
7. Enter **HTTP** in the **Gateway Service Name** field, and select **HTTP** from the **Gateway Service Type** list.
8. Click **Proceed**.
9. Click the **Credentials** tab.
10. Click **New Credential**.
11. Enter **Buyer Key** in the **Alias** field.
12. Click **Upload file** and navigate to the `bcce-<version>/images/samples/keys`

directory.

13. Select **bcpartner1\_key.p12** and click **Open**.
14. Enter **Password1** in the **Password** field.
15. Click the **Ports** tab.
16. Select **Buyer Key** from the **Private Key Credential for Secure Ports** list.
17. Click **Save** and then click **Add**.

## Configuring Trading Hosts

Set up the buyer as a trading host, which consists of the following steps:

- [Setting Up the Trading Host Properties](#)
- [Setting Up ebMS3 Protocol](#)

## Setting Up the Trading Host Properties

To set up the trading host properties, perform the following steps:

### Procedure

1. On the **Partner Management** tile, click **Hosts**.
2. Click **Create Host**, which is on the upper-right corner of the page.
3. Enter **Buyer** in the **Host Name** field.
4. Click **Proceed**.

 **Note:** Steps 5 to 14 are optional. Do the steps only if you want to add information related to business location and contact person.

5. Go to the **Business Locations** tab and click **Add Location**.
6. In the **Location Name** field, enter **Headquarters**.
7. In the Add Location dialog, specify the following details:
  - **Email:** **buyer@buyer.com**
  - **Phone:** **111-111-1111**

- **Fax: 111-111-1111**

8. Click the **Add** icon  next to the **Primary Legal Contact** field.

In the New Contact panel, specify the following details:

- **Contact Type: Legal**
- **First Name: Joe**
- **Last Name: Smith**

9. Click **Add**.

10. Go to the **Details** tab and select **Joe Smith** from the **Primary Legal Contact** list.

11. Click the **Add** icon  next to the **Primary Support Contact** field.

In the New Contact panel, specify the following details:

- **Contact Type: Support**
- **First Name: Jane**
- **Last Name: Jones**

12. Click **Add**.

13. Go to the **Details** tab and select **Jane Jones** from the **Primary Support Contact** list.

14. Click **Save**.

15. Go to the **Credentials** tab, click **New Private Key**.

16. Enter **Buyer Key** in the **Alias** field.

17. Click the **Upload file** and navigate to the `bcce-<version>/images/samples/keys` directory.

18. Select `bcpartner1_key.p12` and click **Open**.

19. Enter **Password1** in the **Password** field.

20. Click **Add**.

## Setting Up ebMS3 Protocol

To set up the **Protocols** tab of a trading host, perform the following steps:

### Procedure

1. Go to the **Protocols** tab.
2. Click **Edit Configurations** next to the **ebMS3** checkbox.
3. Configure the **General** tab:
  - a. Click the **Add** icon  next to the **Default Domain Identity** field.
  - b. On the **Add New** tab:
    - i. Select **URI** from the **Domain Type** list.
    - ii. Enter **uri:buy123** in the **ID** field.
    - iii. Click **Add**.
  - c. Select **URI-uri:buy123** from the **Default Domain Identity** list.
4. Click **Save**.

## Configuring Trading Partners

Set up a trading partner configuration on Machine 1 for the seller, which consists of the following steps:

- [Setting Up the Trading Partner Properties](#)
- [Setting Up ebMS3 Protocol](#)

## Setting Up the Trading Partner Properties

To set up the trading partner properties, perform the following steps:

### Procedure

1. On the **Partner Management** tile, click **Partners**.
2. Click **Create Partner**, which is on the upper-right corner of the page.
3. Enter **seller** in the **Partner Name** field.
4. Click **Proceed**.

 **Note:** Steps 5 to 14 are optional. Do the steps only if you want to add information related to business location and contact person.

5. Go to the **Business Locations** tab and click **Add Location**.
6. In the **Location Name** field, enter **Headquarters**.
7. In the Add Location dialog, specify the following details:
  - **Email:** `seller@seller.com`
  - **Phone:** `222-222-2222`
  - **Fax:** `222-222-2222`
8. Click the **Add** icon  next to the **Primary Legal Contact** field.
9. In the New Contact panel, specify the following details:
  - **Contact Type:** **Legal**
  - **First Name:** **Jack**
  - **Last Name:** **Smith**
10. Click **Add**.
11. Go to the **Details** tab and select **Jack Smith** from the **Primary Legal Contact** list.
12. Click the **Add** icon  next to the **Primary Support Contact** field.

In the New Contact panel, specify the following details:

  - **Contact Type:** **Support**
  - **First Name:** **Jill**
  - **Last Name:** **Jones**
13. Click **Add**.
14. Go to the **Details** tab and select **Jill Jones** from the **Primary Support Contact** list.
15. Click **Save**.
16. Go to the **Credentials** tab, click **New Certificate**.
17. Enter **Seller Cert** in the **Alias** field.
18. Click the **Upload file** and navigate to the `bcce-<version>/images/samples/keys` directory.
19. Select `bcpartner2_cert.p7b` and click **Open**.

20. Click **Add**.

## Setting Up ebMS3 Protocol

To set up the **Protocols** tab of a trading partner, perform the following steps:

### Procedure

1. Go to the **Protocols** tab.
2. If ebMS3 is not selected then select the **ebMS3** checkbox.
3. Click **Edit Configurations** next to the **ebMS3** checkbox.
4. Configure the **General** tab:
  - a. Click the **Add** icon  next to the **Default Domain Identity** field.
  - b. Select **URI** from the **Domain Type** list.
  - c. Enter **uri:sell123** in the **ID** field.
  - d. Click **Add**.
  - e. Select **URI-uri:sell123** from the **Default Domain Identity** list.
  - f. Enter **ebms3** in the **AgreementRef** field.
5. Go to the **Transport** tab and click **Add Outbound Transport**.
6. On the **Add Transport** dialog:
  - a. Enter **HTTP** in the **Transport Name** field.
  - b. Select **HTTP** from the **Transport Type** list.
  - c. In the **URL** field, enter *hostname*:**6700/dmz/ebMS3**, where *hostname* is the name of Machine 2, which configures the seller as its trading host, and **6700** is the HTTP port set in the deployment configuration on the Machine 2 BusinessConnect Container Edition server. See [Configuring Server Transport](#).
7. Click **Add** to finish configuring this trading partner.

## Configuring Business Agreement

To set up an ebMS3 agreement protocol binding, perform the following steps:

## Procedure

1. On the **Partner Management** tile, click **Business Agreements**.
2. Click **Create Agreement**, which is in the upper-right corner of the page.
3. Select **Buyer** in the **Host** field and **Seller** in the **Partner** field.
4. Click **Proceed**.
5. On the **Bind Protocol** tab, select the **ebMS3** checkbox.
6. Click **Edit Configurations** next to the **ebMS3** checkbox.
7. On the **Operation** tab, select the **Allow All Operations** checkbox. You can use the selected host and partner to initiate all ebMS3 transactions installed in the Operations Editor.
8. Click the **Document Exchange** tab.
9. Select **Buyer Key** from the **Signing Key** list.
10. Select **Seller Cert** from the **Encryption Certificate** list.
11. Select **Seller Cert** from the **Verification Certificate** list.
12. Select **Buyer Key** from the **Decryption Key** list.
13. Click the **Transports** tab.
14. In the **Outbound Transports for Host "Buyer"** section, select **HTTP** from the **Primary Transport** list.
15. In the **Inbound Transports for Partner "Seller"** section, make sure that the **HTTP** checkbox is selected.
16. Click **Save** to finish configuring this business agreement.

## Others

Perform the following tasks:

1. Deploy BusinessConnect Container Edition.
2. Start BusinessConnect Container Edition Interior Server and Gateway Server.

For more information, see *TIBCO BusinessConnect™ Container Edition Administration* and *TIBCO BusinessConnect™ Container Edition Installation and Deployment*.

## Setting Up Seller on Machine 2

This section provides the instructions for setting up the BusinessConnect Container Edition server on Machine 2, which acts as the seller in the transaction. The following tasks include:

- [Importing ebMS3 Operations](#)
- [Configuring Server Transport](#)
- [Configuring Trading Hosts](#)
- [Configuring Trading Partners](#)
- [Configuring Business Agreement](#)
- [Configuring External Users](#)
- [Others](#)

### Importing ebMS3 Operations

To import the ebMS3 operations that are used in the transaction, perform the following steps:

#### Procedure

1. On Machine 2, log in to BusinessConnect Container Edition.
2. On the **B2B Administration** tile, click **Operations Editor**.
3. Click the **Import** icon.
4. On the Import Operations dialog:
  - a. Click **Upload file** and navigate to the `bcce-ebxml-1.0.0\samples\ebMS3\tutorial\interfaces` directory.
  - b. Select the `ebms3_sample_operations.csx` file, and then click **Open**.
  - c. Leave the **Password** field empty.
  - d. Click **Import**.

## Configuring Server Transport

To configure the BusinessConnect Container Edition server on Machine 2, perform the following steps:

### Procedure

1. Go to **System Settings > Transport Protocols**.
2. Click **Inbound Protocols**.
3. Select the **HTTP**, **HTTPS**, and **HTTPSCA** checkboxes.
4. Click **Save**.
5. Click **Configure Service** next to the **HTTP**, **HTTPS**, and **HTTPSCA** checkboxes.
6. Click the **Add** icon  .
7. Enter **HTTP** in the **Gateway Service Name** field, and select **HTTP** from the **Gateway Service Type** list.
8. Click **Proceed**.
9. Click the **Credentials** tab.
10. Click **New Credential**.
11. Enter **Seller Key** in the **Alias** field.
12. Click **Upload file** and navigate to the `bcce-<version>/images/samples/keys` directory.
13. Select `bcpartner2_key.p12` and click **Open**.
14. Enter **Password1** in the **Password** field.
15. Click the **Ports** tab.
16. Select **Seller Key** from the **Private Key Credential for Secure Ports** list.
17. Click **Save** and then click **Add**.

