



TIBCO BusinessConnect™ SOAP Protocol

User's Guide

*Software Release 7.1
October 2019*



Important Information

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

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

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

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

TIBCO, the TIBCO logo, the TIBCO O logo, TIB, Information Bus, TIBCO BusinessConnect, TIBCO Runtime Agent, TIBCO ActiveMatrix BusinessWorks, TIBCO ActiveMatrix BusinessWorks Plug-in for BusinessConnect, TIBCO BusinessConnect Palette, TIBCO Administrator, TIBCO Designer, TIBCO Rendezvous, and TIBCO Enterprise Message Service are either registered trademarks or trademarks of TIBCO Software Inc. in the United States and/or other countries.

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

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

This software may be available on multiple operating systems. However, not all operating system platforms for a specific software version are released at the same time. See the readme file for the availability of this software version on a specific operating system platform.

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

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

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

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

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

Contents

Figures	ix
Tables	xi
Preface	xiii
Related Documentation	xiv
TIBCO BusinessConnect SOAP Protocol Documentation	xiv
Other TIBCO Product Documentation	xiv
Third Party Documentation	xiv
Typographical Conventions	xvi
TIBCO Product Documentation and Support Services	xviii
How to Access TIBCO Documentation	xviii
How to Contact TIBCO Support	xviii
How to Join TIBCO Community	xviii
Chapter 1 Introduction to SOAP and TIBCO BusinessConnect SOAP Protocol	1
SOAP Overview	2
TIBCO BusinessConnect SOAP Protocol Overview	3
TIBCO BusinessConnect SOAP Protocol Features	4
Chapter 2 Process Flows and Functionalities	7
TIBCO BusinessConnect SOAP Protocol Messages	8
Private Messages and Processes	8
Public Messages	9
Operation Types and Process Flows	10
Notify Operations	10
Synchronous Request-Response Operations	11
Duplicate Message Detection	15
Inbound Messages	15
Outbound Messages	16
SOAP Message Security	17
Digital Signature and Encryption	17
UsernameToken for Authentication	17
NHIN/esMD Authorization Framework with SAML Assertions	19

WS-Addressing	20
MTOM	22
SOAP Fault	23
Chapter 3 Standalone Private Processes Tutorial	25
Overview	26
Operations	26
Trading Partners	26
Transactions	26
Initiator Machine	28
Responder Machine	28
Prerequisites	29
Configuring the Initiator	30
Importing Operations on the Initiator	30
Setting Up the Initiator Server	31
Setting Up the Initiator as a Trading Host	32
Setting Up the Responder as a Trading Partner	33
Configuring the Business Agreement between the Initiator and the Responder	35
Configuring the Responder	36
Importing Operations on the Responder	36
Setting Up the Responder Server	37
Setting Up the Responder as a Trading Host	38
Setting Up the Initiator as a Trading Partner	39
Configuring the Business Agreements between the Initiator and the Responder	40
Running the Tutorial	41
Running the Standalone Private Processes	41
Viewing the Audit Logs	47
Chapter 4 TIBCO ActiveMatrix BusinessWorks Private Processes Tutorial	49
Overview	50
The TIBCO ActiveMatrix BusinessWorks Project Tutorial	50
Transactions	51
Setting Up the Tutorial	53
Prerequisites	53
Configuring the Machines	53
Configuring TIBCO ActiveMatrix BusinessWorks Private Processes	54
Configuring Private Processes in TIBCO Designer	54
Configuring Private Processes in TIBCO Business Studio	56
Introduction of Process Definitions	58
Introducing Initiator Process Definitions	58
Introducing Responder Process Definitions	60

Introducing NHIN/esMD Process Definitions	62
Running the Tutorial.	65
Running the TIBCO ActiveMatrix BusinessWorks Processes.....	65
Viewing the Audit Logs	67
Chapter 5 Preparing Information with Your Trading Partners	69
Reaching Agreement on Shared Business Documents.....	70
Exchanging URI Definitions.....	71
Exchanging Identity Information.....	72
Chapter 6 Managing SOAP Operations	73
Overview	74
Creating a SOAP Interface	76
Adding a Version to an Interface	78
Adding an Operation to a Version	79
SOAP Operation Properties.....	80
Notify Operation	80
Synchronous Request-Response Operation	84
Importing SOAP Interfaces	90
Exporting SOAP Interfaces	91
Modifying SOAP Operations	92
Modifying the Notify Operation	92
Modifying the Sync Operation.....	93
Chapter 7 Setting Up Trading Hosts and Partners	97
Configuring the SOAP Protocol for a Host.....	98
.....	98
Adding a Domain Identity for a Host.....	99
Configuring the SOAP Protocol for a Partner	100
Configuring General Properties for a Partner.....	100
Configuring Transports for a Partner	103
Chapter 8 Configuring Business Agreements	105
Configuring SOAP Agreement Protocol Binding	106
Operation Bindings Tab	107
Editing an Operation Binding for the Host	108
Editing an Operation Binding for the Partner	111
Document Security Tab	115
Transports Tab.....	116

Overriding Participant Settings	117
Overriding Settings for the Host.	117
Overriding Settings for the Partner	117
Chapter 9 Viewing Logs	119
Overview	120
Accessing Logs	120
Audit Logs	121
Configuring an Audit Log for SOAP	121
Audit Log Views	122
Non-Repudiation Logs	127
Configuring a Non-Repudiation Log for SOAP	127
Resend Logs	129
Resendable Transactions	129
Resend History	130
Chapter 10 Advanced Topics	133
TIBCO-Specific Header: PartyInfo	134
Trading with a Third-Party SOAP Implementation	135
Receiving SOAP Messages without TIBCO BusinessConnect-Specific Headers	135
Sending SOAP Messages without TIBCO BusinessConnect-Specific Headers	136
Using the Passthrough Feature	137
TIBCO BusinessConnect SOAP Protocol Public Messages	140
SOAP Header	140
SOAP Body	140
SOAP Message with a TIBCO BusinessConnect-Specific Header.	142
SOAP Envelope Attributes and Namespaces	144
Envelope Attributes and Namespaces Example	144
Chapter 11 Private Messages	147
Overview	148
Initiator Messages	149
Initiator Outbound Request — Private Process to BusinessConnect	149
Initiator Inbound Response — BusinessConnect to Private Process	154
Responder Messages	157
Responder Inbound Request — BusinessConnect to Private Process	157
Responder Outbound Response — Private Process to BusinessConnect	161
Responder Acknowledgement — BusinessConnect to Private Process	165
Advisory Message	166
Attachment	168

Determining the Content Type of an Attachment	169
Outbound Attachments	169
Inbound Attachments	170
Preserving the SOAP 5.0 Attachment Functionality	171
Attachment Type	172
TradingPartner Type	173
Attributes	174
SOAPFault	175
WS-Addressing	176
XOP	178
AuthenticationToken	179
Appendix A Status Codes	181
statusCode and statusMsg Field Reference	182
Appendix B Schema Validation	191
Overview	192
Schema Validation Errors	193
Appendix C TIBCO BusinessConnect SOAP Protocol WSDL Tool	195
Overview	196
WSDL Import	197
WSDL Export	199
Index	201

Figures

Figure 1	Notification	11
Figure 2	Synchronous Request-Response.	13
Figure 3	Request Response Operation POSync	27
Figure 4	Imported Operations PONotify and POSync	31
Figure 5	Notify Message.	52
Figure 6	Imported Operations for SOAP	55
Figure 7	Send Notify.	59
Figure 8	Send Request and Receive Response	59
Figure 9	Receive Notification Message	60
Figure 10	Receive Request and Send Fault.	61
Figure 11	Receive Request and Send Response	62
Figure 12	Verifying the Name of the Trading Partner	66
Figure 13	Send Notify Process.	67
Figure 14	Configuring New Operation for SOAP	76
Figure 15	New Interface POInterface	77
Figure 16	New Interface Version	78
Figure 17	Importing Operations	90
Figure 18	Exporting SOAP Interface, Version, or Operation	91
Figure 19	Expanding the Interface	92
Figure 20	Notify Operation	93
Figure 21	Synchronous Request-Response Operation: Request Action	94
Figure 22	Synchronous Request-Response Operation: Response Action.	95
Figure 23	Operation Bindings Added for Host and Partner	108
Figure 24	Initiator Outbound Request	149
Figure 25	Initiator Inbound Response	154
Figure 26	Responder Inbound Request	157
Figure 27	Responder Outbound Response	161
Figure 28	Responder Acknowledgement	165

Tables

Table 1	General Typographical Conventions	xvi
Table 2	WS-Addressing	20
Table 3	New HTTP Transport	34
Table 4	Notify Operation Tab	80
Table 5	Notify Request Action Tab	81
Table 6	Synchronous Request-Response Operation Tab.	84
Table 7	Request Action Tab	84
Table 8	Response Action Tab	87
Table 9	Configuring a Host: General Tab	98
Table 10	Configuring a Partner: General Tab	100
Table 11	New Transport for a Partner	103
Table 12	Protocol Bindings Tab	107
Table 13	Override Outbound Settings: Operation Settings Tab	109
Table 14	Override Outbound Settings: Action Settings Tab	109
Table 15	Override Outbound Settings: Transports Tab	111
Table 16	Override Inbound Settings: Operation Settings Tab	112
Table 17	Override Inbound Settings: Action Settings Tab	112
Table 18	Transports Tab	116
Table 19	Show Advanced: Host's Configuration	117
Table 20	Show Advanced: Partner's Configuration	117
Table 21	Audit Log: Search Filters	121
Table 22	Audit Log: Advanced Search Filters	122
Table 23	Audit Log Columns	122
Table 24	Transaction Detail Columns	123
Table 25	State Field Values for Initiator Process	123
Table 26	State Field Values for Responder Process	125
Table 27	Non-Repudiation Log: Search Filters	127
Table 28	Non-Repudiation Log: Advanced Search Filters	128

Table 29	Resend Log: Resendable Transactions	129
Table 30	State Values for Resendable Transactions	130
Table 31	Resend Log: Advanced Search Filters	130
Table 32	Resend Log: Resend History	131
Table 33	Resend Log: Advanced Search Filters	131
Table 34	Initiator Request	149
Table 35	Initiator Response	154
Table 36	Responder Inbound Request	158
Table 37	Responder Outbound Response	161
Table 38	ResponderAck	165
Table 39	Advisory Messages	166
Table 40	Attachment Type	172
Table 41	Trading Partner Type	173
Table 42	Attribute Node	174
Table 43	SOAP Fault Node	175
Table 44	WS-Addressing Node	176
Table 45	The From, ReplyTo, and FaultTo Properties	176
Table 46	The RelatesTo Property	177
Table 47	XOP Node	178
Table 48	AuthenticationToken Node	179
Table 49	statusCode and statusMsg Fields	182
Table 50	BusinessConnect Database Configuration Properties for WSDL Import Process	198
Table 51	WSDL File, Trading Partner, Local Host Name, and SOAP Version Properties for WSDL Import Process 198	
Table 52	BusinessConnect Database Configuration Properties for WSDL Export Process	199
Table 53	WSDL File, Endpoint, SOAP Version, Service, Namespace, and List Properties for WSDL Export Process 199	

Preface

TIBCO BusinessConnect™ SOAP Protocol is a business protocol designed to enhance the capabilities of the TIBCO BusinessConnect™ integration platform. Integrated with TIBCO ActiveMatrix BusinessWorks™, you can use TIBCO BusinessConnect SOAP Protocol to create and send private process messages.

Topics

- [Related Documentation](#), page xiv
- [Typographical Conventions](#), page xvi
- [TIBCO Product Documentation and Support Services](#), page xviii

Related Documentation

This section lists documentation resources you may find useful.

TIBCO BusinessConnect SOAP Protocol Documentation

The following documents form the *TIBCO BusinessConnect SOAP Protocol* documentation set:

- *TIBCO BusinessConnect SOAP Protocol Installation* Read this manual for instructions on site preparation and installation.
- *TIBCO BusinessConnect SOAP Protocol User's Guide* Read this manual for instructions on using the product to manage SOAP operations between trading hosts and partners.
- *TIBCO BusinessConnect SOAP Protocol Release Notes* Read the release notes for a list of new and changed features. This document also contains lists of known issues and closed issues for this release.

Other TIBCO Product Documentation

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

- TIBCO BusinessConnect™
- TIBCO Administrator™
- TIBCO ActiveMatrix BusinessWorks™
- TIBCO Designer™
- TIBCO Runtime Agent™
- TIBCO Rendezvous®
- TIBCO Enterprise Message Service™
- TIBCO BusinessConnect™ Palette
- TIBCO ActiveMatrix BusinessWorks™ Plug-in for BusinessConnect

Third Party Documentation

You may find it useful to read the following third-party documentation:

- The specification for the SOAP protocol
<http://www.w3.org/TR/SOAP>
- Web Services Security UsernameToken Profile 1.1 OASIS Standard Specification, 1 February 2006
<http://docs.oasis-open.org/wss/v1.1/wss-v1.1-spec-os-UsernameTokenProfile.pdf>
- Web Services Addressing (WS-Addressing). W3C Member Submission, 10 August 2004
<http://www.w3.org/Submission/ws-addressing>
- SOAP Message Transmission Optimization Mechanism. W3C Recommendation, 25 January 2005
<http://www.w3.org/TR/2005/REC-soap12-mtom-20050125>
- Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0 OASIS Standard, 15 March 2005
<http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf>
- Authentication Context for the OASIS Security Assertion Markup Language (SAML) V2.0 OASIS Standard, 15 March 2005
<http://docs.oasis-open.org/security/saml/v2.0/saml-authn-context-2.0-os.pdf>
- Web Services Security: SAML Token Profile 1.1 OASIS Standard, 1 February 2006
<https://www.oasis-open.org/committees/download.php/16768/wss-v1.1-spec-os-SAMLTokenProfile.pdf>
- Web Services Security: SOAP Message Security 1.1 (WS-Security 2004) OASIS Standard Specification, 1 February 2006
<http://docs.oasis-open.org/wss/v1.1/wss-v1.1-spec-os-SOAPMessageSecurity.pdf>



TIBCO provides no warranties or assurances regarding the content located in the foregoing websites.

Typographical Conventions

The following typographical conventions are used in this manual.

Table 1 General Typographical Conventions

Convention	Use
<i>ENV_NAME</i>	TIBCO products are installed into an installation environment. A product installed into an installation environment does not access components in other installation environments. Incompatible products and multiple instances of the same product must be installed into different installation environments.
<i>TIBCO_HOME</i>	An installation environment consists of the following properties: <ul style="list-style-type: none"> • Name Identifies the installation environment. This name is referenced in documentation as <i>ENV_NAME</i>. On Microsoft Windows, the name is appended to the name of Windows services created by the installer and is a component of the path to the product shortcut in the Windows Start > All Programs menu. • Path The folder into which the product is installed. This folder is referenced in documentation as <i>TIBCO_HOME</i>.
<i>BCSOAP_HOME</i>	<i>TIBCO BusinessConnect SOAP Protocol</i> installs into a directory within a <i>TIBCO_HOME</i> . This directory is referenced in documentation as <i>BCSOAP_HOME</i> . The default value of <i>BCSOAP_HOME</i> depends on the operating system. For example, on Windows systems, the default value is C:\tibco\bc\version\protocols\soap.
code font	Code font identifies commands, code examples, filenames, pathnames, and output displayed in a command window. For example: Use MyCommand to start the foo process.
bold code font	Bold code font is used in the following ways: <ul style="list-style-type: none"> • In procedures, to indicate what a user types. For example: Type admin. • In large code samples, to indicate the parts of the sample that are of particular interest. • In command syntax, to indicate the default parameter for a command. For example, if no parameter is specified, MyCommand is enabled: MyCommand [enable disable]

Table 1 General Typographical Conventions (Cont'd)

Convention	Use
<i>italic font</i>	<p>Italic font is used in the following ways:</p> <ul style="list-style-type: none"> • To indicate a document title. For example: See <i>TIBCO BusinessConnect Concepts</i>. • To introduce new terms. For example: A portal page may contain several portlets. <i>Portlets</i> are mini-applications that run in a portal. • To indicate a variable in a command or code syntax that you must replace. For example: <code>MyCommand <i>pathname</i></code>
Key combinations	<p>Key name separated by a plus sign indicate keys pressed simultaneously. For example: Ctrl+C.</p> <p>Key names separated by a comma and space indicate keys pressed one after the other. For example: Esc, Ctrl+Q.</p>
	The note icon indicates information that is of special interest or importance, for example, an additional action required only in certain circumstances.
	The tip icon indicates an idea that could be useful, for example, a way to apply the information provided in the current section to achieve a specific result.
	The warning icon indicates the potential for a damaging situation, for example, data loss or corruption if certain steps are taken or not taken.

TIBCO Product Documentation and Support Services

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

How to Access TIBCO Documentation

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

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

Documentation for TIBCO BusinessConnect SOAP Protocol is available on the <https://docs.tibco.com/products/tibco-businessconnect-soap-protocol> Product Documentation page.

How to Contact TIBCO Support

You can contact TIBCO Support in the following ways:

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

How to Join TIBCO Community

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

Chapter 1

Introduction to SOAP and TIBCO BusinessConnect SOAP Protocol

This chapter gives an overview of SOAP and TIBCO BusinessConnect SOAP Protocol.

Topics

- [SOAP Overview, page 2](#)
- [TIBCO BusinessConnect SOAP Protocol Overview, page 3](#)
- [TIBCO BusinessConnect SOAP Protocol Features, page 4](#)

SOAP Overview

TIBCO BusinessConnect SOAP Protocol supports both SOAP versions 1.1 and 1.2.

SOAP 1.1 is a lightweight XML-based messaging protocol for exchanging structured data in a decentralized, distributed environment. Buyers, sellers, and intermediaries can use SOAP to share business documents and messages over the Internet. SOAP can also be used for other types of supply chain integration transactions, such as collaborative forecasting, inventory management, and design collaboration.

SOAP 1.2 is an update version based on SOAP 1.1.

SOAP transactions involve the exchange of documents, most of which are analogous to hardcopy documents traditionally used in business. These documents are simple text files, but they have well-defined structure and contents.

SOAP has the following parts:

- An envelope that defines a framework for describing what is in a message and how to process the message. This defines the message package. The SOAP envelope consists of an optional header and a mandatory body. The envelope is the first element in the document and identifies it as a SOAP message. The sender can add management or control information that can be used for routing, security, or correct handling by the recipient. The body contains the information sent to the receiver.
- SOAP attachments which follow the SOAP with Attachment (SwA) standard. Businesses use SOAP attachments to exchange business documents in arbitrary formats, such as EDI, PDF, and JPG to facilitate business-to-business transactions. SwA is also used with MTOM to carry binary data originally contained in XML infoset.
- A standard for representing request and response.

TIBCO BusinessConnect SOAP Protocol Overview

TIBCO BusinessConnect SOAP Protocol is the implementation by TIBCO of SOAP 1.1 and SOAP 1.2 specifications. The protocol has been developed by TIBCO for exchanging any type of business documents used in electronic business, for example, XML, EDI, and other binary format, such as PDF, JPG, and so on. Based on an agreed-upon process flow and common document format, you and your trading partner can conduct secure and verifiable business transactions online by using TIBCO BusinessConnect SOAP Protocol.

See the information about business protocols in the chapter that explains Operations in *TIBCO BusinessConnect Concepts*.

TIBCO BusinessConnect SOAP Protocol Features

The following are some significant features of the TIBCO BusinessConnect SOAP Protocol:

- Support for the synchronous Request-Response and Notify operation types.
See [Notify Operations on page 10](#) and [Synchronous Request-Response Operations on page 11](#).
- Support for SOAP versions 1.1 and 1.2.
See [SOAP Overview on page 2](#) and [SOAP Version on page 101](#).
- Support for SOAP fault in versions 1.1 and 1.2.
See [SOAP Fault on page 23](#).
- Support for Message Transmission Optimization Mechanism (MTOM).
See [MTOM on page 22](#) and [SOAP MTOM Enabled on page 102](#).
- Support for HTTP, HTTPS, and HTTPSCA transport protocols.
- Multiple server certificate support for HTTPS.
- Access control through trading partner identification and permissions.
- Validation of XML documents with XSDs.
- Signing or encrypting the SOAP body as a whole by using Web Services Security (WSS).
See [SOAP Message Security on page 17](#).
- Support for Security Assertion Markup Language (SAML) 2.0 in SOAP header for digital signature and authorization in conformance with NHIN/esMD Authorization Framework.
See [NHIN/esMD Authorization Framework with SAML Assertions on page 19](#).
- Support for signature confirmation in a response to a specific request.
- Support for WSS UsernameToken authentication.
See [SOAP Message Security on page 17](#).
- Limited support for Web Services Addressing (WS-Addressing) in SOAP header.
See [WS-Addressing on page 20](#).
- Ability to specify certain timing constraints.
- Ability to generate audit records.

See [Audit Logs](#) on page 121.

- Support for operation setting overriding in a specific business agreement.

See [Editing an Operation Binding for the Host](#) on page 108 and [Editing an Operation Binding for the Partner](#) on page 111.

- Support for non-repudiation logging for all operations.

See [Non-Repudiation Logs](#) on page 127.

- Ability to exchange business documents using SwA.

See [Attachment](#) on page 168.

- Support for Web Services Description Language (WSDL). WSDL is an XML-formatted language used to describe the capabilities of a Web service as collections of communication endpoints capable of exchanging messages. TIBCO BusinessConnect SOAP Protocol configurations and operation definitions can be exported to WSDL files, and vice versa.

- Tools available to export and import configurations and operation definitions as WSDL files.

See [TIBCO BusinessConnect SOAP Protocol WSDL Tool](#) on page 195.

Chapter 2 **Process Flows and Functionalities**

This chapter describes process flows and relevant functionalities by using TIBCO BusinessConnect SOAP Protocol for business transactions.

Topics

- [TIBCO BusinessConnect SOAP Protocol Messages, page 8](#)
- [Operation Types and Process Flows, page 10](#)
- [Duplicate Message Detection, page 15](#)
- [SOAP Message Security, page 17](#)
- [NHIN/esMD Authorization Framework with SAML Assertions, page 19](#)
- [WS-Addressing, page 20](#)
- [MTOM, page 22](#)
- [SOAP Fault, page 23](#)

TIBCO BusinessConnect SOAP Protocol Messages

In a SOAP transaction, two partners exchange business documents over the Internet based on a pre-defined business agreement. Among other options, the business agreement describes what message formats and transport protocols the partners have agreed to use. The exchange of business documents is known as the process flow. See [Operation Types and Process Flows on page 10](#).

In a TIBCO BusinessConnect process flow, two types of messages are exchanged: private messages and public messages.

Private Messages and Processes

Private messages are exchanged between TIBCO BusinessConnect and the private processes, which are the interface for TIBCO BusinessConnect to integrate with the back-office system. An example of private process is the TIBCO ActiveMatrix BusinessWorks process. For a detailed description of TIBCO BusinessConnect SOAP Protocol private messages, see [Chapter 11, Private Messages, page 147](#). Private messages can contain a request, a response, or a notification document.

Private processes handle conversion from company back-office data to TIBCO BusinessConnect private messages.

- On the initiator side, the private process converts internal company back-office data to TIBCO BusinessConnect private message. This message represents a request, an acceptance, or a notification document.
- On the responder side, the private process receives a private message from TIBCO BusinessConnect, converts it to internal company back-office format and sends to the back-office system. If this is a request-response operation, the private process receives back a response from inside the company, and converts it to a response private message, and sends it back to TIBCO BusinessConnect.

The following private processes types are available with TIBCO BusinessConnect SOAP Protocol:

- **Standalone** Standalone private processes use TIBCO Rendezvous Certified Messaging to communicate with TIBCO BusinessConnect SOAP Protocol. They are normally implemented by using any programming language that supports TIBCO Rendezvous API. For an example, see [Chapter 3, Standalone Private Processes Tutorial, on page 25](#).
- **TIBCO ActiveMatrix BusinessWorks** The TIBCO ActiveMatrix BusinessWorks processes combined with TIBCO BusinessConnect Palette can be used to send requests to TIBCO BusinessConnect SOAP Protocol, or receive replies from TIBCO

BusinessConnect SOAP Protocol. For an example, see [Chapter 4, TIBCO ActiveMatrix BusinessWorks Private Processes Tutorial](#), on page 49.

Public Messages

Public messages are exchanged over the Internet between two trading partners. TIBCO BusinessConnect SOAP Protocol can be used by an enterprise to send and receive public messages to do transactions with its trading partners. When trading partners use SOAP protocol to do transaction with each other, the public messages exchanged between them are SOAP messages. They can use HTTP, SMTP, FTP, or other transport protocols for the messages exchange. TIBCO BusinessConnect SOAP Protocol supports HTTP and HTTPS as transport protocols. Synchronous request-response and notify transaction types are supported.

- **On the Initiator side** The initiator BusinessConnect server receives requests as private messages from private processes, converts them into SOAP messages and sends them to the responder side. In request-response transaction mode, the initiator BusinessConnect server receives response SOAP messages from the responder side, converts them into private messages that are eventually sent to private processes.
- **On the Responder side** The responder BusinessConnect server receives SOAP request messages from the initiator, which is on the trading partner side, and then converts the SOAP messages into private messages that are sent to private processes. In request-response transaction mode, the responder BusinessConnect server receives the response private messages from the responder private processes, and then converts them into SOAP response messages and then replies back to the initiator side.

Operation Types and Process Flows

When you use TIBCO BusinessConnect to exchange SOAP messages with a partner, you can use different operation types for different kinds of transactions. Two operation types are in SOAP. Any operation defined must be one of these two operation types:

- **Notify** Use this operation type when the initiator only requires an acknowledgment from the responder to verify the receipt of the SOAP message.
- **Synchronous request-response** Use this operation type when the initiator and the responder do a more complex exchange of business documents by exchanging SOAP messages. When the initiator sends a request SOAP message to the responder, a response SOAP message is expected to be returned back from the responder in synchronous HTTP reply.

Notify Operations

The following process flow occurs in a Notify type operation. In this process flow, both the initiator and responder are assumed to be implemented by TIBCO BusinessConnect SOAP Protocol:

1. A private process inside the initiating company sends a request private message that contains the request business document to the initiator BusinessConnect server. See [Initiator Outbound Request — Private Process to BusinessConnect on page 149](#).
2. The initiator BusinessConnect server retrieves relevant information from the private message, packages the message into an appropriate SOAP message and sends the message to the responder.
3. The responder BusinessConnect server receives the SOAP message, validates, unpackages, retrieves the business document, and then forwards the document in private message format to the responder private process. See [Responder Inbound Request — BusinessConnect to Private Process on page 157](#).
4. The responder BusinessConnect server immediately sends a response (acknowledgment) to the initiator on the same HTTP transport channel as the business request of initiator.
5. Alternatively, if the responder BusinessConnect server cannot process the notify request SOAP message, the responder BusinessConnect server sends a SOAP fault to the initiator on the same HTTP transport channel.

Time Constrains and Resend

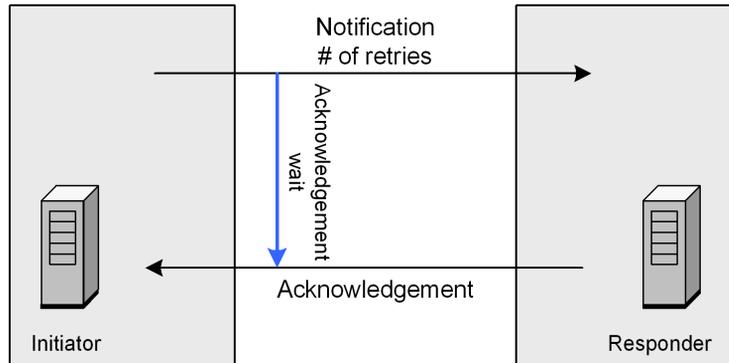
For a Notify type operation, you can specify the time constrains on the initiator side:

For the initiator: configure in HTTP/HTTPS transport configuration for the specific trading partner.

- Retry Count
- Retry Interval
- Socket Timeout (seconds)

The flow of information and the associated predefined timing restrictions are as shown in [Figure 1](#).

Figure 1 Notification



As the Initiator

From the initiator, an operation is processed with the following timing restrictions:

1. The initiator sends out a request to the responder.
2. If communication cannot be established, a specified number of retries occur with specified time intervals between retransmits.
3. If the communication is established and a request is sent over, however, the acknowledgment does not come back within the specified Socket Timeout, an error is generated on the initiator side and sent to the private process.

Synchronous Request-Response Operations

The following process flow occurs in a synchronous request-response operation. In this process flow, both the initiator and responder are assumed to be implemented by TIBCO BusinessConnect SOAP Protocol:

1. A private process inside the initiating company sends a request private message to the initiator BusinessConnect server, which contains the request business document. See [Initiator Outbound Request — Private Process to BusinessConnect](#) on page 149.

2. The initiator BusinessConnect server retrieves relevant information from the private message, packages it into a request SOAP message, and sends the SOAP message to the responder.
3. The responder BusinessConnect server receives the SOAP message, validates, unpackages, retrieves the business document, and then forwards it to the responder private process and waits for the private process to respond. See [Responder Inbound Request — BusinessConnect to Private Process on page 157](#).

If the validation or unpackaging of the inbound SOAP message fails, the responder BusinessConnect server sends a SOAP fault on the same HTTP transport channel to the initiator BusinessConnect server, which forwards the fault content to the private process.

4. The responder private process responds to the responder BusinessConnect server with a response private process message containing the response business document. See [Responder Outbound Response — Private Process to BusinessConnect on page 161](#).