## Configuring Trading Hosts

Set up the seller as a trading host, which consists of the following steps:

- [Setting Up the Trading Host Properties](#)

- [Setting Up ebMS3 Protocol](#)

## Setting Up the Trading Host Properties

To set up the trading host properties, perform the following steps:

### Procedure

1. On the **Partner Management** tile, click **Hosts**.
2. Click **Create Host**, which is on the upper-right corner of the page.
3. Enter **seller** in the **Host Name** field.
4. Click **Proceed**.

**i Note:** Steps 5 to 14 are optional. Do the steps only if you want to add information related to business location and contact person.

5. Go to the **Business Locations** tab and click **Add Location**.
6. In the **Location Name** field, enter **Headquarters**.
7. In the Add Location dialog, specify the following details:
  - **Email:** **seller@seller.com**
  - **Phone:** **222-222-2222**
  - **Fax:** **222-222-2222**
8. Click the **Add** icon  next to the **Primary Legal Contact** field.  
In the New Contact panel, specify the following details:
  - **Contact Type:** **Legal**
  - **First Name:** **Jack**
  - **Last Name:** **Smith**
9. Click **Add**.
10. Go to the **Details** tab and select **Jack Smith** from the **Primary Legal Contact** list.
11. Click the **Add** icon  next to the **Primary Support Contact** field.

In the New Contact panel, specify the following details:

- **Contact Type: Support**
  - **First Name: Jill**
  - **Last Name: Jones**
12. Click **Add**.
  13. Go to the **Details** tab and select **Jill Jones** from the **Primary Support Contact** list.
  14. Click **Save**.
  15. Go to the **Credentials** tab, click **New Private Key**.
  16. Enter **Seller Key** in the **Alias** field.
  17. Click the **Upload file** and navigate to the `bcce-<version>/images/samples/keys` directory.
  18. Select **bcpartner2\_key.p12** and click **Open**.
  19. Enter **Password1** in the **Password** field.
  20. Click **OK**.
  21. Click **Add**.

## Setting Up ebMS3 Protocol

To set up the **Protocols** tab of the trading host, perform the following steps:

### Procedure

1. Go to the **Protocols** tab.
2. Click **Edit Configurations** next to the **ebMS3** checkbox.
3. Configure the **General** tab:
  - a. Click the **Add** icon  next to the **Default Domain Identity** field.
  - b. On the **Add New** tab:
    - i. Select **URI** from the **Domain Type** list.
    - ii. Enter **uri:sell123** in the **ID** field.
    - iii. Click **Add**.
  - c. Select **URI-uri:sell123** from the **Default Domain Identity** list.

4. Click **Save**.

## Configuring Trading Partners

Set up a trading partner configuration on Machine 2 for the buyer, which consists of the following steps:

- [Setting Up the Trading Partner Properties](#)
- [Setting Up ebMS3 Protocol](#)

## Setting Up the Trading Partner Properties

To set up the trading partner properties, perform the following steps:

### Procedure

1. On the **Partner Management** tile, click **Partners**.
2. Click **Create Partner**, which is on the upper-right corner of the page.
3. Enter **Buyer** in the **Partner Name** field.
4. Click **Proceed**.

 **Note:** Steps 5 to 14 are optional. Do the steps only if you want to add information related to business location and contact person.

5. Go to the **Business Locations** tab and click **Add Location**.
6. In the **Location Name** field, enter **Headquarters**.
7. In the Add Location dialog, specify the following details:
  - **Email:** **buyer@buyer.com**
  - **Phone:** **111-111-1111**
  - **Fax:** **111-111-1111**
8. Click the **Add** icon  next to the **Primary Legal Contact** field.
9. In the New Contact panel, specify the following details:
  - **Contact Type:** **Legal**

- **First Name: Joe**
  - **Last Name: Smith**
10. Click **Add**.
  11. Go to the **Details** tab and select **Joe Smith** from the **Primary Legal Contact** list.
  12. Click the **Add** icon  next to the **Primary Support Contact** field.

In the New Contact panel, specify the following details:

- **Contact Type: Support**
  - **First Name: Jane**
  - **Last Name: Jones**
13. Click **Add**.
  14. Go to the **Details** tab and select **Jane Jones** from the **Primary Support Contact** list.
  15. Click **Save**.
  16. Go to the **Credentials** tab, click **New Certificate**.
  17. Enter **Buyer Cert** in the **Alias** field.
  18. Click the **Upload file** and navigate to the `bcce-<version>/images/samples/keys` directory.
  19. Select `bcpartner1_cert.p7b` and click **Open**.
  20. Click **Add**.

## Setting Up ebMS3 Protocol

To set up the **Protocols** tab of a trading partner, perform the following steps:

### Procedure

1. Go to the **Protocols** tab.
2. If ebMS3 is not selected then select the **ebMS3** checkbox.
3. Click **Edit Configurations** next to the **ebMS3** checkbox.
4. Configure the **General** tab:

- a. Click the **Add** icon  next to the **Default Domain Identity** field.
  - b. Select **URI** from the **Domain Type** list.
  - c. Enter **uri:buy123** in the **ID** field.
  - d. Click **Add**.
  - e. Select **URI-uri:buy123** from the **Default Domain Identity** list.
  - f. Enter **ebms3** in the **AgreementRef** field.
5. Go to the **Transport** tab and click **Add Outbound Transport**.
  6. On the **Add Transport** dialog:
    - a. Enter **HTTP** in the **Transport Name** field.
    - b. Select **HTTP** from the **Transport Type** list.
    - c. In the **URL** field, enter *hostname*:**6700/dmz/ebMS3**, where *hostname* is the name of Machine 1, which configures the buyer as its trading host, and **6700** is the HTTP port set in the deployment configuration on the Machine 1 BusinessConnect Container Edition server. See [Configuring Server Transport](#).
  7. Click **Add** to finish configuring this trading partner.

## Configuring Business Agreement

To set up an ebMS3 agreement protocol binding, perform the following steps:

### Procedure

1. On the **Partner Management** tile, click **Business Agreements**.
2. Click **Create Agreement**, which is on the upper-right corner of the page.
3. Select **Seller** in the **Host** field and **Buyer** in the **Partner** field.
4. Click **Proceed**.
5. On the **Bind Protocol** tab, select the **ebMS3** checkbox.
6. Click **Edit Configurations** next to the **ebMS3** checkbox.
7. On the **Operation** tab, select the **Allow All Operations** checkbox. You can use the selected host and partner to initiate all ebMS3 transactions installed in the Operations Editor.

8. Click the **Document Exchange** tab.
9. Select **Seller Key** from the **Signing Key** list.
10. Select **Buyer Cert** from the **Encryption Certificate** list.
11. Select **Buyer Cert** from the **Verification Certificate** list.
12. Select **Seller Key** from the **Decryption Key** list.
13. Click the **Transports** tab.
14. In the **Outbound Transports for Host "Seller"** section, select **HTTP** from the **Primary Transport** list.
15. In the **Inbound Transports for Partner "Buyer"** section, make sure that the **HTTP** checkbox is selected.
16. Click **Save** to finish configuring this business agreement.

## Configuring External Users

 **Note:** This task is only used for the one-way pull, send one-way push transaction with username authentication, and two-way push and pull transaction tutorials.

To set up an external user, perform the following steps:

### Procedure

1. On the **User Management** tile, click **Users**.
2. Go to the **External** tab and click **Create User**, which is on the upper-right corner of the page.
3. From the **Partner** dropdown list, select **Buyer**.
4. In the **Email** and **Confirm Email** fields, enter **exuser2@tibco.com**.
5. In the **Password** field, enter **exuser2**.
6. Click **Save**.

## Others

Perform the following tasks:

1. Deploy BusinessConnect Container Edition.
2. Start BusinessConnect Container Edition Interior Server and Gateway Server.

For more information, see *TIBCO BusinessConnect™ Container Edition Administration* and *TIBCO BusinessConnect™ Container Edition Installation and Deployment*.

## Configuring Private Processes in TIBCO Business Studio

To configure the tutorial in TIBCO Business Studio, complete the following tasks:

### Opening the TIBCO ActiveMatrix BusinessWorks Project

To set up the example project on the initiator and the responder machines, perform the following steps:

1. Start TIBCO Business Studio.
2. Go to **File > Import**.
3. On the Import page, expand the **General** folder and select **Existing Studio Projects into Workspace**. Click **Next**.
4. Click **Browse** next to the **Select archive file** field to navigate to the `bcce-ebxml-1.0.0/samples/ebMS3/tutorial/bw` directory, and select the `ebMS3Demo_for_bw6.zip` file. Click **Open** and then click **Finish**.
5. Expand **ebMS3 > Module Descriptors** in the Project Explorer view.
6. Double-click **Module Properties**.
7. Change the default value of the **BCHOME** property according to your environment.

### Configuring Connections to BusinessConnect Container Edition

To configure connections to BusinessConnect Container Edition, perform the following steps:

1. In the Project Explorer view, expand **Resources** and double-click

**BCConnection.bcResource.**

2. Click the **Server Access** tab.
3. Specify the values in the **JDBC Driver**, **JDBC URL**, **DB User**, and **DB Password** fields.

**i Note:** JDBC settings must be the same as the settings in your BusinessConnect Container Edition installation. To see the JDBC settings, review the `deployment.properties` file of BusinessConnect Container Edition.

4. Click the **Configuration** tab, and click **Update from Configuration Store**.
5. Select **ebMS3** from the **Protocol Name** list.  
If you select the **Select Operations** checkbox, you can select any of the configured and imported operations. For this tutorial, select all operations and click **OK**.
6. Click **Import Selected Business Protocol**.  
When you import the protocol, TIBCO ActiveMatrix BusinessWorks retrieves information from the BusinessConnect Container Edition configuration store and puts it in the project folder.
7. Click **Save**.