The responder private process can also return an error message as a SOAP fault to responder BusinessConnect server if the private process cannot process the request SOAP message. To send a SOAP fault as the response on the responder side, ensure that the value in the **statusCode** field of the `ae/SOAP/ResponderResponse` AE message is not in the range of 200-299.

The `ae/SOAP/ResponderResponse` AE message class in the SOAP fault private message includes a field called **soapFault**. TIBCO BusinessConnect SOAP Protocol uses the values in the **soapFault** field to generate the public SOAPFault response message.

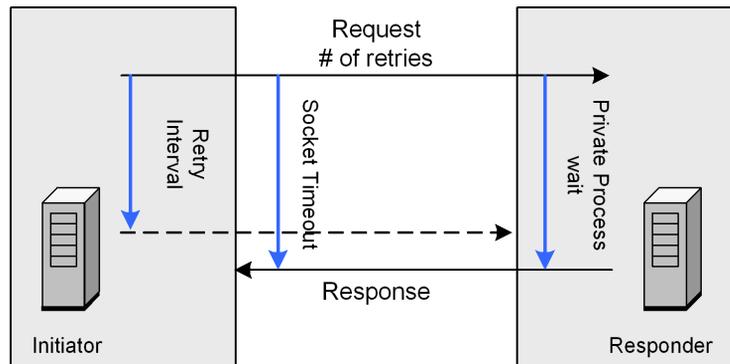
5. The responder BusinessConnect server forwards the business response or a SOAP fault on the same HTTP transport channel to the initiator BusinessConnect server.
6. The responder BusinessConnect server sends an acknowledgment to the responder private process if no error occurs during the reply of the response message. See [Responder Acknowledgement — BusinessConnect to Private Process on page 165](#).
7. The initiator BusinessConnect server forwards the response or the SOAP fault to the initiator private process. See [Initiator Inbound Response — BusinessConnect to Private Process on page 154](#).

Time Constrains and Resend

For a synchronous request-response operation, you can specify the following time constrains:

- For the initiator: configure in HTTP or HTTPS transport configuration for the specific trading partner.
 - Retry Count
 - Retry Interval
 - Socket Timeout (seconds)
- For the responder: configure in the response action of the specific operation definition.
 - Private Process Wait (seconds)

Figure 2 Synchronous Request-Response



As the Initiator

From the initiator, an operation is processed with the following timing restrictions:

1. The initiator sends out a request to the responder.
2. If communication cannot be established, a specified number of retries occurs with specified time intervals between retransmits.
3. If the response does not come back within the specified Socket Timeout, an error is generated on the initiator side and sent to the private process.

As the Responder

From the responder, an operation is processed with the following timing restrictions:

1. The responder receives a request SOAP message from the initiator.
2. The responder BusinessConnect server hands off the request business document to the responder private process.
3. The responder private process returns the response business document to the responder BusinessConnect server. The maximum wait time between the request

business document sent to the private process and the response from the private process can be specified as Private Process Wait. If the response message is not received within the Private Process Wait time, an error is generated by the responder BusinessConnect server and a SOAP fault is returned back to the initiator.

4. After the responder BusinessConnect server receives the response from the private process, the response is packaged into a SOAP message and sent back to the initiator.
5. Unless a transport-level error is returned, the responder BusinessConnect server sends an acknowledgment to the responder private process indicating that the message has been replied to the initiator.

Duplicate Message Detection

TIBCO BusinessConnect SOAP Protocol checks for duplicates on all inbound and outbound messages. If duplicates are found, the duplicate field value in the request or response message sent to a private process is set to `true`.

Inbound Messages

For inbound messages, duplicate detection is performed in the same way for the inbound request and inbound response.

Duplicate detection criteria for the inbound request and inbound response messages are as follows:

- Protocol name
- Protocol version
- Installation name
- Host name
- Trading partner name
- Operation ID
- Body element of the SOAP envelope

Outbound Messages

For outbound messages, duplicate detection is performed only for the outbound request received from a private process.

Duplicate detection criteria for the OutboundRequest messages are as follows:

- Protocol name
- Protocol version
- Installation name
- Host name
- Trading partner name
- Transaction ID
- Operation ID
- Request coming from the private process (no header)



The domains of the host and trading partner in private process request messages are not taken into consideration when performing duplicate detection. Only the values `hostname` and `tpname` are considered.

SOAP Message Security

TIBCO BusinessConnect SOAP Protocol supports the following security features:

- [Digital Signature and Encryption](#)
- [UsernameToken for Authentication](#)

Digital Signature and Encryption

In TIBCO BusinessConnect SOAP Protocol, Web Services Security (WSS) is used to sign or to encrypt the SOAP body as a whole. Individual elements of the body or any other parts, such as header elements or attachments, cannot be signed or encrypted.

For more details on how to sign and encrypt, see the **Require Digital Signature** and **Require Content Encryption** for the following operations:

- [Notify Request Action Tab on page 81](#)
- [Request Action Tab on page 84](#)
- [Response Action Tab on page 87](#)



TIBCO BusinessConnect SOAP Protocol support DES3, AES-128, AES-192, AES-256, AES-128-GCM, AES-192-GCM, and AES-256-GCM encryption algorithms and the SHA1, SHA256, SHA384 and SHA512 digest algorithm for signatures.

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 UsernameToken only digest password in the current implementation for outbound messages. For inbound messages, either the digest or plain text passwords are accepted. Plain text password is normally used together with SSL transport.

When UsernameToken is used for authentication, the `wssc:UsernameToken` element is in the “primary” security header along with other security entities such as digital signature and encryption element if they are present.

You can set TIBCO BusinessConnect SOAP Protocol to use the UsernameToken for authentication in TIBCO Administrator by clicking **BusinessConnect > Operations Editor > SOAP > Operation Type > Request/Response Action > General > Require UsernameToken Authentication**.

If you select the **Require UsernameToken Authentication** check box on the outbound side, the user name and password must be typed in the private process.

If you select the **Require UsernameToken Authentication** check box on the inbound side, the inbound request and response SOAP messages are required to embed the UsernameToken element, and the user has to be defined as an external user and be associated with this trading partner from whom the message is received. You can define and configure an external user in TIBCO Administrator by clicking **BusinessConnect > User Management > Users > External**.

The UsernameToken for Authentication is not supported for SOAP fault messages.

For more details, see the **Require UsernameToken Authentication** check box for the following operations:

- [Notify Request Action Tab on page 81](#)
- [Request Action Tab on page 84](#)
- [Response Action Tab on page 87](#)

Sample of the UsernameToken Part

```
<wsse:UsernameToken xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
  <wsse:Username>Test</wsse:Username>
  <wsse:Password
Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.0#PasswordDigest">NdiRI/Z4CnWfbFah1BFWxlt7xLM
=</wsse:Password>
  <wsse:Nonce
EncodingType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0#Base64Binary">Prvav23eGuMul/XSIGI07A=
=</wsse:Nonce>
  <wsu:Created
xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">2014-01-13T03:14:56.640Z</wsu:Created>
</wsse:UsernameToken>
```

NHIN/esMD Authorization Framework with SAML Assertions

SAML is an XML-based, open-standard data format for exchanging authentication and authorization data between parties. SAML is a product of the OASIS Security Services Technical Committee. The Nationwide Health Information Network (NHIN) is a set of standards, services, and policies for secure exchange of the health information over the Internet. The NHIN Authorization Framework Specification defines the primary set of services and protocols that is required to establish a messaging, security, and privacy foundation for the NHIN.

Electronic Submission of Medical Documentation (esMD), as part of NHIN, follows the NHIN Authorization Framework with SAML 2.0 assertions. By using SAML assertions, esMD defines the exchange of metadata used to characterize the initiator of an esMD request, so the request might be evaluated by the esMD Gateway in local authorization decisions. The purpose of these SAML assertions exchange is to provide the esMD Gateway with the required information to make an authorization decision by using the policy enforcement point for the requested esMD function. Each initiating SOAP message must convey information by using SAML 2.0 assertions. The NHIN/esMD Authorization Framework is used by Council for Affordable Quality Healthcare (CAQH) CORE in exchanging healthcare information.

TIBCO BusinessConnect SOAP Protocol supports SAML assertions in conformance with the NHIN Authorization Framework. TIBCO BusinessConnect SOAP Protocol can work as either the esMD Gateway or clients. On esMD Gateway side, requests with SAML assertions are authenticated before all the Health Information Handler (HIH) and authorization attributes are forwarded to back office systems for further evaluation in authorization decisions. Responses sent back to the initiators in asynchronous mode can use SAML assertion signature, and also convey the HIH and authorization attributes. Responses sent back to the initiators in synchronous mode can use signature confirmation to verify the SAML assertion signature.

For more details on how to configure SAML assertions in TIBCO Administrator, see [Request Action Tab on page 84](#) and [Response Action Tab on page 87](#). For more details on the NHIN/esMD process definitions, see [Introducing NHIN/esMD Process Definitions on page 62](#).



SAML 2.0 assertions only support Secure Hash Algorithm 1 (SHA1) digest algorithm.

WS-Addressing

Web Services Addressing (WS-Addressing) is a specification of transport-neutral mechanism that specifies the endpoints of a SOAP message.

TIBCO BusinessConnect SOAP Protocol supports WS-Addressing partially in the SOAP header. When TIBCO BusinessConnect SOAP Protocol processes the inbound messages that contain the following seven properties, these properties are forwarded in the messages sent to a private process.

Table 2 WS-Addressing

Field	Type	Description
MessageID	String	The unique URI of the message.
To	String	Specifies the receiver URI of the message.
From	N/A	Specifies the endpoint of the service that dispatched the message.
ReplyTo	N/A	Specifies the endpoint of the receiver for reply messages.
FaultTo	N/A	Specifies the endpoint of the receiver for fault messages.
Action	String	The specific semantic operation of the message.
RelatesTo	N/A	A pair of values that specify the relationship between this message and another message. The original request MessageID must correspond with this property.



- If MessageID is not specified from the private process, or from any inbound request or response, all the properties are ignored.
- In the response message replied back to the initiator, the To property is set to the same as the ReplyTo property of the original request message. However, if the private process specifies the WS-Addressing To property for the response message, it is used to override the ReplyTo property in the original request message.
- The RelatesTo property of the response message replied to the initiator is set to the same as the MessageID property of the original request message.

Samples of the Messages by Using WS-Addressing

Here is a sample of a request message:

```

<S:Envelope
  xmlns:S="http://www.w3.org/2003/05/soap-envelope"
  xmlns:wsa="http://www.w3.org/2005/08/addressing">
  <S:Header>
    <wsa:MessageID>http://example.com/someuniquestring</wsa:MessageID>
    <wsa:ReplyTo>
      <wsa:Address>http://example.com/business/client1</wsa:Address>
    </wsa:ReplyTo>
    <wsa:To>mailto:fabrikam@example.com</wsa:To>
    <wsa:Action>http://example.com/fabrikam/mail/Delete</wsa:Action>
  </S:Header>
  <S:Body>
    <f>Delete xmlns:f="http://example.com/fabrikam">
      <maxCount>42</maxCount>
    </f.Delete>
  </S:Body>
</S:Envelope>

```

Here is a sample of a response message:

```

<S:Envelope
  xmlns:S="http://www.w3.org/2003/05/soap-envelope"
  xmlns:wsa="http://www.w3.org/2005/08/addressing">
  <S:Header>
    <wsa:MessageID>http://example.com/someotheruniquestring</wsa:MessageID>
    <wsa:RelatesTo>http://example.com/someuniquestring</wsa:RelatesTo>
    <wsa:To>http://example.com/business/client1</wsa:To>
    <wsa:Action>http://example.com/fabrikam/mail/DeleteAck</wsa:Action>
  </S:Header>
  <S:Body>
    <f>DeleteAck xmlns:f="http://example.com/fabrikam"/>
  </S:Body>
</S:Envelope>

```

MTOM

TIBCO BusinessConnect SOAP Protocol supports MTOM standard for optimizing the transmission of binary data in XML infoset.

The business document in a request or response might contain large binary data encoded in base64. MTOM is a standard to optimize this type of data, because the base64 encoded data might be much larger than the original binary data. Once this standard is applied, the binary element in the XML infoset is externalized into a separate SOAP attachment.

In the current implementation, you can enable or disable MTOM feature in TIBCO Administrator. You can configure each trading partner separately. Only the binary element in the XML document specified in the `body` node of the request private process is processed by using MTOM standard. You can provide a value in the `xop > tagName > value` field to indicate which binary data element in the XML document to be processed by MTOM standard.

For inbound messages, if optimized messages by MTOM are detected, the messages are parsed and the original binary data extracted from the MTOM attachments will be encoded in base64 and put back into the SOAP body, before the SOAP message is further processed.

MTOM and attachments from a private process cannot be used together. So if the MTOM is enabled for outbound message and in the meanwhile some attachments are also specified from a private process request message, the request will be rejected and the outbound process will not proceed.

MTOM can also be used together with content encryption. If the XML data is encrypted for outbound message, the binary content referred to the value in the `xop > tagName > value` field in the private process message is ignored. The encrypted data in the `CipherValue` element after encryption is used to construct the MTOM attachment.

See [Configuring General Properties for a Partner on page 100](#) on how to select SOAP MTOM.

SOAP Fault

TIBCO BusinessConnect SOAP Protocol supports SOAP fault in both versions 1.1 and 1.2. For inbound messages, when an exception or error occurs, TIBCO BusinessConnect SOAP Protocol generates a SOAP fault in the same version as the incoming message, and sends back the SOAP fault to the trading partner.



For TIBCO BusinessConnect SOAP Protocol, the TIBCO ActiveMatrix BusinessWorks private process supports the SOAP fault only in version 1.2.

- On BusinessConnect responder side, when an error occurs in the back office system, a SOAP fault is sent back. The responder private process implemented by TIBCO ActiveMatrix BusinessWorks only supports sending version 1.2 SOAP fault to the responder BusinessConnect server, regardless of the SOAP version of the original request message. If the original request message is a version 1.1 SOAP message, the responder BusinessConnect server converts the version 1.2 SOAP fault into a version 1.1 SOAP fault and sends it back to the initiator. Because of the different fault structure in versions 1.1 and 1.2, only the first one of the fault reasons in version 1.2 is extracted and used in the version 1.1 SOAP fault.
- On BusinessConnect initiator side, if the initiator receives a SOAP version 1.1 fault for whatever reason, BusinessConnect server converts this SOAP fault to version 1.2 before it is forwarded to the private process.



For TIBCO BusinessConnect SOAP Protocol, the TIBCO ActiveMatrix BusinessWorks private process only supports SOAP fault in version 1.2 with BusinessConnect. If you want to use SOAP fault in version 1.1, you must do the conversion yourself.

If you use TIBCO ActiveMatrix BusinessWorks to convert when you send a SOAP fault, perform the following steps:

1. Populate the value in the **soapFault > Code > Value** field as your `faultCode`.
2. Populate the value in the **soapFault > Reason > ReasonText > content** field as your `faultString`.
3. Populate the value in the **soapFault > Role** field as your `faultActor`.

After performing these steps, the SOAP fault you send from TIBCO ActiveMatrix BusinessWorks private process to BusinessConnect is in version 1.2. BusinessConnect automatically converts the SOAP fault back to version 1.1 before the SOAP fault is sent to a trading partner if the trading partner wants the SOAP fault in version 1.1.

Sample of a SOAPFault in Version 1.1

```

<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <SOAP-ENV:Fault>
      <faultcode>SOAP-ENV:Server</faultcode>
      <faultstring>Server Error</faultstring>
      <faultactor>SOAPClient</faultactor>
      <detail>
        <ei:ErrorInfo xmlns:ei="http://www.tibco.com/
namespaces/bc/2002/04/errorinfo.xsd">
          <code>922</code>
          <description>Actual error from schema validation</description>
        </ei:ErrorInfo>
      </detail>
    </SOAP-ENV:Fault>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

Sample of a SOAPFault in Version 1.2

```

<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope"
  xmlns:m="http://www.example.org/timeout"
  xmlns:xml="http://www.w3.org/XML/1998/namespace">
  <env:Body>
    <env:Fault>
      <env:Code>
        <env:Value>env:Sender</env:Value>
        <env:Subcode>
          <env:Value>m:MessageTimeout</env:Value>
        </env:Subcode>
      </env:Code>
      <env:Reason>
        <env:Text xml:lang="en">Sender Timeout</env:Text>
      </env:Reason>
      <env:Detail>
        <m:MaxTime>P5M</m:MaxTime>
      </env:Detail>
    </env:Fault>
  </env:Body>
</env:Envelope>

```

Standalone Private Processes Tutorial

This chapter guides you through the steps necessary for running a TIBCO BusinessConnect SOAP Protocol business-to-business transaction.

The example in this chapter uses standalone private processes.

Topics

- [Overview, page 26](#)
- [Prerequisites, page 29](#)
- [Configuring the Initiator, page 30](#)
- [Configuring the Responder, page 36](#)
- [Running the Tutorial, page 41](#)

Overview

This chapter provides a short tutorial that demonstrates how to use an operation to send a sample document from a trading host to a trading partner.

Operations

Two operations are defined in the sample interface file:

- **PONotify** This operation is used only when the initiator requires an acknowledgment from the responder to verify business document receipt.
- **POSync** This is a synchronous request-response operation that is used when the initiator and the responder do a more complex business exchange of business documents.

Only the `POSync` operation is used in this tutorial.

Trading Partners

The trading host is known as the *initiator*. The trading partner is known as the *responder*.

This tutorial includes the sample initiator, responder and operation configuration files along with the standalone initiator and responder private processes.

Two machines are used in the tutorial: an initiator machine and a responder machine. Each machine hosts a private process and BusinessConnect.

Transactions

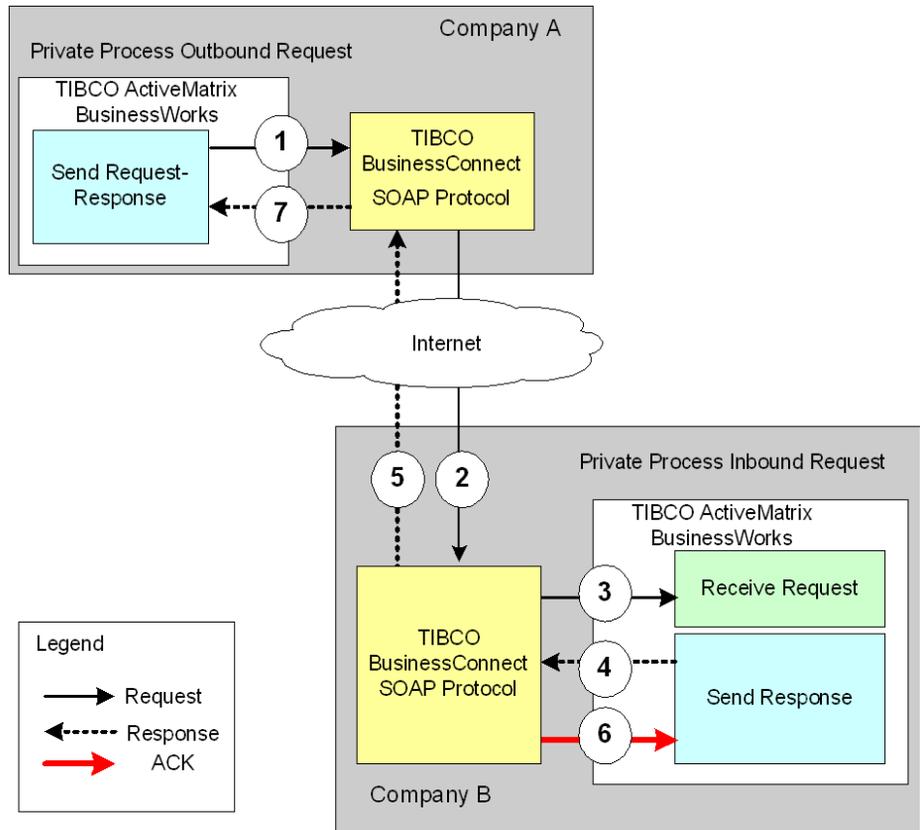
The `POSync` operation consists of the following message flows:

1. The initiator private process sends a message to the initiator BusinessConnect server.
2. The initiator BusinessConnect server sends the message to the responder BusinessConnect server.
3. The responder BusinessConnect server sends the message to the responder private process.
4. The responder private process sends a response to the responder BusinessConnect server.
5. The responder BusinessConnect server sends a response to the initiator BusinessConnect server.

6. The responder BusinessConnect server sends an acknowledgement (ack) to the responder private process.
7. The initiator BusinessConnect server sends a response to the initiator private process.

A detailed diagram of the POSync operation is displayed in [Figure 3](#).

Figure 3 Request Response Operation POSync



Initiator Machine

The following sample files for the initiator machine are used in the tutorial and are available in the *TIBCO_HOME*/bc/*version_number*/protocols/soap/samples/client directory.

- SOAPClient.properties Contains information on the operation ID, trading partner, and attachments.
- runSOAPClient.bat OR runsoapclient Executes the initiator.
- operations.csx Contains sample SOAP operations to be imported by the initiator.

Responder Machine

The following sample files for the responder machine are used in the tutorial and are available in the *TIBCO_HOME*/bc/*version_number*/protocols/soap/samples/server directory.

- SOAPServer.properties Contains information on attachments.
- runSOAPServer.bat OR runsoapservice Executes the responder.
- operations.csx Contains sample SOAP operations to be imported by the responder.

Prerequisites

Before starting this tutorial, perform the following prerequisites:

1. Install the following software packages:
 - a. TIBCO BusinessConnect
 - b. TIBCO BusinessConnect SOAP Protocol
2. If you are unfamiliar with the SOAP standard, see [Chapter 1, Introduction to SOAP and TIBCO BusinessConnect SOAP Protocol, page 1](#).
3. See *TIBCO BusinessConnect Interior Server Administration* and *TIBCO BusinessConnect Trading Partner Administration* for complete information on setting up and running BusinessConnect.

Configuring the Initiator

This section guides you through the activities you must perform to configure the initiator trading host:

1. [Importing Operations on the Initiator, page 30](#)
2. [Setting Up the Initiator Server, page 31](#)
3. [Setting Up the Initiator as a Trading Host, page 32](#)
4. [Setting Up the Responder as a Trading Partner, page 33](#)
5. [Configuring the Business Agreement between the Initiator and the Responder, page 35](#)

Importing Operations on the Initiator

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

1. Log in to TIBCO Administrator.
2. Expand **BusinessConnect > Operations Editor** in the left panel.
3. Click **Import**.
4. Click **change**.
5. Click **Browse** and navigate to the `TIBCO_HOME/bc/version_number/protocols/soap/samples/client` directory, and select the `operations.csx` file.

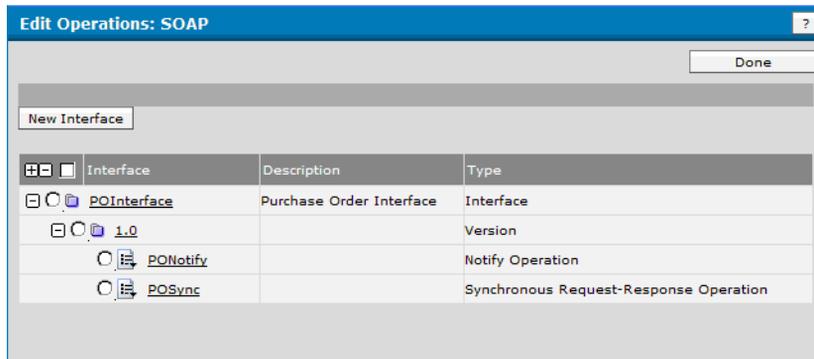


If you want to use NHIN/esMD Authorization Framework with SAML assertions, import the `caqh.csx` file in the `TIBCO_HOME/bc/version_number/protocols/soap/samples/caqh` directory. This file contains several sample operations that might help you understand how to use this feature.

6. Click **Open**.
7. Click **OK**.
8. Type a password. This is optional.
9. Click **Import**.

The Edit Operations window is displayed with imported operations, as shown in [Figure 4](#).

Figure 4 Imported Operations POnotify and POSync



10. Click **Done**.

Setting Up the Initiator Server

You must set up the initiator server to communicate with its trading partners. To set up the initiator, perform the following steps:

1. Create the deployment configuration.
See *TIBCO BusinessConnect Interior Server Administration*.
2. [Setting Up the Initiator Server Transport](#).

Setting Up the Initiator Server Transport

To configure the BusinessConnect server, perform the following steps:

1. In TIBCO Administrator, expand **BusinessConnect > System Settings** in the left panel.
2. Click **Inbound Public Transport Types** in the right panel.
3. Select the **HTTP**, **HTTPS** and **HTTPSCA** check boxes, and click **Enable**.
4. Click **Done**.
5. Expand **BusinessConnect > Gateway** in the left panel.
6. Click **Gateway Services** in the right panel.
7. Click **New**.
8. Type **HTTP** in the **Name** field, and select **HTTP** from the **Type** list.
9. Click **OK**.
10. In the **General** tab, select the **Active** check box.

11. In the **Credentials** tab, Click **New Private Key**.
12. Type **Host Key** in the **Alias** field.
13. Click **change** link.
14. Click **Browse** and navigate to the following directory:
TIBCO_HOME/bc/version_number/samples/keys.
15. Select `bcpartner1_key.p12` and click **Open**.
16. Click **OK**.
17. Click **set** next to **Password**.
18. Type **Password1** in both the **Enter password** and **Enter password again** fields.
19. Click **OK**.
20. Click **Save**.
21. Click the **Transport** tab, select **Host Key** from the **Private Key Credential for Secure Ports** list.
22. Click **Save**.

TIBCO BusinessConnect on the initiator side is configured with HTTP and HTTPS transports.

Setting Up the Initiator as a Trading Host

On the initiator machine, set up the initiator as a trading host.

The trading host setup for the initiator consists of these steps:

1. [Setting Up the Initiator Host, page 32](#)
2. [Setting Up the SOAP Protocol for the Initiator Host, page 33](#)

See [Setting Up the Responder as a Trading Partner on page 33](#).

Setting Up the Initiator Host

The trading host is typically defined when setting up TIBCO BusinessConnect.

If the host is set, go to [Setting Up the SOAP Protocol for the Initiator Host on page 33](#). If the host is not set, perform the following steps:

1. In TIBCO Administrator, expand **BusinessConnect** > **Participants** in the left panel.
2. Click **New**.
3. Type `SOAPClient` in the **Name** field.
4. Select **Host** from the **Type** list.

5. Click **OK**.
6. In the New Host Participant window with the participant name SOAPClient, select the **Active** check box.
7. Click **Save**.
8. Expand **BusinessConnect > System Settings** in the left panel.
9. Click **General** in the right panel.
10. Confirm that SOAPClient is selected in the **Default Host** list.
11. Click **Save**.

Setting Up the SOAP Protocol for the Initiator Host

To set up the SOAP protocol, perform the following steps:

1. Expand **BusinessConnect > Participants** in the left panel.
2. Click the **SOAPClient** link in the right panel.
3. Click the **Protocols** tab.
4. Verify that **SOAP** is listed in the **Protocol Name** list.

If **SOAP** is not displayed in the list of protocols:

- a. Click **Enable**.
- b. Select the **SOAP** check box.
- c. Click **OK**.
- d. Click **Save**.

Setting Up the Responder as a Trading Partner

The responder trading partner setup consists of these steps:

1. [Setting Up the Responder Partner, page 33](#)
2. [Setting Up the SOAP Protocol for the Responder Partner, page 34](#)

Setting Up the Responder Partner

To set up the responder partner, perform the following steps:

1. Expand **BusinessConnect > Participants** in the left panel.
2. Click **New**.
3. Type SOAPServer in the **Name** field.

4. Select **Partner** from the **Type** list.
5. Click **OK**.
6. In the New Partner Participant window with SOAPServer in the **Participant Name** field, select the **Active** check box.

Setting Up the SOAP Protocol for the Responder Partner

To set up the SOAP protocol, perform the following steps:

1. In the Edit Partner Participant with SOAPServer window, click the **Protocols** tab.
2. Click **Enable**.
3. Select the **SOAP** check box.
4. Click **OK**.
5. Click the **SOAP** link.
6. Click the **Transports** tab.
7. Click **Add**.
8. In the New Transport window, type **HTTP** in the **Name** field.
9. Select **HTTP** from the **Type** list.
10. Click **OK**.
11. In the New HTTP Transport window, type information as shown in [Table 3](#).

Table 3 New HTTP Transport

Field	Description
Transport Name	Required. Type a name for this transport: HTTP .
URL	Required. URL of the company: <code>www.hostname:6700/dmz/SOAP</code> .
Use HTTP Basic Authentication	Clear the Use HTTP Basic Authentication check box.
Username	No entry required for this tutorial.
Password	No entry required for this tutorial.
Retry Count	Leave the default value to 3.
Retry Interval	Leave the default value to 60.
Socket Timeout (seconds)	Leave the default value to 300.

12. Click **Save** three times.

Configuring the Business Agreement between the Initiator and the Responder

To configure the business agreements, perform the following steps:

1. Expand **BusinessConnect > Business Agreements** in the left panel.
2. Click **New** in the right panel.
3. Click **SOAPClient** in the **Host Party** area and **SOAPServer** in the **Partner Party** area.
4. Click **OK**.
5. In the **Protocol Bindings** area, click **Add Protocol Binding**.
6. In the Select Protocol window, select the **SOAP** check box.
7. Click **OK**.
8. Click the **SOAP** link that is displayed in the **Agreement Protocol Binding** list.
9. In the **Operation Bindings** tab, verify that the **Allow All Operations** check box is selected. The host and trading partner can initiate all enabled operations.
10. Click the **Transports** tab.
11. In the **Outbound Transports for Host 'SOAPClient'** area, select **HTTP** from the **Primary Transport** list.
12. In the **Allowed Inbound Transports for Partner 'SOAPServer'** area, ensure that the **HTTP** check box is selected.
13. Click **Save** twice.

After performing these steps, the initiator configuration is completed.

Configuring the Responder

This section guides you through the activities you must perform to configure the responder trading partner:

1. [Importing Operations on the Responder](#), page 36
2. [Setting Up the Responder Server](#), page 37
3. [Setting Up the Responder as a Trading Host](#), page 38
4. [Setting Up the Initiator as a Trading Partner](#), page 39
5. [Configuring the Business Agreements between the Initiator and the Responder](#), page 40

Importing Operations on the Responder

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

1. In TIBCO Administrator, expand **BusinessConnect > Operations Editor** in the left panel.
2. Click **Import**.
3. Click **change**.
4. Click **Browse** and navigate to the *TIBCO_HOME/bc/version_number/protocols/soap/samples/server* directory and select the *operations.csx* file.



If you want to use NHIN/esMD Authorization Framework with SAML assertions, import the *caqh.csx* file in the *TIBCO_HOME/bc/version_number/protocols/soap/samples/caqh* directory. This file contains several sample operations that might help you understand how to use this feature.

5. Click **Open**.
6. Click **OK**.
7. Type a password. This is optional.
8. Click **Import**.
9. Click **Done**.

Setting Up the Responder Server

You must set up the responder server to communicate with its trading partners. To set up the responder, perform the following steps:

1. Create the deployment configuration.
See *TIBCO BusinessConnect Interior Server Administration*.
2. [Setting Up the Responder Server Transport](#)

Setting Up the Responder Server Transport

To configure the BusinessConnect server, perform the following steps:

1. In TIBCO Administrator, expand **BusinessConnect > System Settings** in the left panel.
2. Click **Inbound Public Transport Types** in the right panel.
3. Select the **HTTP, HTTPS** and **HTTPSCA** check boxes, and click **Enable**.
4. Click **Done**.
5. Expand **BusinessConnect > Gateway** in the left panel.
6. Click **Gateway Services** in the right panel.
7. Click **New**.
8. Type **HTTP** in the **Name** field, and select **HTTP** from the **Type** list.
9. Click **OK**.
10. In the **General** tab, select the **Active** check box.
11. In the **Credentials** tab, Click **New Private Key**.
12. Type **Partner Key** in the **Alias** field.
13. Click **change** link.
14. Click **Browse** and navigate to the following directory:
TIBCO_HOME/bc/version_number/samples/keys.
15. Select **bcpartner2_key.p12** and click **Open**.
16. Click **OK**.
17. Click **set** next to **Password**.
18. Type **Password1** in both the **Enter password** and **Enter password again** fields.
19. Click **OK**.
20. Click **Save**.

21. Click the **Transport** tab, select **Partner Key** from the **Private Key Credential for Secure Ports** list.
22. Click **Save**.

TIBCO BusinessConnect on the responder side is configured with HTTP and HTTPS transports.

Setting Up the Responder as a Trading Host

On the responder machine, set up the responder as a trading host.

The trading host setup for the responder consists of these steps:

1. [Setting Up the Responder Host, page 38](#)
2. [Setting Up the SOAP Protocol for the Responder Host, page 38](#)

See [Setting Up the Initiator as a Trading Partner on page 39](#).

Setting Up the Responder Host

The trading host name property is typically defined when setting up BusinessConnect.

If the property is set, go to [Setting Up the SOAP Protocol for the Responder Host on page 38](#). If the property is not set, perform the following steps.

1. In TIBCO Administrator, expand **BusinessConnect** > **Participants** in the left panel.
2. Click **New**.
3. Type **SOAPServer** in the **Name** field.
4. Select **Host** from the **Type** list.
5. Click **OK**.
6. In the New Host Participant window with the participant name **SOAPServer**, select the **Active** check box.
7. Click **Save**.
8. Expand **BusinessConnect** > **System Settings** in the left panel.
9. Click **General** in the right panel.
10. Confirm that **SOAPServer** is selected in the **Default Host** list.
11. Click **Save**.

Setting Up the SOAP Protocol for the Responder Host

To set up the SOAP protocol, perform the following steps:

1. Expand **BusinessConnect > Participants** in the left panel.
2. Click the **SOAPServer** link in the right panel.
3. Click the **Protocols** tab.
4. Verify that **SOAP** is listed in the **Protocol Name** list.
If **SOAP** is not displayed in the list of protocols:
 - a. Click **Enable**.
 - b. Select the **SOAP** check box.
 - c. Click **OK**.
 - d. Click **Save**.

Setting Up the Initiator as a Trading Partner

The initiator trading partner setup consists of these steps:

1. [Setting Up the Initiator Partner, page 39](#)
2. [Setting Up the SOAP Protocol for the Initiator Partner, page 39](#)

Setting Up the Initiator Partner

To set up the initiator partner, perform the following steps:

1. Expand **BusinessConnect > Participants** in the left panel.
2. Click **New**.
3. Type **SOAPClient** in the **Name** field.
4. Select **Partner** from the **Type** list.
5. Click **OK**.
6. In the New Partner Participant window with **SOAPClient** in the **Participant Name** field, select the **Active** check box.

Setting Up the SOAP Protocol for the Initiator Partner

To set up the SOAP protocol, perform the following steps:

1. In the Edit Partner Participant with **SOAPClient** window, click the **Protocols** tab.
2. Click **Enable**.
3. Select the **SOAP** check box.
4. Click **OK**.

5. Click the **SOAP** link.
6. Click the **Transports** tab.
7. Click **Add**.
8. In the New Transport window, type **HTTP** in the **Name** field.
9. Select **HTTP** from the **Type** list.
10. Click **OK**.
11. In the New HTTP Transport window, type information as in [Table 3](#).
12. Click **Save** three times.

Configuring the Business Agreements between the Initiator and the Responder

To configure the business agreements, perform the following steps:

1. Expand **BusinessConnect > Business Agreements** in the left panel.
2. Click **New** in the right panel.
3. Click **SOAPServer** in the **Host Party** area and **SOAPClient** in the **Partner Party** area.
4. Click **OK**.
5. In the **Protocol Bindings** area, click **Add Protocol Binding**.
6. In the Select Protocol window, select the **SOAP** check box.
7. Click **OK**.
8. Click the **SOAP** link that is displayed in the **Agreement Protocol Binding** list.
9. In the **Operation Bindings** tab, verify that the **Allow All Operations** check box is selected. The host and trading partner can initiate all enabled operations.
10. Click the **Transports** tab.
11. In the **Outbound Transports for Host 'SOAPServer'** area, select **HTTP** from the **Primary Transport** list.
12. In the **Allowed Inbound Transports for Partner 'SOAPClient'** area, ensure that the **HTTP** check box is selected.
13. Click **Save** twice.

After performing these steps, the responder configuration is completed.

Running the Tutorial

The following parts demonstrate how to run the standalone private processes on the initiator and responder machines, and how to view the audit logs.

Running the Standalone Private Processes

To run the tutorial, perform the following steps:

1. Start the initiator and responder runtime servers.
2. Open the `runSOAPServer.bat` or `runsoapsrvr` file in the `TIBCO_HOME/bc/version_number/protocols/soap/samples/server` directory on the responder machine with a text editor and make the following changes:
 - In the line `SET BC_INSTANCE=%%BC_INSTANCE%%`, replace `%%BC_INSTANCE%%` with the name of your installation, such as `BC-Responder`.
 - In the line `SET JDK_DIR=%%JDK_DIR%%`, replace `%%JDK_DIR%%` with `JAVA_directory`, such as `C:\tibco\tibcojre\1.7.0`.
 - In the line `SET RV_DIR=%%RV_DIR%%`, replace the value `%%RV_DIR%%` with the `RV` library location, such as `C:\tibco\tibrv\8.4`.

A sample edited `runSOAPServer.bat` file looks as follows:

```
@ECHO OFF

SET BC_INSTANCE=BC-Responder
SET JDK_DIR=C:\tibco\tibcojre\1.7.0
SET RV_DIR=C:\tibco\tibrv\8.4

SET CLASSPATH=
.;\..\..\..\lib\common\configstore-core.jar;%RV_DIR%\lib\tibrvj.jar;%CLASSPATH%
%JDK_DIR%\bin\java SOAPServer SOAPServer.properties %BC_INSTANCE% body.xml header.xml
```

3. Start the responder private process on the responder machine, navigate to the `TIBCO_HOME/bc/version_number/protocols/soap/samples/server` directory and run the following command on a command line:

- **Windows** `runSOAPServer.bat`
- **UNIX** `./runsoapsrvr`

The sample example is performed on a Microsoft Windows machine.

Example: runSOAPServer.bat

```
C:\tibco\bc\6.2\protocols\soap\samples\server>runSOAPServer.bat
**** SOAP Server Private Process ****
listening on: AX.BC.BC-Responder.SOAP.RESPONDER.REQUEST
```

4. Open the runSOAPClient.bat or runsoapclient file in the *TIBCO_HOME/bc/version_number/protocols/soap/samples/client* directory on the initiator machine with a text editor and make the following changes:
 - In the line `SET BC_INSTANCE=%BC_INSTANCE%`, replace `%BC_INSTANCE%` with the name of your installation, such as `BC-Initiator`.
 - In the line `SET JDK_DIR=%JDK_DIR%`, replace `%JDK_DIR%` with `JAVA_directory`, such as `C:\tibco\tibcojre\1.7.0`.
 - In the line `SET RV_DIR=%RV_DIR%`, replace the value `%RV_DIR%` with the `RV` library location, such as `C:\tibco\tibrv\8.4`.

A sample edited runSOAPClient.bat file looks as follows:

```
@ECHO OFF

SET BC_INSTANCE=BC-Initiator
SET JDK_DIR=C:\tibco\tibcojre\1.7.0
SET RV_DIR=C:\tibco\tibrv\8.4

TITLE SOAP Client private process

SET CLASSPATH=
.;\..\..\..\lib\common\configstore-core.jar;%RV_DIR%\lib\tibrvlib.jar;%CLASSPATH%
;%JDK_DIR%\bin\java SOAPClient SOAPClient.properties %BC_INSTANCE% body.xml header.xml
```

5. On the initiator machine, start the initiator private process, navigate to the *TIBCO_HOME/bc/version_number/protocols/soap/samples/client* directory and run the following command on a command line:

- **Windows** runSOAPClient.bat
- **UNIX** ./runsoapclient

In this example, the following operations are processed:

- The operation `POSync` containing a purchase order for five copies of a word processing application is sent from the initiator to the responder.
- The responder confirms the receipt of that order.
- The responder sends an invoice to the initiator.

The following text is displayed when SOAP Protocol 1.1 is selected:

Example: runSOAPClient.bat

```

**** SOAPClient Private Process ****
listening on: AX.BC.BC-Initiator.SOAP.INITIATOR.RESPONSE

Hit [Enter] to send request:
Sending request...
publishing on subject: AX.BC.BC-Initiator.SOAP.INITIATOR.REQUEST
Sending request:
Trading partner: SOAPServer
Operation ID: POInterface/1.0/POSync
Header: <ep:endpoints xmlns:ep="http://user.org/header" >
  <ep:to>
    <ep:address>
      <name>Book Orders</name>
      <street>1st Street</street>
      <city>New York</city>
      <zip>1111</zip>
    </ep:address>
  </ep:to>
  <ep:from>
    <ep:address>
      <name>Book Lovers</name>
      <street>1st Street</street>
      <city>Los Angeles</city>
      <zip>90210</zip>
    </ep:address>
  </ep:from>
</ep:endpoints>
<prop:properties xmlns:prop = "http://user.org/header" >
  <identity>uuid:74b9f5d0-33fb-4a81-b02b-5b760641c1d6</identity>
  <sentAt>2000-05-14T03:00:00+08:00</sentAt>
  <expiresAt>2000-05-15T04:00:00+08:00</expiresAt>
  <topic>http://electrocommerce.org/purchase_order</topic>
</prop:properties>

Body: <cpo:CommonPO xmlns:cpo="http://po.org/body" >
  <cpo:POHeader CreationDate="2000-06-23" Number="12345" Purpose="PO" Type="EZ" />
  <cpo:BillTo/>
  <cpo:ShipTo ContactName="BonifazLuis" ContactNumber="12345" ContactType ="CT"/>
  <cpo:Item>
    <cpo:ItemHeader ExtendedPrice="499.75" Price="99.95" Quantity="5" UnitOfMeasure="EA"/>
    <cpo:ItemDescription Description="Word Processing Application" Type="F"/>
  </cpo:Item>
  <cpo:Total LineItemTotal="87" POTotal="544.15" QuantityTotal="34">12.34</cpo:Total>
  <cpo:Description>VGhpcyBpcyBhIHBlcmNoYXNlIG9yZGVyIGRlc2NyaXB0aW9uLg==</cpo:Description>
</cpo:CommonPO>

```

Attachment : po.txt

Attachment : image.gif

Hit [Enter] to send request:

Received message from BusinessConnect:

Status code: 200

Status message: OK

operation id: POInterface/1.0/POSync

transaction id: bsoap:DA75A27A-5BCA-401F-930E-1B7BF0EEB59D

response header: <ep:endpoints xmlns:ep="http://user.org/header">

```
<ep:to>
  <ep:address>
    <name>Book Orders</name>
    <street>1st Street</street>
    <city>New York</city>
    <zip>1111</zip>
  </ep:address>
</ep:to>
<ep:from>
  <ep:address>
    <name>Book Lovers</name>
    <street>1st Street</street>
    <city>Los Angeles</city>
    <zip>90210</zip>
  </ep:address>
</ep:from>
</ep:endpoints>
```

response: <ci:CommonInvoice xmlns:ci="http://ci.org/body">

```
<ci:InvoiceHeader Date="2000-05-16" Number="12345" PODate="2000-05-16" PONumber="12345" ShipDate="2000-05-16" Type=
EZ" />
  <ci:Seller ContactName="BonifazLuis" ContactNumber="12345" ContactType="CT" />
  <Buyer />
  <ci:Item>
    <ci:ItemHeader LineNumber="1" Price="99.95" Quantity="5" QuantityDiff="10" UnitOfMeasure="EA"/>
    <ci:ItemTaxReference Amount="3.75" Description="VAT" />
    <ci:ItemDescription Description="Word Processing Application" Type="F" />
  </ci:Item>
  <ci:InvoiceSummary Amount="544.15" />
  <ci:Description>VGhpcyBpcyBhIGludm9pY2UgZGVzY3JpcHRpb24u</ci:Description>
</ci:CommonInvoice>
```

On the Responder machine, the complete transaction is displayed as follows:

**** SOAP Server Private Process ****

listening on: AX.BC.BC-Responder.SOAP.RESPONDER.REQUEST

Request header:

```
<ep:endpoints xmlns:ep="http://user.org/header">
  <ep:to>
    <ep:address>
      <name>Book Orders</name>
      <street>1st Street</street>
      <city>New York</city>
```