## Running the Tutorial

The following sample transactions are described in this tutorial:

- [One-Way Push Transactions](#)
- [One-Way Pull Transactions](#)
- [Two-Way Asynchronous/Synchronous Transactions](#)
- [Two-Way Push and Pull Transactions](#)

## Creating a Process Invoker

### Procedure

1. In the Project Explorer view, expand **ebXMLDemo > Processes**.

2. Right-click on Processes, from the context menu, select **New > BusinessWorks Process**.
3. In the BusinessWorks Process Creation dialog, in the **Process Name** field, enter InvokerProcess and click **Finish**.
4. From the **Palette Library > General Activities**, add the **Timer** activity to the process.
5. From the **Palette Library > Basic Activities**, add the **Invoke** activity to the process.
6. Drag the  icon to create a transition between the added activities.
7. Select the Invoke activity on the **Properties** tab, click the  icon next to the **Service** field.
8. From the Select a Service dialog, select the process that you want to invoke and click **OK**.

## One-Way Push Transactions

This section describes the processes defined for one-way push transactions.

This example contains the following processes in TIBCO ActiveMatrix BusinessWorks:

- [Send One Way Push Message](#) and [Send One-Way Push Message with Username Authentication](#)
- [Receive Pushed Message](#)

### Send One Way Push Message

Use the **Send One Way Push Message** process on Machine 1 to initiate a one-way push transaction. You can access this process from the TIBCO ActiveMatrix BusinessWorks Project panel in the **Initiator** folder. The private process reads a request from a local directory and sends the document to the BusinessConnect Container Edition server.

### Send One-Way Push Message with Username Authentication

Use the **Send One-Way Push Message with Username Authentication** process on Machine 1 to initiate a one-way push message with a username authentication transaction. You can access this process from the TIBCO ActiveMatrix BusinessWorks Project panel in

the **Initiator** folder. The private process reads a request from a local directory and sends the document to the BusinessConnect Container Edition server.

## Receive Pushed Message

Use the **Receive Pushed Message** process on Machine 2 to respond to the push request message. You can access this process from the TIBCO ActiveMatrix BusinessWorks Project panel in the **Responder** folder.



## Running One-Way Push Transaction

To run this example, complete the following tasks:

### Running Receive Push Message on Machine 2

Follow these instructions to start the **Receive Push Message** process on Machine 2:

1. In the opened TIBCO ActiveMatrix BusinessWorks project on Machine 2, expand the **Responder** folder.
2. Click the **Receive Pushed Message** process in the Project Explorer.
3. Click the **Receive Request** activity.
4. On the **General** tab, select **rosettanet.org/3A7/1.3/Notify Of Purchase Order Acceptance** from the **Operation Name** list.
5. Click **Apply** to save the changes.
6. On the **Tester** tab, click the dropdown icon next to the **BWApplication** icon .
7. Select **Run Configuration** from the dropdown list.

**i Note:** If you want to run the process in the debug mode, click **Debug BWApplication** icon > **Debug Configurations**.

8. In the **Run Configurations** dialog, expand **BusinessWorks Application** > **BWApplication**.
9. From the list of the processes displayed, you can select one or multiple items. In this tutorial, select **Responder** > **Receive Pushed Message**.
10. Click **Apply** > **Run**.

You can see BusinessWorks logs on the **Console** tab.

## Running Send One Way Push Message with or without Username Authentication on Machine 1

You must start the **Receive Pushed Message** process on Machine 2 before proceeding with this section. Follow these instructions to start the **Send One Way Push Message** or the **Send One-Way Push Message with Username Authentication** process on Machine 1:

1. In the open TIBCO ActiveMatrix BusinessWorks project on Machine 1, expand the **Initiator** folder.
2. Click the **Send One Way Push Message** or the **Send One-Way Push Message with Username Authentication** process in Project Explorer.
3. Click the **Send Purchase Order Acceptance** activity.
4. On the **General** tab, select **rosettanet.org/3A7/1.3/Notify Of Purchase Order Acceptance** from the **Operation Name** list.
5. On the **Input** tab, enter **seller** in the **tpName** field.
6. On the **Module Properties** tab, the username and password defined must match the ones you configured on Machine 2 in BusinessConnect Container Edition when you define the external user and associate with the trading partner Buyer. If you follow the instructions, the username is `exuser2@tibco.com`. See [Configuring External Users](#).
7. Click **Apply** to save the changes.
8. On the **Tester** tab, click the dropdown icon next to the **BWApplication** icon .
9. Select **Run Configuration** from the dropdown list.

**i** **Note:** If you want to run the process in the debug mode, click **Debug BWAApplication** icon > **Debug Configurations**.

10. In the **Run Configurations** dialog, expand **BusinessWorks Application** > **BWAApplication**.
11. From the list of the processes displayed, you can select one or multiple items. In this tutorial, select **Initiator** > **Send One Way Push Message** or **Send One-Way Push Message with Username Authentication**.
12. Click **Apply** > **Run**.

You can see BusinessWorks logs on the **Console** tab.

## Expected Results

If everything works as expected, the one-way push request is sent to the trading partner and then the process ends on Machine 1.

Machine 2 receives a push request from the trading partner. The process ends.

## One-Way Pull Transactions

This section describes the processes defined for one-way pull transactions.

This example contains the following processes in TIBCO ActiveMatrix BusinessWorks:

- [Send UserMessage to be Pulled](#)
- [Send One Way Pull Request](#)

### Send UserMessage to be Pulled

Use the **Send UserMessage to be Pulled** process to send a user message to the BusinessConnect Container Edition server on Machine 2. You can access this process from the TIBCO ActiveMatrix BusinessWorks Project panel in the **Initiator** folder.



## Send One Way Pull Request

Use the **Send One Way Pull Request** process on Machine 1 to send a pull request to the trading partner. You can access this process from the TIBCO ActiveMatrix BusinessWorks Project panel in the **Initiator** folder.

## Running One-Way Pull Transaction

To run this example, complete the following tasks:

## Running SendUserMessage to be Pulled on Machine 2

Follow these instructions to start the **SendUserMessage to be Pulled** process on Machine 2:

1. In the opened TIBCO ActiveMatrix BusinessWorks project on Machine 2, expand the **Initiator** folder.
2. Click the **Send UserMessage to be Pulled** process in Project Explorer.
3. Click the **Send UserMessage for Pulling** activity.
4. On the **General** tab, select **rosettnet.org/3A7/1.3/Notify Of Purchase Order Acceptance** from the **Operation Name** list.
5. On the **Input** tab:
  - Set the `tpName` attribute to Buyer
  - Set the `toBePulled` attribute to true
  - Provide a valid URI in the **mpc** field
  - Provide the username for this message

The username provided must match the external user that you configured in

BusinessConnect Container Edition on Machine 2 and associated with the trading partner Buyer. If you follow the instructions, the username is exuser2@tibco.com. See [Configuring External Users](#).

Therefore, the message is saved into Store-and-Forward on Machine 2, that is targeted to be pulled by the trading partner Buyer from this MPC channel.

6. Click **Apply** to save the changes.
7. On the **Tester** tab, click the dropdown icon next to the **BWApplication** icon .
8. Select **Run Configuration** from the dropdown list.

**i Note:** If you want to run the process in the debug mode, click **Debug BWApplication** icon > **Debug Configurations**.

9. In the **Run Configurations** dialog, expand **BusinessWorks Application** > **BWApplication**.
10. From the list of the processes displayed, you can select one or multiple items. In this tutorial, select **Initiator** > **Send UserMessage to be Pulled**.
11. Click **Apply** > **Run**. You can see BusinessWorks logs on the **Console** tab.

## Running Send One Way Pull Request on Machine 1

Follow these instructions to start the **Send One Way Pull Request** process on Machine 1:

1. In the open TIBCO ActiveMatrix BusinessWorks project on Machine 1, expand the **Initiator** folder.
2. Click the **Send One Way Pull Request** process in Project Explorer.
3. Click the **Send Pull Request** activity.
4. On the **General** tab, select **tibco.com/EBMS Pull Request/1.0: Pull Request** from the **Operation Name** list.
5. On the **Input** tab:
  - Set the tpName attribute to Seller
  - Provide a valid URI in the **mpc** field
  - Provide the username and password  
Provide the same username and password that you set for the Send

UserMessage for Pulling activity. The username and password defined in the Global Variables must match the ones you configured in BusinessConnect Container Edition on Machine 2 when you define the external user and associate with the trading partner Buyer.

6. Click **Apply** to save the changes.
7. On the **Tester** tab, click the dropdown icon next to the **BWApplication** icon .
8. Select **Run Configuration** from the dropdown list.

**i Note:** If you want to run the process in the debug mode, click **Debug BWApplication** icon > **Debug Configurations**.

9. In the **Run Configurations** dialog, expand **BusinessWorks Application** > **BWApplication**.
10. From the list of the processes displayed, you can select one or multiple items. In this tutorial, select **Initiator** > **Send One Way Pull Request**.
11. Click **Apply** > **Run**.

You can see BusinessWorks logs on the **Console** tab.

## Expected Results

If everything works as expected, the user message is sent to BusinessConnect Container Edition successfully. Internally the message is processed and packaged and stored in Store-and-Forward, which is targeted to be pulled by the trading partner Buyer from this MPC channel.

BusinessConnect Container Edition on Machine 2 retrieves the user message from the specified MPC with the specified username and password, and then sends the user message with matched MPC, username, and password to Machine 1. The private process on Machine 1 receives the pulled message.

## Two-Way Asynchronous/Synchronous Transactions

This section describes the processes defined for two-way asynchronous/synchronous transactions.

This example contains the following processes in TIBCO ActiveMatrix BusinessWorks:

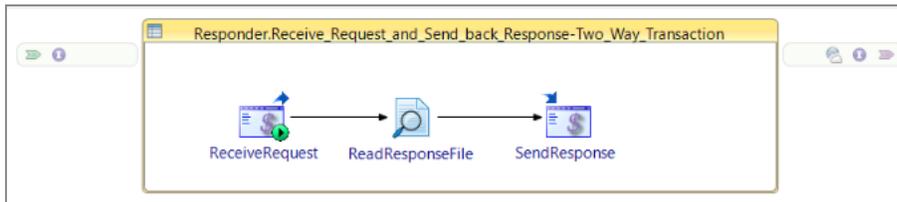
- [Send Request and Wait for Response - Two-Way Transaction](#)
- [Receive Request and Send back Response - Two-Way Transaction](#)

## Send Request and Wait for Response - Two-Way Transaction

Use the **Send Request and Wait for Response - Two Way Transaction** process on Machine 1 to initiate a two-way asynchronous or synchronous transaction. You can access this process from the TIBCO ActiveMatrix BusinessWorks project panel in the **Initiator** folder. The private process reads a request from a local directory and sends the document to the BusinessConnect Container Edition server, and waits for the responder response asynchronously or synchronously.

## Receive Request and Send back Response - Two-Way Transaction

Use the **Receive Request and Send back Response - Two Way Transaction** process on Machine 2 to respond to the request message. You can access this process from the TIBCO ActiveMatrix BusinessWorks project panel in the **Responder** folder.



## Running Two-Way Asynchronous/Synchronous Transactions

To run this example, complete the following tasks:

### Running Receive Request and Send back Response - Two Way Transaction on Machine 2

Follow these instructions to start the **Receive Request and Send back Response - Two Way Transaction** process on Machine 2:

#### Procedure

1. In the opened TIBCO ActiveMatrix BusinessWorks project on Machine 2, expand the

**Responder** folder.

2. Click the **Receive Request and Send back Response - Two Way Transaction** process in Project Explorer.
3. Click the **Receive Request** activity.
4. On the **General** tab, select **rosettanet.org/3A4/1.4/Purchase Order Request** from the **Operation Name** list.
5. Click **Apply** to save the changes.
6. On the **Tester** tab, click the dropdown icon next to the **BWApplication** icon .
7. Select **Run Configuration** from the dropdown list.

 **Note:** If you want to run the process in the debug mode, click **Debug BWApplication** icon > **Debug Configurations**.

8. In the **Run Configurations** dialog, expand **BusinessWorks Application > BWApplication**.
9. From the list of the processes displayed, you can select one or multiple items. In this tutorial, select **Responder > Receive Request and Send back Response - Two Way Transaction**.
10. Click **Apply > Run**.

You can see BusinessWorks logs on the **Console** tab.

## Running Send Request and Wait for Response - Two Way Transaction on Machine 1

Start the **Receive Request and Send back Response - Two Way Transaction** process on Machine 2 before proceeding with this section. Follow these instructions to start the **Send Request and Wait for Response - Two Way Transaction** process on Machine 1:

### Procedure

1. In the open TIBCO ActiveMatrix BusinessWorks project on Machine 1, expand the **Initiator** folder.
2. Click the **Send Request and Wait for Response - Two Way Transaction** process in

Project Explorer.

3. Click the **Send Purchase Request** activity.
4. On the **General** tab, select **rosettanet.org/3A4/1.4/Purchase Order Request** from the **Operation Name** list.
5. On the **Input** tab, enter **seller** in the **tpName** field.
6. Click **Apply** to save the changes.
7. On the **Tester** tab, click the dropdown icon next to the **BWApplication** icon .
8. Select **Run Configuration** from the dropdown list.

**i** **Note:** If you want to run the process in the debug mode, click **Debug BWApplication** icon > **Debug Configurations**.

9. In the **Run Configurations** dialog, expand **BusinessWorks Application** > **BWApplication**.
10. From the list of the processes displayed, you can select one or multiple items. In this tutorial, select **Initiator** > **Send Request and Wait for Response - Two Way Transaction**.
11. Click **Apply** > **Run**.

You can see BusinessWorks logs on the **Console** tab.

## Expected Results

If everything works as expected, the request is sent to the trading partner on Machine 2. Machine 1 waits for the response from this trading partner.

Machine 2 receives a request from the trading partner and sends the response asynchronously or synchronously to Machine 1. The private process on Machine 1 receives the response message. The process ends.

## Two-Way Push and Pull Transactions

This section describes the processes defined for two-way push and pull transactions.

This example contains the following processes in TIBCO ActiveMatrix BusinessWorks:

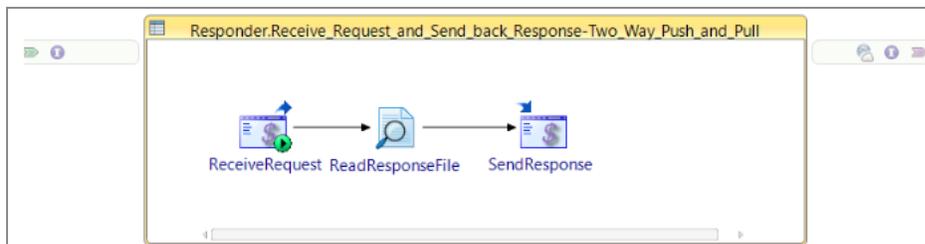
- [Send Request and Wait for Response - Two Way Push and Pull Transaction](#)
- [Receive Request and Send back Response - Two Way Push and Pull Transaction](#)

## Send Request and Wait for Response - Two Way Push and Pull Transaction

Use the **Send Request and Wait for Response - Two Way Push and Pull Transaction** process on Machine 1 to initiate a two-way push and pull transaction. You can access this process from the TIBCO ActiveMatrix BusinessWorks project panel in the **Initiator** folder. The private process reads a request from a local directory and sends the document to the BusinessConnect Container Edition server, and waits for the responder response asynchronously.

## Receive Request and Send back Response - Two Way Push and Pull Transaction

Use the **Receive Request and Send back Response - Two Way Push and Pull Transaction** process on Machine 2 to respond to the request message. You can access this process from the TIBCO ActiveMatrix BusinessWorks project panel in the **Responder** folder.



## Running Two-Way Push and Pull Transaction

To run this example, complete the following tasks:

### Running Receive Request and Send back Response - Two Way Push and Pull Transaction on Machine 2

Follow these instructions to start the **Receive Request and Send back Response - Two Way Push and Pull Transaction** process on Machine 2:

1. In the opened TIBCO ActiveMatrix BusinessWorks project on Machine 2, expand the **Responder** folder.
2. Click the **Receive Request and Send back Response – Two Way Push and Pull Transaction** process in Project Explorer.
3. Click the **Receive Request** activity.
4. On the **General** tab, select **tibco.com/EBMS Push and Pull/1.0/Push and Pull** from the **Operation Name** list.
5. Click **Apply** to save the changes.
6. On the **Tester** tab, click the dropdown icon next to the **BWApplication** icon .
7. Select **Run Configuration** from the dropdown list.

 **Note:** If you want to run the process in the debug mode, click **Debug BWApplication** icon > **Debug Configurations**.

8. In the **Run Configurations** dialog, expand **BusinessWorks Application > BWApplication**.
9. From the list of the processes displayed, you can select one or multiple items. In this tutorial, select **Responder > Receive Request and Send back Response - Two Way Push and Pull Transaction**.
10. Click **Apply > Run**.

You can see BusinessWorks logs on the **Console** tab.

## Running Send Request and Wait for Response - Two Way Push and Pull Transaction on Machine 1

Start the **Receive Request and Send back Response – Two Way Push and Pull Transaction** process on Machine 2 before proceeding with this section. Follow these instructions to start the **Send Request and Wait for Response – Two Way Push and Pull Transaction** process on Machine 1:

### Procedure

1. In the open TIBCO ActiveMatrix BusinessWorks project on Machine 1, expand the

**Initiator** folder.

2. Click the **Send Request and Wait for Response - Two Way Push and Pull Transaction** process in Project Explorer.
3. Click the **Send Purchase Request** activity.
4. On the **General** tab, select **tibco.com/EBMS Push and Pull/1.0/Push and Pull** from the **Operation Name** list.
5. On the **Input** tab, enter **seller** in the **tpName** field.
6. On the **Module Properties** tab, the username and password defined must match the ones you configured on Machine 2 in BusinessConnect Container Edition when you define the external user and associate with the trading partner Buyer. If you follow the instructions, the username is `exuser2@tibco.com`. See [Configuring External Users](#).
7. Click **Apply** to save the changes.
8. On the **Tester** tab, click the dropdown icon next to the **BWApplication** icon .
9. Select **Run Configuration** from the dropdown list.

**i Note:** If you want to run the process in the debug mode, click **Debug BWApplication** icon > **Debug Configurations**.

10. In the **Run Configurations** dialog, expand **BusinessWorks Application > BWApplication**.
11. From the list of the processes displayed, you can select one or multiple items. In this tutorial, select **Initiator > Send Request and Wait for Response - Two Way Push and Pull Transaction**.
12. Click **Apply > Run**.

You can see BusinessWorks logs on the **Console** tab.

## Expected Results

If everything works as expected, the request with the specified username and password is sent to the trading partner on Machine 2. After sending a pulled message, Machine 1 waits for the response from the trading partner.

Machine 2 receives a request from the trading partner with the specified username and password, processes it and the response is stored in Stored-and-Forward that is targeted to be pulled by the trading partner Buyer from this MPC. After receiving the pulling request,

Machine 2 sends the response asynchronously with the matched username and password to Machine 1. The private process on Machine 1 receives the response message. The process ends.

# Status Codes

---

This section describes the status codes used by the BusinessConnect Container Edition - ebXML Protocol ebMS3/AS4 standard. 200~699 are standard HTTP codes and BusinessConnect Container Edition-defined codes, when 1100~4999 are codes defined by BusinessConnect Container Edition - ebXML Protocol.

These status codes and their descriptions are displayed in the following locations:

- **Private messages** The following private message classes include the **statusCode** and **statusMsg** fields. See [Private Process Message Formats](#):
  - ae/ebMS3/InitiatorRequest
  - ae/ebMS3/InitiatorResponse
  - ae/ebMS3/Advisory (for both error and advisory signal messages)
- **Audit log** The **Description** field in the transaction detail view displays a status code with accompanying description. See Transaction Details View.