```

    <zip>1111</zip>
  </ep:address>
</ep:to>
<ep:from>
  <ep:address>
    <name>Book Lovers</name>
    <street>1st Street</street>
    <city>Los Angeles</city>
    <zip>90210</zip>
  </ep:address>
</ep:from>
</ep:endpoints>
<prop:properties xmlns:prop="http://user.org/header">
  <identity>uuid:74b9f5d0-33fb-4a81-b02b-5b760641c1d6</identity>
  <sentAt>2000-05-14T03:00:00+08:00</sentAt>
  <expiresAt>2000-05-15T04:00:00+08:00</expiresAt>
  <topic>http://electrocommerce.org/purchase_order</topic>
</prop:properties>

```

request body:

```

<cpo:CommonPO xmlns:cpo="http://po.org/body">
  <cpo:POHeader CreationDate="2000-06-23" Number="12345" Purpose="PO" Type="EZ"/>
  <cpo:BillTo/>
  <cpo:ShipTo ContactName="BonifazLuis" ContactNumber="12345" ContactType="CT"/>
  <cpo:Item>
    <cpo:ItemHeader ExtendedPrice="499.75" Price="99.95" Quantity="5" UnitOfMeasure="EA"/>
    <cpo:ItemDescription Description="Word Processing Application" Type="F"/>
  </cpo:Item>
  <cpo:Total LineItemTotal="87" POTotal="544.15" QuantityTotal="34">12.34</cpo:Total>
  <cpo:Description>VGhpcyBpcyBhIHBB1cmNoYXNlIG9yZGVyIGRlc2NyaXB0aW9uLg==</cpo:Description>
</cpo:CommonPO>

```

operation ID: POInterface/1.0/POSync

closure:9F3E47B8-54CE-42D7-B3F0-E85CF7B1C363

Attachment saved to file po.txt

Attachment saved to file image.gif

Hit [Enter] to send response:

Sending response...

publishing on subject: AX.BC.BC-Responder.SOAP.RESPONDER.RESPONSE

response header: <ep:endpoints xmlns:ep="http://user.org/header">

```

  <ep:to>
    <ep:address>
      <name>Book Orders</name>
      <street>1st Street</street>
      <city>New York</city>
      <zip>1111</zip>
    </ep:address>
  </ep:to>
  <ep:from>
    <ep:address>
      <name>Book Lovers</name>
      <street>1st Street</street>
      <city>Los Angeles</city>
      <zip>90210</zip>
    </ep:address>
  </ep:from>
</ep:endpoints>

```

```

response:      <ci:CommonInvoice xmlns:ci="http://ci.org/body">
                <ci:InvoiceHeader Date="2000-05-16" Number="12345" PODate="2000-05-16" PONumber="12345" Type="EZ"
ShipDate="2000-05-16"/>
                <ci:Seller ContactType="CT" ContactName="BonifazLuis" ContactNumber="12345"/>
                <Buyer/>
                <ci:Item>
                    <ci:ItemHeader LineNumber="1" Quantity="5" Price="99.95" UnitOfMeasure="EA" QuantityDiff="10"/>
                    <ci:ItemTaxReference Description="VAT" Amount="3.75"/>
                    <ci:ItemDescription Type="F" Description="Word Processing Application"/>
                </ci:Item>
                <ci:InvoiceSummary Amount="544.15"/>
                <ci:Description>VGhpcyBpcyBhlGludm9pY2UgZGVzY3JpcHRpb24u</ci:Description>
            </ci:CommonInvoice>

```

Changing the Message Type

To change the message type that is sent from the initiator to the responder by using the command-line interface, perform the following steps:

1. Using a text editor, open the `SOAPClient.properties` file in the `TIBCO_HOME/bc/version_number/protocols/soap/samples/client` directory.

The content of the `SOAPClient.properties` file looks as follows:

```

#
# SOAP Client Private process property file.
#

#client.operationID:      POInterface/1.0/PONotify
client.operationID:      POInterface/1.0/POSync
client.tradingPartnerID:SOAPServer
client.encoding:ISO8859_1
#client.transactionID:uuid:1234567917

attachment1.name:po.txt
attachment1.contentID:<process-1@xyz.com>
attachment1.contentType:text/plain

attachment2.name:image.gif
attachment2.contentID:<process-2@xyz.com>
attachment2.contentType:binary

```

2. Comment out one of the two following lines:

```
#client.operationID: POInterface/1.0/PONotify
```

or

```
#client.operationID: POInterface/1.0/POSync
```

3. The operation that is not commented out is executed: `PONotify` or `POSync`.

Viewing the Audit Logs

To view the audit logs on the initiator or responder machines, perform the following steps:

1. Log in to TIBCO Administrator, expand **Business Connect > Log Viewer** in the left panel.
2. Click the **SOAP** link in the right panel.
3. Select items from the following lists: **Status**, **Connection**, and **Date Range**.
4. Click **Search**.

See *TIBCO BusinessConnect Trading Partner Administration* for information on searching the logs.

Chapter 4

TIBCO ActiveMatrix BusinessWorks Private Processes Tutorial

This chapter gives an overview of how to use TIBCO ActiveMatrix BusinessWorks with TIBCO BusinessConnect SOAP Protocol.

Topics

- [Overview, page 50](#)
- [Setting Up the Tutorial, page 53](#)
- [Configuring TIBCO ActiveMatrix BusinessWorks Private Processes, page 54](#)
- [Introduction of Process Definitions, page 58](#)
- [Running the Tutorial, page 65](#)

Overview

You can use TIBCO BusinessConnect SOAP Protocol to exchange data with a TIBCO ActiveMatrix BusinessWorks installation that is acting as a private process and the interface to enterprise back office systems.

Two operations are defined in the sample interface file:

- `PONotify` This operation is used only when the initiator requires an acknowledgment from the responder to verify business document receipt.
- `POSync` This is a synchronous request-response operation that is used when the initiator and the responder do a more complex business exchange of business documents.

Only the `PONotify` operation is used in this tutorial.

You must have the TIBCO BusinessConnect palette working as a plug-in of TIBCO ActiveMatrix BusinessWorks to communicate with TIBCO BusinessConnect.

You must install TIBCO BusinessConnect Palette or TIBCO ActiveMatrix BusinessWorks Plug-in for BusinessConnect after installing TIBCO BusinessConnect, so the BusinessConnect palette is installed with TIBCO ActiveMatrix BusinessWorks.

TIBCO BusinessConnect palette consists of several activities, which work in TIBCO ActiveMatrix BusinessWorks processes for sending and receiving messages from TIBCO BusinessConnect. By using TIBCO BusinessConnect palette, you can also connect to TIBCO BusinessConnect configuration store to import the configurations of SOAP operations, which is useful in the interactions with TIBCO BusinessConnect SOAP Protocol.

TIBCO Designer and TIBCO Business Studio are the design time tools of TIBCO ActiveMatrix BusinessWorks processes. You can use TIBCO Designer or TIBCO Business Studio to design TIBCO ActiveMatrix BusinessWorks processes and interact with TIBCO BusinessConnect SOAP Protocol.

See *TIBCO BusinessConnect Palette User's Guide* or *TIBCO ActiveMatrix BusinessWorks Plug-in for BusinessConnect User's Guide*.

The TIBCO ActiveMatrix BusinessWorks Project Tutorial

The TIBCO ActiveMatrix BusinessWorks project tutorial is included in the TIBCO BusinessConnect SOAP Protocol installation in the `TIBCO_HOME/bc/version_number/protocols/soap/samples/bcpalette` directory.

Two files are in this directory:

- **bcpalette.zip** This file is used with TIBCO ActiveMatrix BusinessWorks 5.

- **bcpalette_for_bw6.zip** This file is used with TIBCO ActiveMatrix BusinessWorks 6.

The tutorial demonstrates the use of TIBCO ActiveMatrix BusinessWorks to interact with TIBCO BusinessConnect SOAP Protocol. Knowledge of TIBCO ActiveMatrix BusinessWorks, TIBCO BusinessConnect SOAP Protocol, and business-to-business is a prerequisite to run this tutorial.

These simple samples have been designed to illustrate the TIBCO BusinessConnect activities available in TIBCO ActiveMatrix BusinessWorks. In a real-life scenario, the requesting TIBCO ActiveMatrix BusinessWorks engine would perform a lot of processing to generate a request. Likewise, the receiving TIBCO ActiveMatrix BusinessWorks engine would also perform a lot of processing and transformation from the request received from TIBCO BusinessConnect server to integrate with the backend systems.

To use TIBCO ActiveMatrix BusinessWorks as a private process, it must connect to TIBCO BusinessConnect configuration store to retrieve SOAP operation configurations.

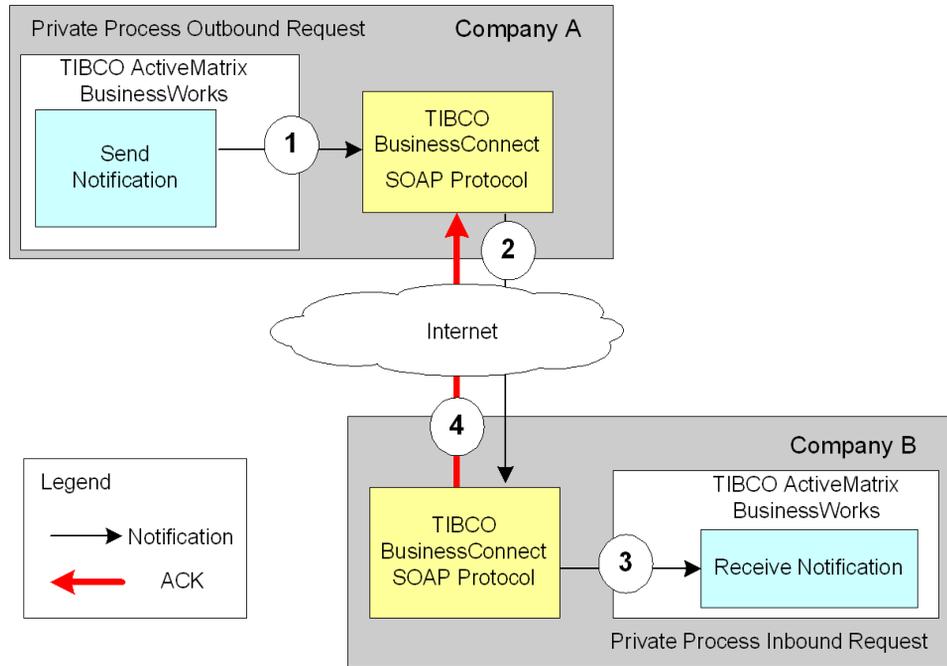
Transactions

The PONotify operation consists of the following message flows:

1. The initiator private process sends a private process message to the initiator BusinessConnect server.
2. The initiator BusinessConnect server sends a request SOAP message to the responder BusinessConnect server.
3. The responder BusinessConnect server sends the request private process message to the responder private process.
4. After the message is received, the responder BusinessConnect server sends an acknowledgement (ack) back to the initiator BusinessConnect server.

A detailed diagram of the PONotify operation is displayed in [Figure 5](#).

Figure 5 Notify Message



Setting Up the Tutorial



Before you proceed, ensure that you have completed the tutorial in [Chapter 3](#).

Prerequisites

To run this tutorial you must perform the following prerequisites:

- Make sure that you have satisfied all prerequisites listed in [Prerequisites on page 29](#).
- Install TIBCO ActiveMatrix BusinessWorks.
- Install TIBCO BusinessConnect Palette or TIBCO ActiveMatrix BusinessWorks Plug-in for BusinessConnect, so the BusinessConnect palette is installed with TIBCO ActiveMatrix BusinessWorks.

Configuring the Machines

The configuration of the machines is the same as described in the standalone private processes tutorial. If you have not configured the initiator and responder machines yet, see the following sections:

- [Configuring the Initiator, page 30](#)
- [Configuring the Responder, page 36](#)

Once both machines are configured, proceed with [Configuring TIBCO ActiveMatrix BusinessWorks Private Processes on page 54](#).

Configuring TIBCO ActiveMatrix BusinessWorks Private Processes

TIBCO BusinessConnect SOAP Protocol contains an example TIBCO ActiveMatrix BusinessWorks project, which sends a transaction to BusinessConnect.

This section describes how to configure private processes in the following ways:

- [Configuring Private Processes in TIBCO Designer, page 54](#)
- [Configuring Private Processes in TIBCO Business Studio, page 56](#)

Configuring Private Processes in TIBCO Designer

To configure the tutorial in TIBCO Designer, perform the following steps:

Task A Opening the TIBCO ActiveMatrix BusinessWorks Project

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

1. Start TIBCO Designer.
2. Click **New empty project**.
3. In the Save Project window, click **Cancel**.
4. Click **Project > Import Full Project**.
5. Click the **ZIP Archive** tab in the Import Project dialog.
6. Click  and navigate to the `TIBCO_HOME/bc/version_number/protocols/soap/samples/bcpalette` directory.
7. Select the `bcpalette.zip` file and click **Open**. Click **OK**.
8. In the **Options** tab, select **Try rename in case of name conflict**. Click **Apply**.
9. Click **Project > Save As**.
10. In the **Project Directory** field, navigate to a folder where you want to save the sample project.
11. Select **UTF-8** from the **TIBCO Messaging Encoding** list.
12. Select **None** from the **Multi-User System** list. Click **OK**.

Task B Configuring Connections to TIBCO BusinessConnect

To configure connections to TIBCO BusinessConnect, perform the following steps:

1. Click the **Global Variables** tab.

2. Verify that the `BC_HOME` variable is set to `TIBCO_HOME\bc\version_number`.
3. Click the **Project** tab.
4. Expand the **Connections** folder and click **BCServer Connection**.
5. Click the **BusinessConnect Server Access** tab in the right panel.
 - a. Select the JDBC driver you use to communicate with the BusinessConnect configuration store from the **JDBC Driver Type** list.
 - b. Type the *URL* for the configuration store in the **JDBC URL** field.
 - c. Type the configuration store values in the **DB User** and **DB Password** fields.
 - d. Click **Apply**.
6. Click the **Configuration** tab.
7. If the **Update Transport Settings** check box is not selected, select it.
8. Click **Update from Configuration Store**.
9. Select **SOAP** from the **Protocol Name** list.
10. Click **Import Selected Business Protocol**.

The operations previously imported to configuration store are displayed, as shown in [Figure 6](#).



This connection setup must be done on the initiator and the responder sides separately. The JDBC connection parameters for the TIBCO BusinessConnect configuration store on the initiator and the responder sides are different.

Figure 6 Imported Operations for SOAP

The screenshot shows a configuration window titled "BCServer Connection (BusinessConnect Connection)". It has three tabs: "Configuration", "BusinessConnect Server Access", and "Rendezvous". The "Configuration" tab is selected. The fields are as follows:

- Name: BCServer Connection
- Description: (empty)
- Server: BC-Initiator
- PP Transport: Tibrv
- Update Transport Settings:
- Protocol Name: SOAP (dropdown menu)
- Imported Operations: POInterface/1.0/PONotify, POInterface/1.0/POSync
- Use Smart ID:

At the bottom, there are three buttons: "Update from Configuration Store", "Import Selected Business Protocol", and "Apply".

When you import TIBCO BusinessConnect SOAP Protocol, TIBCO Designer retrieves the schema information for the business documents associated with the SOAP operations. The schema information has been imported to TIBCO BusinessConnect configuration store previously. By retrieving the schemas of the business documents to TIBCO Designer, you can use the TIBCO ActiveMatrix BusinessWorks processes to construct the business document to be sent as a request, or validate and parse the received business document.

The retrieved schemas are in the **BCSchemas** project folder. If you expand this folder, you can see the `CommonPO` and `CommonInvoice` schemas in **BCSchemas > SOAP > POInterface > 1.0 > PONotify** and **POSync**.

11. Click **Apply**.
12. Click the **Save**  icon to save the project.

Configuring Private Processes in TIBCO Business Studio

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

Task A Opening the TIBCO ActiveMatrix BusinessWorks Project

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

1. Start TIBCO Business Studio.
2. Click **File > Import**.
3. On the Import page, expand the **General** folder and select **Existing Studio Projects into Workspace**. Click **Next**.
4. Click **Browse** next to the **Select archive file** field to navigate to the `TIBCO_HOME/bc/version_number/protocols/soap/samples/bcpalette` directory, and select the `bcpalette_for_bw6.zip` file. Click **Open**.
5. Click **Finish**.

After importing the sample, perform the following steps:

1. Expand **bcpalette > Module Descriptors** in the Project Explorer view.
2. Double-click **Module Properties**.
3. Change the default value of the **BC_HOME** property according to your environment.

Task B Configuring Connections to TIBCO BusinessConnect

To configure connections to TIBCO BusinessConnect, perform the following steps:

1. In the Project Explorer view, expand **Resources > bcpalette**, and double-click **BCConnection.bcResource**.
2. Click the **Server Access** tab.
3. Type information as explained in [step 5](#).
4. Click the **Configuration** tab, and click **Update from Configuration Store**.
5. Select **SOAP** from the **Protocol Name** list.

If you select the **Select Operations** check box, 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 TIBCO BusinessConnect configuration store and puts them in the project folder.

7. Click **Save**.



This connection setup must be done on the initiator and the responder sides separately. The JDBC connection parameters for the TIBCO BusinessConnect configuration store on the initiator and the responder sides are different.

Introduction of Process Definitions

After you have opened the TIBCO ActiveMatrix BusinessWorks project, and configured connections to TIBCO BusinessConnect, you can access the initiator and responder process definitions:

- [Introducing Initiator Process Definitions, page 58](#)
- [Introducing Responder Process Definitions, page 60](#)

You can also access the NHIN/esMD process definitions: [Introducing NHIN/esMD Process Definitions, page 62](#).



Process names are different in TIBCO Designer and TIBCO Business Studio. For example, the **Send Notify** process in TIBCO Designer equals the **SendNotify** process in TIBCO Business Studio. The following description takes processes in TIBCO Designer as an example.

Introducing Initiator Process Definitions

In TIBCO Designer, you can access the initiator process definitions by clicking the **Project** tab, and then expanding the **Initiator** folder. The following processes are available:

- [Send Notify](#)
- [Send Request and Receive Response](#)

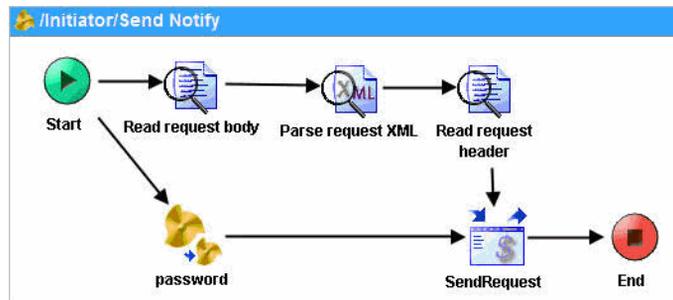
Only the Send Notify operation is used in this tutorial.

Send Notify

This process uses the TIBCO ActiveMatrix BusinessWorks Read File activity to read the notify document stored in the samples directory. This process uses the Parse XML activity to parse the content into an XML tree and pass the tree to the Send Request/Notification activity.

This process includes a subprocess, the **password** process. This process converts the clear text password to base64 encoded data. This is used for sending the user name and password information to TIBCO BusinessConnect, so you can send a SOAP message with WSS UsernameToken.

The Send Request/Notification activity creates a notify message and sends it to the initiator BusinessConnect server, as shown in [Figure 7](#).

Figure 7 *Send Notify*

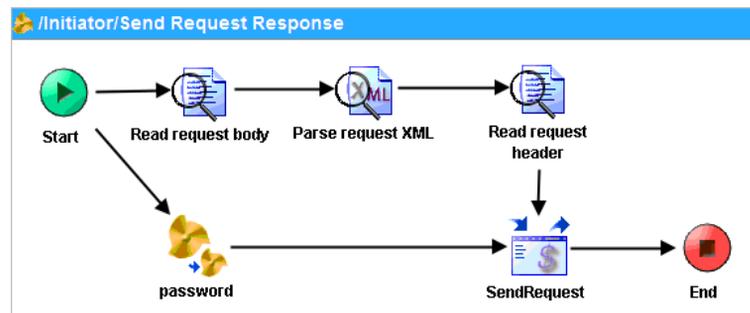
The initiator BusinessConnect server validates the notification message, and converts it into a well-formed SOAP document. The initiator BusinessConnect server transmits the SOAP document to the responder BusinessConnect server, which sends the message to the responder private process.

Send Request and Receive Response

This process uses the Read File activity to read the request document stored in the samples directory. The process uses the Parse XML activity to parse the content into an XML tree and pass the tree to the Send Request/Notification activity.

This process includes a subprocess, the `password` process. This process converts the clear text password to base64 encoded data.

The Send Request/Notification activity creates a request message and sends it synchronously to the initiator BusinessConnect server. The activity waits for the response from the trading partner, as shown in Figure 8.

Figure 8 *Send Request and Receive Response*

When the response document arrives, the initiator BusinessConnect server validates the response document for syntactical correctness and sends it to the TIBCO ActiveMatrix BusinessWorks private process.

The document is received by the Send Request/Notification activity, which is waiting for the response.

Receive Response

The response from the trading partner could be a SOAP message with a business document or a SOAP fault. If the trading partner response is a SOAP fault, BusinessConnect server sends this fault message to the TIBCO ActiveMatrix BusinessWorks private process. The Send Request/Notification activity parses the fault message. The fault message fields can be retrieved from this activity.

Introducing Responder Process Definitions

In TIBCO Designer, you can access the responder process definitions by clicking the **Project** tab, and then expanding the **Responder** folder. The following processes are available:

- [Receive Notification Message](#)
- [Receive Request and Send Fault](#)
- [Receive Request and Send Response](#)

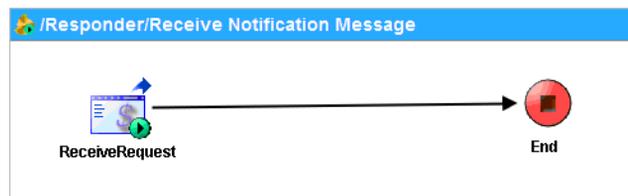
Only the Receive Notification Message operation is used in this tutorial.

Receive Notification Message

This process shows how a TIBCO ActiveMatrix BusinessWorks private process receives a SOAP notify message initiated by a trading partner. The process uses the Receive Request/Notification activity.

The Receive Request/Notification activity is configured to wait for a request from BusinessConnect server, as shown in [Figure 9](#).

Figure 9 Receive Notification Message



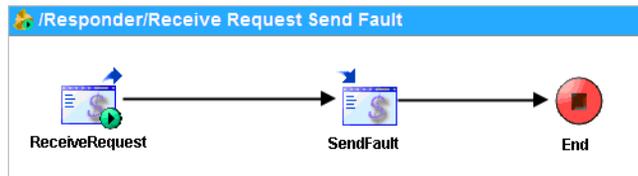
When the responder BusinessConnect server receives a SOAP document from a trading partner, it validates the SOAP message. After that, the message is sent to a TIBCO ActiveMatrix BusinessWorks private process for processing.

Receive Request and Send Fault

This process shows how TIBCO ActiveMatrix BusinessWorks can act as a responder by receiving a SOAP message initiated by a trading partner and sending back a response. If the received SOAP message is rejected by default, and a SOAP fault is generated and sent back to the trading partner.

The process uses the Receive Request/Notification and Send Response activities to process, as shown in [Figure 10](#).

Figure 10 Receive Request and Send Fault



When the responder BusinessConnect server receives a SOAP document from a trading partner, it validates the SOAP message. After that, the message is sent to a TIBCO ActiveMatrix BusinessWorks private process for processing. In this process, the Send Response activity sends a fault message to the trading partner.

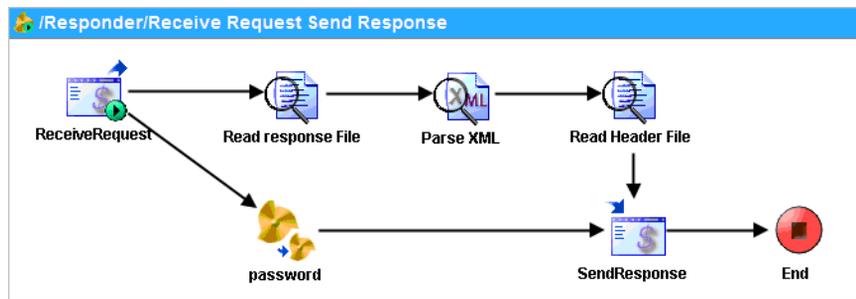
Receive Request and Send Response

This process shows how TIBCO ActiveMatrix BusinessWorks can act as a responder by receiving a SOAP message initiated by a trading partner and sending a response to the trading partner.