## Status Codes

Code (statusCode field)	Description (statusMsg field)	Role	Comment
200	OK	HTTP(S)	
201~299	HTTP(S) OK codes	HTTP(S)	
300~503	HTTP(S) error codes	HTTP(S)	
510	No valid HTTP response.	HTTP(S)	The most likely causes are that the trading partner sent a trading partner identity that was not

Code (statusCode field)	Description (statusMsg field)	Role	Comment
			recognized by BusinessConnect Container Edition, or that the trading partner did not send a correct HTTP response.
699	Failed to connect to trading partner.	BusinessConnect Container Edition	
1100	Received [request response error] message from TP.	Information	
1101	Received [signed ] [encrypted ] [sync/async ] [action ] from TP.	Information	
1102	Ignoring [Request/response Receipt Acknowledgment MSH Acknowledgment] message. Reason: <i>detailed_description</i> .	Information	
1651	Receipt Acknowledgment received for message ID: <i>message_id</i> .	Information	The trading partner has received a receipt acknowledgment for the indicated message.
1652	Acceptance Acknowledgment received for message ID: <i>message_id</i> .	Information	
1653	MSH Acknowledgment received	Information	The trading partner

Code (statusCode field)	Description (statusMsg field)	Role	Comment
	for message ID: <i>message_id</i> .		has sent an MSH acknowledgment for the indicated message.
1999	Info: <i>detailed_description</i> .	Information	
2999	Warning: <i>detailed_description</i> .	Warning	
3101	Missing either operation ID or To TP name.	Error	
3102	Configuration read error. Reason: <i>detailed_description</i> .	Error	
3103	Operation not allowed. Reason: <i>detailed_description</i> .	Error	
3104	Host Identity for Authentication not configured - message cannot be signed.	Error	
3105	To or From Role not specified - receipt acknowledgment cannot be constructed.	Error	
3106	Partner Identity for <i>trading_partner_name</i> not configured - [Request Response Receipt MshAck Error Exception] cannot be [signed encrypted].	Error	
3107	Config read failed for remote TP. Reason: <i>detailed_description</i> .	Error	

Code (statusCode field)	Description (statusMsg field)	Role	Comment
3108	Config read failed for response action from TP.	Error	
3109	Inconsistency between configuration and message. Reason: <i>detailed_description</i> .	Error	
3110	Unknown Service Action. Service= <i>service</i> Service type= <i>service_type</i> Action= <i>action</i> .	Error	
3111	To party Domain <i>domain</i> and Identity <i>identity</i> does not match with configuration.	Error	
3112	Host not allowed to send the response to TP <i>trading_partner_name</i> . Reason: <i>detailed_description</i> .	Error	
3113	Failed to determine the transport. Reason: <i>detailed_description</i> .	Error	
3114	Private process supplied 'businessDocumentName' of <i>businessDocumentName</i> does not match configured names of <i>name_in_GUI</i> .	Error	
3115	Message ordering is only available for transactions with Once-And-Only-Once semantics and SyncReplyMode of 'none'.	Error	
3116	'MessageOrder' element must not be present with the 'SyncReply' element.	Error	

Code (statusCode field)	Description (statusMsg field)	Role	Comment
3117	Time To Live expired	Error	
3118	Failed to get payload against the manifest. Reason: <i>detailed_description</i> .	Error	
3119	CPAId <i>cpa_id</i> does not match with configuration.	Error	
3120	Version <i>business_process_version</i> is not supported.	Error	
3121	From party Domain <i>domain</i> and Identity <i>identity</i> does not match with configuration.	Error	
3122	CPAID in CPA and EBMS3 document does not match.	Error	
3123	Authorization info is missing when mpc is specified by back office for Pull Request message.	Error	
3124	Username or password is not supplied by private process.	Error	
3125	The Receipt has been received for a message that was previously sent by the MSH Invalid.	Error	
3133	The consistency check of AT0/SBR SAML Security failed. The AT0/SBR SAML Security and Require Non-repudiation of Request checkboxes are required to	Error	

Code (statusCode field)	Description (statusMsg field)	Role	Comment
	be selected on request action.		
3201	Post aborted - exhausted maximum number of retries ( <i>max_retries</i> ).	Error	<i>max_retries</i> is the maximum number of retries configured in the GUI.
3202	Unknown Error while sending [Request/response Receipt Acknowledgment MSH Acknowledgment].	Error	
3203	Posting to <i>trading_partner_name</i> failed with code <i>http_return_code</i> . Reason: <i>detailed_description</i> .	Error	
3204	The MSH is experiencing temporary or permanent failure in trying to open a transport connection with a remote MSH.	Error	The same as EBMS: 0005.  See <a href="#">ebMS Errors</a> .
3301	Timed out waiting for [sync async] [Acknowledgment Receipt Acknowledgment Acceptance Acknowledgment].	Error	
3302	Unexpected error while waiting for async response. Reason: <i>detailed_description</i> .	Error	
3303	No MSH Acknowledgment received after exhausting sending max. number of retries ( <i>max_retries</i> ).	Error	<i>max_retries</i> is the maximum number of retries

Code (statusCode field)	Description (statusMsg field)	Role	Comment
			configured in the GUI.
3304	No response data was received from trading partner.	Error	
3305	No MSH Receipt received after sending back pulled message.	Error	
3401	MIME formatting error. Reason: <i>detailed_description</i> .	Error	
3402	EBMS3 SOAP envelope generation error. Reason: <i>detailed_description</i> .	Error	
3403	Failed to create MSH Ack Envelope. Reason: <i>detailed_description</i> .	Error	
3404	Failed to create Receipt acknowledgment. Reason: <i>detailed_description</i> .	Error	
3405	Failed to create Attachment Sequence. Reason: <i>detailed_description</i> .	Error	
3501	Client authentication failed for Service= <i>service</i> Action= <i>action</i> . Reason: <i>detailed_description</i> .	Error	
3502	[S/MIME XML] Encryption failed for Service= <i>service</i> Action= <i>action</i> . Reason: <i>detailed_description</i> .	Error	

Code (statusCode field)	Description (statusMsg field)	Role	Comment
	[S/MIME XML] Encryption failed. Reason: <i>detailed_description</i> .		
3502-1	[S/MIME XML] Encryption failed. Reason: <i>detailed_description</i> .	Error	Short error message for 3502.
3503	[S/MIME XML] Decryption failed for Service= <i>service</i> Action= <i>action</i> . Reason: <i>detailed_description</i> .  [S/MIME XML] Decryption failed. Reason: <i>detailed_description</i> .	Error	
3503-1	[S/MIME XML] Decryption failed. Reason: <i>detailed_description</i> .	Error	Short error message for 3503.
3504	EBMS3 SOAP envelope signing failed for Service= <i>service</i> Action= <i>action</i> . Reason: <i>detailed_description</i> .	Error	
3504-1	EBMS3 SOAP envelope signing failed. Reason: <i>detailed_description</i> .	Error	Short error message for 3504.
3505	EBMS3 SOAP envelope signature verification failed for Service= <i>service</i> Action= <i>action</i> . Reason: <i>detailed_description</i> .	Error	
3505-1	EBMS3 SOAP envelope signature verification failed. Reason: <i>detailed_description</i> .	Error	Short error message for 3505.

Code (statusCode field)	Description (statusMsg field)	Role	Comment
3506	MSH Acknowledgment signature references verification failed.	Error	
3507	Receipt Acknowledgment NR digests verification failed. Reason: <i>detailed_description</i> .	Error	
3508	UsernameToken Authentication Processing failed for Service= <i>service</i> Action= <i>action</i> . Reason: <i>detailed_description</i> .	Error	
3509	WSS Security Processing failed for Service= <i>service</i> Action= <i>action</i> . Reason: <i>detailed_description</i> .	Error	
3601	MIME message parsing error. Reason: <i>detailed_description</i> .	Error	
3602	EBMS3 SOAP envelope parsing error. Reason: <i>detailed_description</i> .	Error	
3603	Error while processing response from TP. Reason: <i>detailed_description</i> .	Error	
3604	Unable to compute the operation ID from message.	Error	
3605	Failed to validate EBMS3 envelope parameters against configuration. Reason: <i>detailed_description</i> .	Error	
3606	Inbound Payload/Attachments	Error	

Code (statusCode field)	Description (statusMsg field)	Role	Comment
	extraction error. Reason: <i>detailed_description</i> .		
3607	Incorrect RefToMessageId for [Request/response Receipt Acknowledgment MSH Acknowledgment] message.	Error	
3608	Sync Response Initialization failed.	Error	
3609	Failed to process Synchronous [response MSH Acknowledgment].	Error	
3610	Error while sending request to private process. Reason: <i>detailed_description</i> .	Error	
3611	Timeout waiting for [response acknowledgment] from private process.	Error	
3612	SOAP Fault (code: [server client], text: <i>brief_</i> <i>description</i> , actor: <i>actor</i> , details: <i>detailed_description</i> ).	Error	
3613	MSH Error (severity: [Warning Error]).	Error	<i>msh_error_code</i> is one of the MSH error codes defined by the ebMS specifications.
3614	Application supplied a duplicate message ID.	Error	
3615	Could not find expected SOAP Fault message.	Error	

Code (statusCode field)	Description (statusMsg field)	Role	Comment
3616	Unexpected error while processing sequenced messages. Reason: <i>detailed_description</i> .	Error	
3617	Received message missing expected SequenceNumber.	Error	
3618	Sequenced message timed out while queued.	Error	
3619	Application supplied a duplicate message ID - not allowed for Once-And-Only-Once semantics.	Error	
3620	Sequenced message timed out while received.	Error	
3621	HTTP reply does not include a content but status code is not 204.	Warning	
3622	There is no message for mpc: <i>mpcuri</i> .	Error	
3623	There is no message available for pulling, mpc= <i>mpcuri</i> .	Error	
3624	The Receipt has not been received for message: <i>message description</i> .	Error	
3650	Generic Exception received for messageID: <i>message_id</i> . Reason: <i>detailed_description</i> .	Error	Used by the responder private process to indicate that an exception occurred when

Code (statusCode field)	Description (statusMsg field)	Role	Comment
			processing an inbound request.  See <a href="#">Outbound Response Format</a> .
3651	Receipt Exception received for messageID: <i>message_id</i> . Reason: <i>detailed_description</i> .	Error	
3652	Acceptance Exception received for messageID: <i>message_id</i> . Reason: <i>detailed_description</i> .	Error	Used by the responder private process to indicate that an exception occurred when processing an inbound request.  See <a href="#">Outbound Response Format</a> .
3691	Processing failed for Service= <i>service</i> Action= <i>action</i> . Reason: <i>detailed_description</i> .	Error	
3692	Failed to process [MSH Receipt] Acknowledgment for [request response Receipt Acknowledgment]. Reason: <i>detailed_description</i> .	Error	
3699	General processing error: <i>detailed_description</i> .	Error	
3701	Reliable storage insert failed. Reason: <i>detailed_description</i> .	Error	

<b>Code (statusCode field)</b>	<b>Description (statusMsg field)</b>	<b>Role</b>	<b>Comment</b>
3702	Reliable storage update failed. Reason: <i>detailed_description</i> .	Error	
3703	Reliable storage query failed. Reason: <i>detailed_description</i> .	Error	
3801	EBMS3 SOAP envelope validation error. Reason: <i>detailed_description</i> .	Error	
3802	[Incoming Outgoing] request payload failed schema validation. Reason: <i>detailed_description</i> .	Error	
3803	[Incoming Outgoing] response payload failed schema validation. Reason: <i>detailed_description</i> .	Error	
3804	Receipt acknowledgment failed schema validation.	Error	
3805	Missing schema URI for validation.	Error	
3806	Message validation error. Reason: <i>detailed_description</i> .	Error	
3807	Exception message failed schema validation. Reason: <i>detailed_description</i> .	Error	
3808	Processing Mode mismatch.	Error	
3901	Reply to DMZ failed. Reason: <i>detailed_description</i> .	Error	

Code (statusCode field)	Description (statusMsg field)	Role	Comment
3991	Unkown Internal Engine Error. Reason: <i>detailed_description</i> .	Error	
3992	Missing error details resource. Reason for original error: <i>detailed_description</i> .	Error	Failed to find the error description resource. This is most likely because of packaging error.
3993	RefId does not exist.	Error	
3994	The digest associated does not match the signature digest for NRR.	Error	
3995	code: <i>code</i> , short description: <i>description</i> , severity: <i>severity</i> , category: <i>category</i> , origin: <i>origin</i> , refMessageID: <i>refmessageid</i> , error detail: <i>error detail</i> , description: <i>description</i> .	Error	The EBMS3 standard errors.  See <a href="#">ebMS Errors</a> .
3997	User info not exist. Reason: <i>detailed_description</i> .	Error	
4999	Debug: <i>detailed_description</i> .	Error	

# Property Reference

---

The following table lists the properties that you can set under **BusinessConnect Container Edition > System Settings > Others > Activate Protocol Plugins > ebXML**.

## Property Reference

Property	Description
<code>ebxml.sequence.pollInterval</code>	(In milliseconds) The interval at which BusinessConnect Container Edition - ebXML Protocol polls for queued messages that are ready to be dispatched and messages that have expired.  <b>Default:</b> 300000 milliseconds
<code>ebxml.sequence.iterateWait</code>	(In milliseconds) The interval at which BusinessConnect Container Edition - ebXML Protocol dispatches queued messages to the private processes.  <b>Default:</b> 5000 milliseconds
<code>ebxml.duns.check</code>	Enforces the correct identity format in the Participants configuration panel when the DUNS domain is used. Select <b>duns</b> to enforce a nine-digit DUNS number, <b>duns+4</b> to enforce a thirteen-digit DUNS number, and <b>none</b> to turn off this function.
<code>ebxml.fileref.threshold</code>	(In bytes) Content size threshold to use file reference in sending messages to private processes. <b>Default:</b> 5 MB (5000000).
<code>ebms3.persist.duration</code>	(In minutes) Maximum length of time that the data from a sent message is kept in persistent storage by a receiving MSH.  <b>Default:</b> 120 minutes

Property	Description
ebms3.ob.keyExchange.algo	The encryption algorithm for the encrypted key exchange.  The available options are: <code>rsa-1_5</code> , <code>rsa-oaep</code> , or <code>rsa-oaep-mgf1p</code> .
ebms3.at0.sbr.sdk.classpath	The directory in which all the required SBR Authentication Java Client SDK JAR files and related third-party JAR files are contained.
ebms3.engine.name.closure	Allows to add engine name to the closure tag to INITIATOR.RESPONSE for push-pull transactions.
ebxml.samlToken.storeInRuntime	Saves and manages the SAML token in the BusinessConnect runtime store.

# Smart Routing

---

By using BusinessConnect Container Edition, you can define business rules to route messages to specific private processes. This function is called smart routing, which requires configuration of the BusinessConnect Container Edition server through the BusinessConnect Container Edition console, and configuration of the BusinessConnect palette resources in TIBCO Business Studio. In the BusinessConnect Container Edition console, you can set up the business rules and specify the smart ID to be assigned to messages that meet the conditions of the rule. In the BusinessConnect palette resources, you can configure which private processes receive messages that include specific smart IDs.

To configure the smart routing fields, go to **BusinessConnect Container Edition > System Settings > Others > Smart Routing > Private Process Smart Routing > Add New Smart Route**.

## Smart Routing Fields for ebMS3

Field	Description
<b>Enabled</b>	Select this checkbox to use the private process smart routing.
<b>Protocol</b>	The business protocol for the message. Use the asterisk character (*) to match all protocols.  This is a required field.
<b>From</b>	The domain identity for the trading partner that sends the original message. If Host that is your company sends a request to Partner and Partner sends a response, you can use smart routing for the response. In this case, the <b>From</b> field is matched by Host because Host is the originator of the business transaction.  Use the asterisk character (*) to match all hosts and partners, but do not use the asterisk character with a string. For example, do not use TIB*.  This is a required field.

Field	Description
<b>To</b>	<p>The domain identity for the trading partner that receives the original message. If a partner sends a request to a host that is your company, you can use smart routing for the request. In this case, the <b>To</b> field is matched by Host because Host is the recipient of the request.</p> <p>Use the asterisk character (*) to match all hosts and partners, but do not use the asterisk character with a string. For example, do not use TIB*.</p> <p>This is a required field.</p>
<b>Direction</b>	<p>The business direction of the message: inbound or outbound. For example, if a partner sends a request to a host that is your company, both the business direction and the message direction are inbound. However, if Host sends a request to Partner and Partner sends a response, the message direction of the response is inbound, but the business direction of the response is outbound because the original message was outbound.</p> <p>The asterisk character (*) matches both directions. This is a required field.</p>
<b>Operation ID</b>	<p>The location and identifier of the operation. This takes the form of a series of nodes.</p> <p>Use one asterisk character (*) to match all operations directly under a specific node. For example:</p> <p>BCCE/*/* matches BCCE/MyNotify/Test but not BCCE/MyNotify/Test/notify1</p> <p>Use two asterisk characters (**) to match operations recursively. Use double asterisks alone or use them as the last node. For example:</p> <p>BCCE/MyNotify/** matches BCCE/MyNotify/1.3/Test</p> <p>BCCE/MyNotify/**/notify1 is the same as BCCE/MyNotify/**. The software ignores any nodes after a double asterisk.</p> <p>You can use both a single asterisk and a double asterisk, for example:</p> <p>BCCE*/1.0/** matches BCCE/Test-01/1.0/A/B</p> <p>This is a required field.</p>
<b>Smart ID</b>	<p>An identifier that indicates which smart routing rules the message satisfied.</p>

---

Field	Description
	Any combination of alphanumeric characters can be used, with a minimum of one character and a maximum of 25 characters.
	This is a required field.

---

## Public Messages

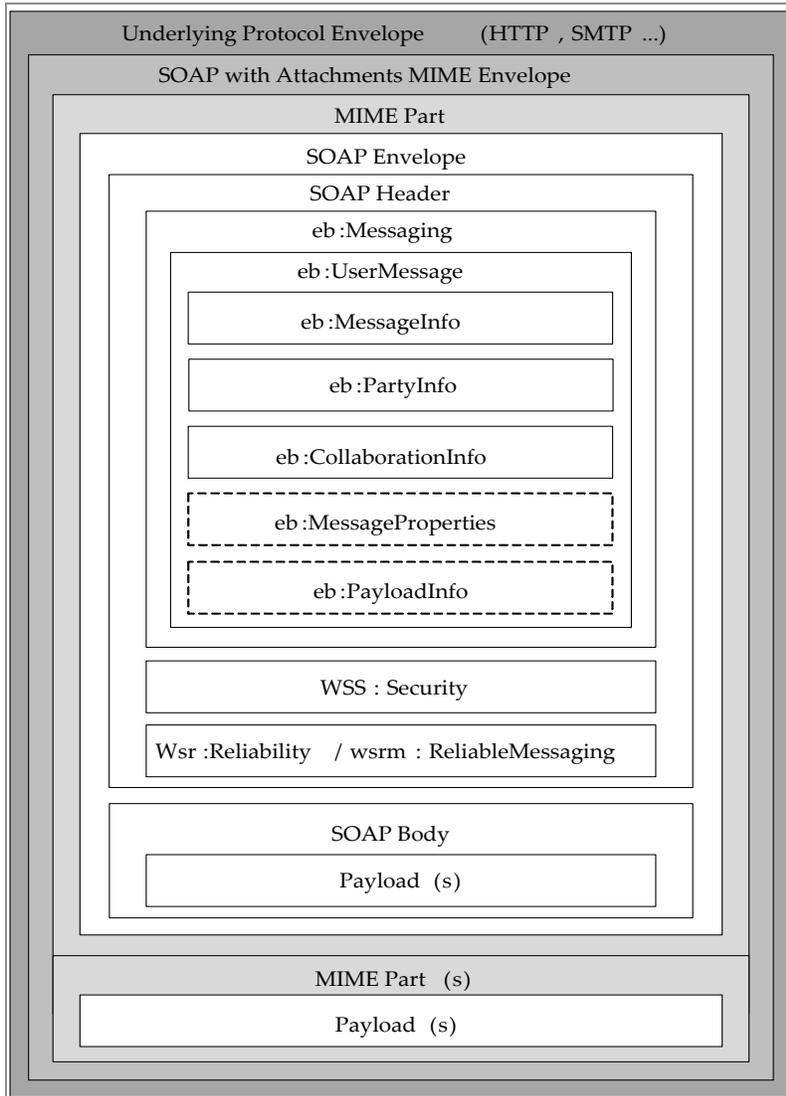
---

This section describes the elements and attributes in the ebMS3 messages and how they correspond to GUI elements in the BusinessConnect Container Edition console or to private message fields.

### ebMS3 Public Message Structure

The structure of an ebMS user message is shown in the following figure:

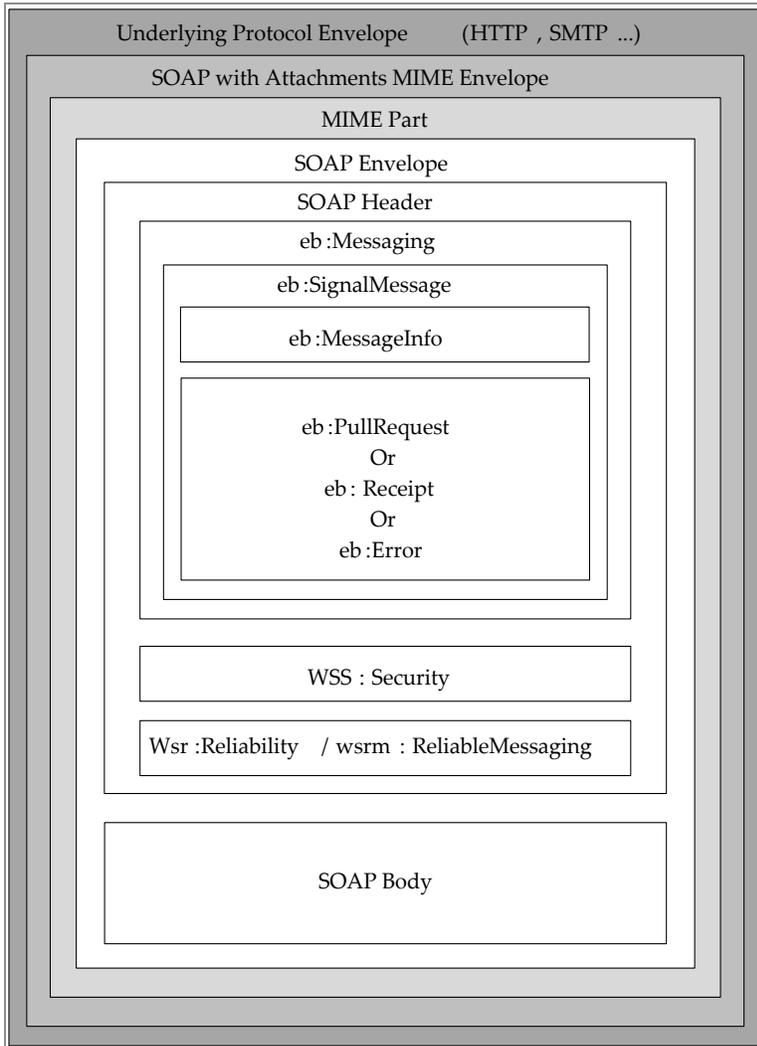
Figure 12: ebMS3 User Message Structure



**i Note:** Wsr:Reliability and wsr:ReliableMessaging are not implemented because they are not required by AS4 profile.

The structure of an ebMS3 signal message is shown in the following figure:

Figure 13: ebMS3 Signal Message Structure



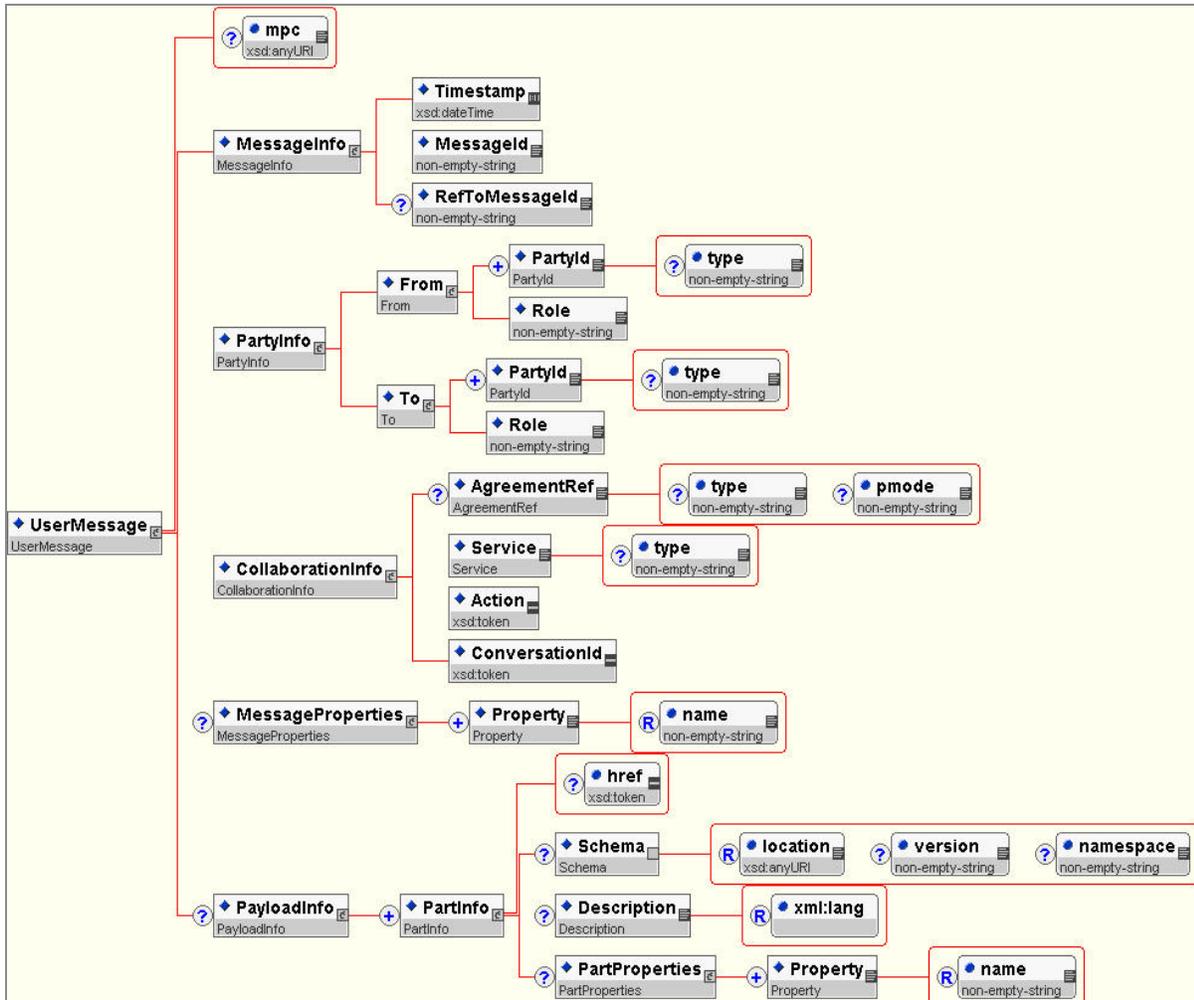
**i Note:** Wsr:Reliability and wsr:ReliableMessaging are not implemented because they are not required by AS4 profile.

## SOAP Envelope

This section describes the elements and attributes in the ebXML SOAP envelope and how they correspond to GUI elements in the BusinessConnect Container Edition console or to private message fields.

# UserMessage Element

The UserMessage element, as defined by the ebXML Message Service Specification version 3.0, is shown in the following diagram.



See the following table for the corresponding GUI element in the BusinessConnect Container Edition console or the corresponding private message field of each element or attribute.

### UserMessage Element in ebMS3 Messages

Element/Attribute	Source of Value
mpc	Identifies the message partitioning channel to

Element/Attribute	Source of Value
	which the message is assigned.
<b>MessageInfo</b>	
Timestamp	This is for outbound message only.  Automatically generated by BusinessConnect Container Edition - ebXML Protocol.
MessageId	This is for outbound message only.  The <b>messageID</b> field in the <code>ae/ebMS3/InitiatorRequest</code> message class. If a value is not specified, BusinessConnect Container Edition - ebXML Protocol randomly generates one. See <a href="#">Outbound Request Format</a> .
RefToMessageId	The <code>MessageData/MessageId</code> element in the related inbound message. This element is not included in request messages.
<b>PartyInfo</b>	
From/PartyId	The ID string, which is without the domain string, is in  <b>BusinessConnect Container Edition &gt; Partner Management &gt; Partners &gt; participant &gt; Protocols &gt; ebMS3 &gt; Edit Configurations &gt; General &gt; Default Domain Identity.</b>
From/PartyId/type	The domain string, which is without the domain string, is in  <b>BusinessConnect Container Edition &gt; Partner Management &gt; Partners &gt; participant &gt; Protocols &gt; ebMS3 &gt; Edit Configurations &gt; General &gt; Default Domain Identity.</b>

Element/Attribute	Source of Value
From/Role	<p data-bbox="771 310 1339 415"><b>Note:</b> For outbound messages, this attribute is not included if the domain string is URI.</p> <p data-bbox="751 478 1370 632"><b>BusinessConnect Container Edition &gt; B2B Administration &gt; Operations Editor &gt; ebMS3 &gt; transaction &gt; name &gt; Transaction &gt; Roles &gt; Initiating Role (or Responding Role).</b></p> <p data-bbox="751 659 1354 695">Or, in the case of override in operation binding:</p> <p data-bbox="751 722 1411 873"><b>BusinessConnect Container Edition &gt; Partner Management &gt; Business Agreements &gt; agreement &gt; ebMS3 &gt; Operations &gt; transaction &gt; Initiating Role.</b></p>
To/PartyId	<p data-bbox="751 1121 1409 1192">The ID string, which is without the domain string, is in</p> <p data-bbox="751 1226 1414 1377"><b>BusinessConnect Container Edition &gt; Partner Management &gt; Partners &gt; participant &gt; Protocols &gt; ebMS3 &gt; Edit Configurations &gt; General &gt; Default Domain Identity.</b></p>
To/PartyId/type	<p data-bbox="751 1423 1409 1495">The domain string, which is without the ID string, is in</p> <p data-bbox="751 1528 1414 1675"><b>BusinessConnect Container Edition &gt; Partner Management &gt; Partners &gt; participant &gt; Protocols &gt; ebMS3 &gt; Edit Configurations &gt; General &gt; Default Domain Identity.</b></p>

Element/Attribute	Source of Value
To/Role	<p data-bbox="771 310 1338 411"><b>Note:</b> For outbound messages, this attribute is not included if the domain string is URI.</p> <p data-bbox="751 478 1370 632"><b>BusinessConnect Container Edition &gt; B2B Administration &gt; Operations Editor &gt; ebMS3 &gt; transaction &gt; name &gt; Transaction &gt; Roles &gt; Initiating Role (or Responding Role).</b></p> <p data-bbox="751 659 1354 690">Or, in the case of override in operation binding:</p> <p data-bbox="751 722 1354 873"><b>BusinessConnect Container Edition &gt; Partner Management &gt; Business Agreements &gt; agreement &gt; ebMS3 &gt; Edit Configurations &gt; Operations &gt; transaction &gt; Initiating Role.</b></p> <p data-bbox="771 919 1378 1052"><b>Note:</b> If this field is empty, and then the To/Role element is not included. See <a href="#">Operation Settings Tab</a>.</p>
<b>CollaborationInfo</b>	
AgreementRef/type	<b>BusinessConnect Container Edition &gt; Partner Management &gt; Partners &gt; participant &gt; Protocols &gt; ebMS3 &gt; Edit Configurations &gt; General &gt; AgreementRef.</b>
AgreementRef/pmode	<b>BusinessConnect Container Edition &gt; B2B Administration &gt; Operations Editor &gt; ebMS3 &gt; transaction &gt; name &gt; Transaction &gt; Service Information &gt; P-Mode ID.</b> This value can be overridden in Business Agreement Operation Binding.
Service	<b>BusinessConnect Container Edition &gt; B2B Administration &gt; Operations Editor &gt; ebMS3 &gt;</b>

Element/Attribute	Source of Value
	<b>transaction &gt; name &gt; Transaction &gt; Service Information &gt; Service</b>
Service/type	<b>BusinessConnect Container Edition &gt; B2B Administration &gt; Operations Editor &gt; ebMS3 &gt; transaction &gt; name &gt; Transaction &gt; Service Information &gt; Service Type.</b>
Action	<b>BusinessConnect Container Edition &gt; B2B Administration &gt; Operations Editor &gt; ebMS3 &gt; transaction &gt; name &gt; Transaction &gt; Action &gt; General &gt; Name.</b>
ConversationId	This is for outbound message only.  The <b>conversationID</b> field in the <code>ae/ebMS3/InitiatorRequest</code> message class. If a value is not specified, BusinessConnect Container Edition - ebXML Protocol randomly generates one in the outbound sequenced message. See <a href="#">Outbound Request Format</a> .

## MessageProperties

Property/name	The property name of the message that must be agreed on between partners.
---------------	---------------------------------------------------------------------------

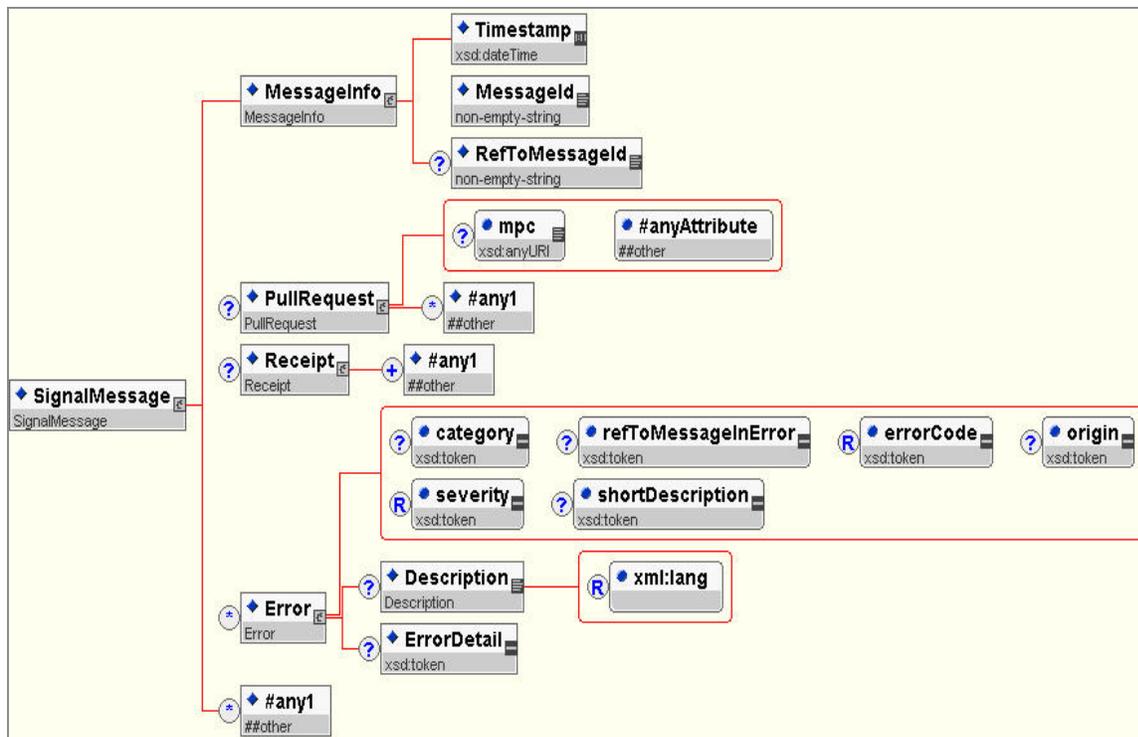
## PayloadInfo

PartInfo	The supplemental information of a payload, either the payload in the SOAP Body, or a payload in an attachment.
PartInfo/href	The reference of the payload for which this PartInfo is related. For outbound messages, it can be specified from a private process or automatically generated by BusinessConnect Container Edition -

Element/Attribute	Source of Value
PartInfo/Schema	The schema of the payload. You can use this file to specify the schema URI, which can be useful to the recipient.
PartInfo/Description	A short description of the related payload and can be specified from the private process.
PartInfo/PartProperties/Property	The customer property name and value pairs regarding this payload that can be specified by the sender.

## SignalMessage Element

The SignalMessage element, as defined by the ebXML Message Service Specification version 3.0, is shown in the following diagram.



See the following table for the corresponding GUI element in the BusinessConnect Container Edition console or the corresponding private message field of each element or attribute.

### Signal Message Element in ebMS3 Messages

Element/Attribute	Source of Value
any	Not included.

### MessageInfo

Timestamp	This is for outbound message only.  Automatically generated by BusinessConnect Container Edition - ebXML Protocol.
MessageId	This is for outbound message only.  The <b>messageID</b> field in the ae/ebMS3/InitiatorRequest message class. If a value is not specified there, and then BusinessConnect Container Edition - ebXML Protocol randomly generates one. See <a href="#">Outbound Request Format</a> .
RefToMessageId	The MessageData/MessageId element in the related inbound message. This element is not included in request messages.

### PullRequest

mpc	Identifies the message partitioning channel to which the message is assigned.
any	Not included.

### Receipt

any	A single ebbpsig:NonRepudiationInformation child element or a copy of the original user message element.
-----	----------------------------------------------------------------------------------------------------------

### Error

<b>Element/Attribute</b>	<b>Source of Value</b>
Description	A narrative description of the error in the language defined by the <code>xml:lang</code> attribute.
ErrorDetail	The detailed description of the error message.
Error/category	The type of the error, for example, Content, Packaging, UnPackaging, Communication, and InternalProcess.
Error/ refToMessageInError	The MessageId of the message in the error, for which this error is raised.
Error/errorCode	The unique identifier for the type of error.
Error/origin	The functional module within which the error occurred. Possible values for this attribute include ebMS, reliability, and security.
Error/severity	The severity of the error. Possible values are warning and failure.
Error/shortDescription	The short description of the error.

# TIBCO Documentation and Support Services

---

For information about this product, you can read the documentation, contact TIBCO 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 following documentation for this product is available on the [TIBCO BusinessConnect™ Container Edition - ebXML Protocol Product Documentation](#) page:

- *TIBCO BusinessConnect™ Container Edition - ebXML Protocol Release Notes*
- *TIBCO BusinessConnect™ Container Edition - ebXML Protocol Installation*
- *TIBCO BusinessConnect™ Container Edition - ebXML Protocol User Guide - ebMS2 Standard*
- *TIBCO BusinessConnect™ Container Edition - ebXML Protocol User Guide - ebMS3/AS4 Standard*

## Other TIBCO Product Documentation

When working with TIBCO BusinessConnect™ Container Edition - ebXML Protocol, you may find it useful to read the documentation of the following TIBCO products:

- TIBCO BusinessConnect™ Container Edition
- TIBCO Enterprise Message Service™
- TIBCO ActiveMatrix BusinessWorks™

- TIBCO ActiveMatrix BusinessWorks™ Plug-in for BusinessConnect

## 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, ActiveMatrix BusinessWorks, BusinessConnect, BusinessConnect Container Edition, Business Studio, Enterprise Message Service, and TIBCO Business Studio are either registered trademarks or trademarks of Cloud Software Group, Inc. in the United States and/or other countries.

Java and all Java based trademarks and logos are trademarks or registered trademarks of Oracle 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. 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.tibco.com/patents>.

Copyright © 2000-2023. Cloud Software Group, Inc. All Rights Reserved.