The process uses the Receive Request/Notification, Read File, and Send Response activities to process, read, and respond to the request message. This process includes a subprocess, the `password` process. This process converts the clear text password to base64 encoded data. This is used for sending the user name and password information to TIBCO BusinessConnect, so you can send a SOAP message with WSS UsernameToken.

See [Figure 11](#).

Figure 11 Receive Request and Send Response



When the responder BusinessConnect server receives a SOAP document from a trading partner, it validates the SOAP message. After that, the message is sent to a TIBCO ActiveMatrix BusinessWorks private process for processing. In this process, the Send Response activity sends a SOAP message back to the trading partner.

Introducing NHIN/esMD Process Definitions

In TIBCO Designer, you can access the NHIN/esMD process definitions by clicking the **Project** tab, and then expand the **NHIN/esMD** folder. This folder contains two subfolders: the **Initiator** and **Responder** folders.

- The following processes are available in the **Initiator** folder:
 - CAQH Send Batch Request
 - CAQH Send RealTime Request
- The following processes are available in the **Responder** folder:
 - CAQH Receive Batch Request Send Response
 - CAQH Receive RealTime Request Send Response

These processes are samples of how to use the NHIN/esMD SAML feature for initiators and responders. You do not have to implement your private process exactly the same way as these samples do, but you might learn detailed information, such as the `AttributeStatement` element, the `AuthzDecisionStatement` element, and so on. You can also learn how to handle these elements on the responder side, and how to prepare these elements on the initiator side.

In TIBCO ActiveMatrix BusinessWorks private process, the **CAQH Send RealTime Request** process and the **CAQH Receive RealTime Request Send Response** processes constitute a synchronous request-response transaction for RealTime scenario. The **CAQH Send Batch Request** process and the **CAQH Receive Batch Request Send Response** constitute a synchronous request-response transaction for Batch Submission scenario. These transactions are implemented with WS-Addressing to convey routing information..



The NHIN/esMD process definitions are not used in this tutorial.

Before you run this NHIN/esMD Process sample, you must enable **Require Digital Signature** and **NHIN/esMD Authorization Framework** in the Request Action tab and you must enable **NHIN/esMD Authorization Framework** in the Response Action tab for SOAP sample operation **CAQH/2.2.0/RealTimeTransaction** and **CAQH/2.2.0/BatchSubmitTransaction**.

CAQH Batch/Real time Transactions

Below are the CAQH operations available at CORERule2.2.0.xsd and CORERule 4.0.0.xsd:

CORE Rule 2.2.0

- Batch Results Acknowledgement Transaction (Operation name is *CAQH/2.2.0/BatchResultsAckSubmit Transaction*)
- Batch Results Retrieval Transaction (Operation name is *CAQH/2.2.0/BatchResultsRetrievalTransaction*)
- Batch Submit Acknowledgement Retrieval Transaction (Operation name is *CAQH/2.2.0/BatchSubmitAckRetrievalTransaction*)
- Batch Submit Transaction (Operation name is *CAQH/2.2.0/BatchSubmit Transaction*)
- Generic Batch Receipt Confirmation Transaction (Operation name is *CAQH/2.2.0/GenericBatchReceiptConfirmationTransaction*)
- Generic Batch Retrieval Transaction (Operation name is *CAQH/2.2.0/GenericBatchRetrievalTransaction*)
- Generic Batch Submission Acknowledgement Retrieval Transaction (Operation name is *CAQH/2.2.0/GenericBatchSubmissionAckRetrievalTransaction*)
- Generic Batch Submission Transaction (Operation name is *CAQH/2.2.0/GenericBatchSubmissionTransaction*)
- Real Time Transaction (Operation name is *CAQH/2.2.0/RealTimeTransaction*)

CORE Rule 4.0.0

- Batch Results Acknowledgement Transaction (Operation name is *CAQH/4.0.0/BatchResultsAckSubmitTransaction*)

- Batch Results Retrieval Transaction (Operation name is *CAQH/4.0.0/BatchresultsRetrievalTransaction*)
- Batch Submit Acknowledgement Retrieval Transaction (Operation name is *CAQH/4.0.0/BatchSubmitAckRetrievalTransaction*)
- Batch Submit Transaction (Operation name is *CAQH/4.0.0/BatchSubmitTransaction*)
- Generic Batch Receipt Confirmation Transaction (Operation name is *CAQH/4.0.0/GenericBatchReceiptConfirmationTransaction*)
- Generic Batch Retrieval Transaction (Operation name is *CAQH/4.0.0/GenericBatchRetrievalTransaction*)
- Generic Batch Submission Acknowledgement Retrieval Transaction (Operation name is *CAQH/4.0.0/GenericBatchSubmissionAckretrievalTransaction*)
- Generic batch Submission Transaction (Operation name is *CAQH/4.0.0/GenericBatchSubmissionTransaction*)
- Real Time Transaction (Operation name is *CAQH/4.0.0/RealTimeTransaction*)

Running the Tutorial



The steps of how to load and run private processes in TIBCO Business Studio are similar to TIBCO Designer. See TIBCO ActiveMatrix BusinessWorks Documentation for more details.

To run this tutorial, perform the following steps:

1. Set up the TIBCO BusinessConnect SOAP Protocol trading partners on the initiator and responder machines as described in [Configuring the Initiator on page 30](#) and [Configuring the Responder on page 36](#).
2. Send the transaction.
See [Running the TIBCO ActiveMatrix BusinessWorks Processes on page 65](#).
3. Check the results of sending the transaction.
See [Viewing the Audit Logs on page 67](#).

Running the TIBCO ActiveMatrix BusinessWorks Processes

Run TIBCO ActiveMatrix BusinessWorks processes on the responder and initiator machines.

Task A Running Process on the Responder Machine

To run the TIBCO ActiveMatrix BusinessWorks process, perform the following steps:

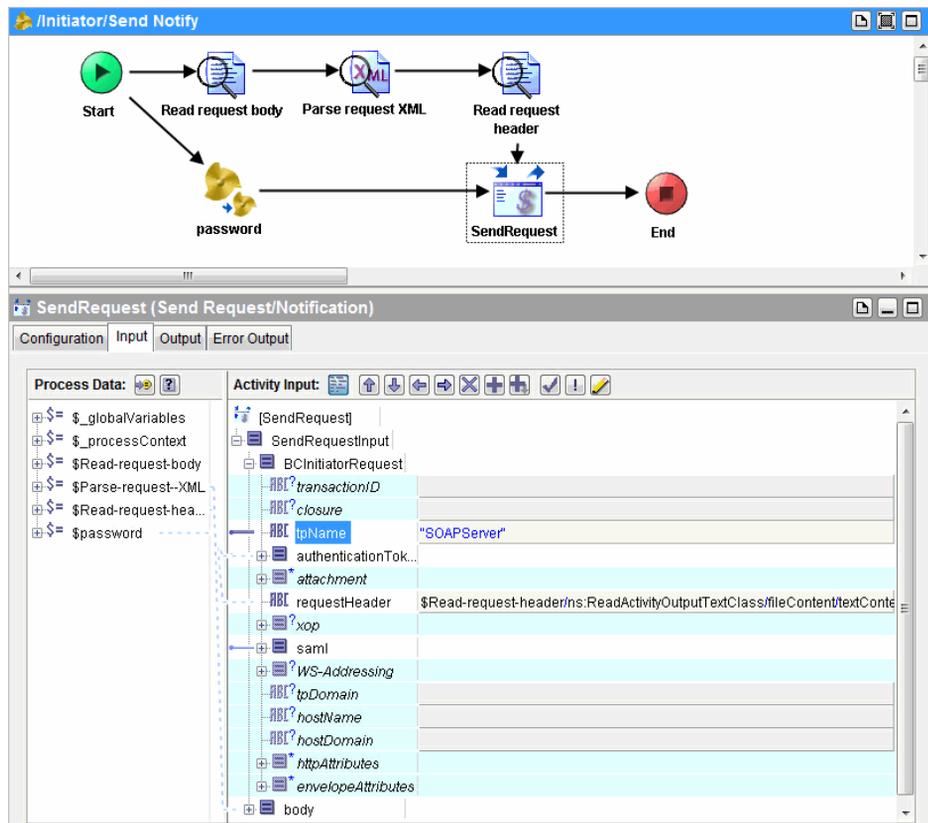
1. In the opened TIBCO Designer project, click the **Tester** tab.
2. Click .
3. Select **Responder > Receive Notification message**.
4. Click **Load Selected**.

Task B Running Process on the Initiator Machine

To run the TIBCO ActiveMatrix BusinessWorks process, perform the following steps:

1. In the opened TIBCO Designer project, expand the **Initiator** folder in the **Project** tab, and click the **Send Notify** process.
2. Click the **SendRequest** activity, and then click the **Input** tab to verify the name of the trading partner, as shown in [Figure 12](#).

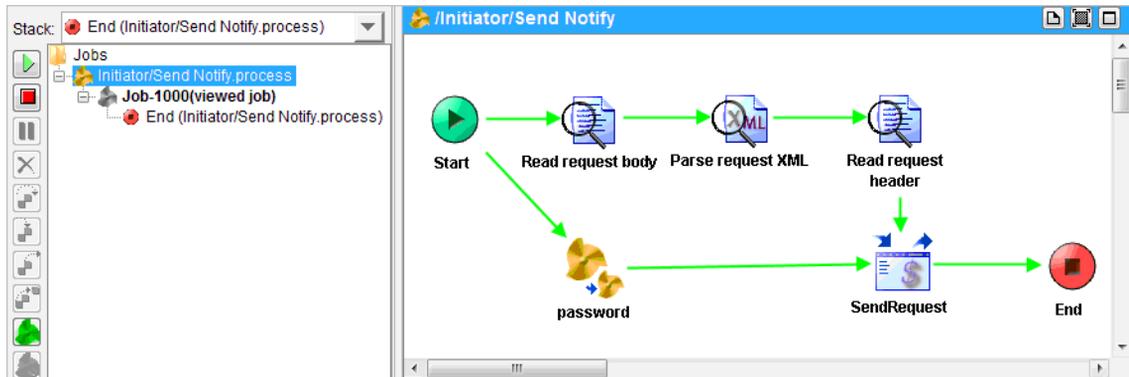
Figure 12 Verifying the Name of the Trading Partner



3. Make sure that you have the correct tpName for the SendRequest activity. Change the name if needed, and click **Apply**.
4. Click the **Tester** tab.
5. Click .
6. Select **Initiator > Send Notify**.
7. Click **Load Selected**.
8. In the **Jobs** folder, select Initiator/Send Notify.process and right-click **Create a Job**.

The operation is tested and verified, as shown in [Figure 13](#).

Figure 13 Send Notify Process



The `password` subprocess is for generating a base64 encoded password that is used for generating a `UsernameToken` element in the outbound SOAP request message. In this tutorial, this feature is not used, because the **Require UsernameToken Authentication** check box is not selected for the request action in the configuration store.

Expected Results

After a couple of seconds, if the request has been validated and processed correctly by the responder BusinessConnect server and responder private process, the line turns green on the responder machine. When processing is ended, the initiator receives an acknowledgment from the responder to verify the receipt of the SOAP message.

If something fails at the responder machine, it is likely that nothing is returned from the responder, and the Send Request activity times out.

Viewing the Audit Logs

To view the audit logs on the initiator or responder machine, perform the following steps:

1. Log in to TIBCO Administrator, expand **Business Connect > Log Viewer** in the left panel.
2. Click the **SOAP** link in the right panel.
3. Select items from the following lists: **Status**, **Connection**, and **Date Range**.
4. Click **Search**.

See *TIBCO BusinessConnect Trading Partner Administration* for information on searching the logs.

Chapter 5

Preparing Information with Your Trading Partners

This chapter gives an overview of the basic steps you and your trading partners must take to begin working together with TIBCO BusinessConnect SOAP Protocol.

Topics

- [Reaching Agreement on Shared Business Documents, page 70](#)
- [Exchanging URI Definitions, page 71](#)
- [Exchanging Identity Information, page 72](#)

Reaching Agreement on Shared Business Documents

Before trading partners can do business, they must agree on the structure and content of each XML business document that they plan to exchange over the Internet. To do this, trading partners must create XSDs that define the contents and structure of their associated XML business documents.

Trading partners can exchange an XSD file by email, the web, or any other method. This is like two people trading dictionaries before they communicate in different languages.

You can use any third-party tools to create and validate XSDs.

Exchanging URI Definitions

Partners must exchange Uniform Resource Identifiers (URIs) as part of the business agreement before they can transact electronic business. Trading partners can trade URI definitions by email, the web or any other method. The URI is part of the trading partner URL.

See [Configuring Transports for a Partner on page 103](#) for instructions on how to type the URI when setting up your trading partner.

The following is a list of the available transport protocols and their URI formats. If you want to use TIBCO BusinessConnect SOAP Protocol as the business-to-business gateway, you must give the URIs in these formats to your trading partners, so the trading partners can connect to these URIs to exchange SOAP messages with TIBCO BusinessConnect SOAP Protocol.

- **HTTP** URI format: `http://server:port/dmz/SOAP`
- **HTTPS** URI format: `https://server:port/dmz/SOAP`

Exchanging Identity Information

Trading partners must exchange information that they can use to identify the other partner when conducting electronic business. Trading partners can exchange identity information by email, the web, or any other method.

The identity information that you must share with a trading partner includes the following:

- **Trading partner name** Trading partner name is one of the ways how trading partners identify themselves from each other. See the sections on setting up trading partners for information on how to specify your partner company name.

- [Setting Up the Responder Partner on page 33](#)

- [Setting Up the Initiator Partner on page 39](#)

- **Domain identifiers** Domain and identifiers are another way for trading partners to identify themselves from each other. This is typically when both sides of the trading partners are using TIBCO BusinessConnect SOAP Protocol for implementations.

See the sections on setting up trading partners for information on where you add the domain and identifier of your partner:

- [Adding a Domain Identity for a Partner on page 102](#)

- **Public certificates** Trading partners must exchange their public certificates for encryption and authentication. Your trading partner public certificate is used to encrypt outgoing messages and authenticate incoming messages signed by your trading partner.

See *TIBCO BusinessConnect Concepts* for information on security and certificates files in general, and *TIBCO BusinessConnect Trading Partner Administration* for instructions on how to set security credentials.

Chapter 6 **Managing SOAP Operations**

This chapter explains how to manage interfaces, versions, and operations.

Topics

- [Overview, page 74](#)
- [Creating a SOAP Interface, page 76](#)
- [Adding a Version to an Interface, page 78](#)
- [Adding an Operation to a Version, page 79](#)
- [SOAP Operation Properties, page 80](#)
- [Importing SOAP Interfaces, page 90](#)
- [Exporting SOAP Interfaces, page 91](#)
- [Modifying SOAP Operations, page 92](#)

Overview

TIBCO BusinessConnect SOAP Protocol uses interfaces, versions, and operations to categorize and define actions in an electronic business transaction.

SOAP Interfaces

An interface is a category of operations, for example, Manage Purchase Order. This category includes a group of related operations, such as the following operations:

- Create Purchase Order
- Change Purchase Order
- Cancel Purchase Order

Each interface has a version number. Administrator can preserve older versions of interfaces when creating newer ones. Each interface contains some specific operations.

For example, a Manage Purchase Order interface could have the following definition:

The purpose of the interface is to specify the purchase order management process between trading partners. The management process includes the creation, change, and cancellation Business Document. All purchase order acknowledgments of acceptance are "substantive acceptance". A substantive acceptance returns some part of the original business document without modifications.

SOAP Operations

Each operation defines an action. For example, the definition of a Create Purchase Order operation might include the following:

- The operation description and transaction type. Transaction type might be notify or synchronous request-response.
- The transaction type request parameters, such as the schemas for header and body element validation, and signing and encryption options for incoming and outgoing requests.
- The operation response parameters, such as the schemas for header and body element validation, signing and encryption options for outgoing and incoming response.

Each installation can be the initiator for certain interfaces. At the same time, the responder can also be the initiator for the same or other interfaces.

Interfaces are defined in terms of their input and output, with no inherent initiator or responder. Trading partners can exchange interfaces by exporting or importing them.

After interfaces and operations have been set up, they must be associated with protocol bindings.

For step-by-step instructions, see [Configuring SOAP Agreement Protocol Binding](#) on page 106.

Creating a SOAP Interface

Before you can create versions and operations, you must create or import a SOAP interface.

1. In TIBCO Administrator, expand **BusinessConnect > Operations Editor**.

In the Operations Editor window, group the available installed protocols by **Plug-in** or **None**, as shown in [Figure 14](#).

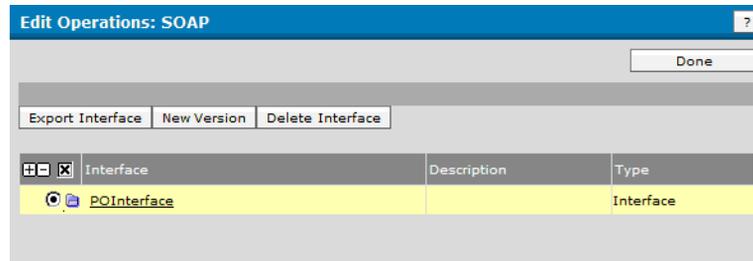
Figure 14 Configuring New Operation for SOAP



By using this window, you can do the following operations:

- **Import** Import a new operation by clicking **Import**.
 - **Export** Export the existing operations by clicking **SOAP**, and then click **Export**.
2. Click the **SOAP** link in the right panel.
 3. Click **New Interface**.
 4. In the New Interface window, perform the following steps:
 - a. In the **Name** field, type an interface name, such as **POInterface**.
 - b. In the **Description** field, type a brief description for this interface. This is optional.
 5. Click **Save**.

After the configuration, the new interface **POInterface** is created, as shown in [Figure 15](#).

Figure 15 New Interface POInterface

Adding a Version to an Interface

To add a version to an interface, perform the following steps:

1. Select an interface, such as **POInterface**.
2. Click **New Version**.
3. In the New Version window:
 - a. In the **Name** field, type a version name, such as **1.0**.
 - b. In the **Description** field, type a brief description for this version. This is optional.
4. Click **Save**.

After the configuration, the new version **1.0** is created, as shown in [Figure 16](#).

Figure 16 New Interface Version



Adding an Operation to a Version

To add an operation to a version, perform the following steps:

1. Select a version.
2. Click **New Operation**.
3. In the New Operation window, select **Notify** or **Synchronous Request-Response** from the **Operation Type** list.
4. Click **OK**.

Depending on the operation you have selected, this window has the following content:

- The New Notify Operation window contains the **Notify Operation** and **Notify Request Action** tabs.
- The New Synchronous Request-Response Operation window contains the **Synchronous Request-Response Operation**, **Request Action**, and **Response Action** tabs.

SOAP Operation Properties

Signing, encryption, and UsernameToken authentication can be enabled for an operation by using the **Notify Request Action** tab for the Notify operation, and by using the **Request Action** and **Response Action** tabs for the synchronous Request-Response operation. Both the request and response messages can be signed, encrypted, and authenticated.

Details are explained in the fields **Require Digital Signature**, **Require Content Encryption**, and **Require UsernameToken Authentication** for these tabs:

- [Notify Request Action Tab on page 81](#)
- [Request Action Tab on page 84](#)
- [Response Action Tab on page 87](#)

See [Document Security Tab on page 115](#) for information on how to set certificates and keys for signing and encryption.



If the **Require Digital Signature**, **Require Content Encryption**, and **Require UsernameToken Authentication** check boxes are cleared, these security elements are not applied to the outbound messages. However, for inbound messages, if certain check box is cleared on inbound side operation configuration, whether the specific security feature is applied or not, the inbound messages are received.

Notify Operation

You can use two tabs for Notify operation: the **Notify Operation** tab and **Notify Request Action** tab.

Notify Operation Tab

Use the **Notify Operation** tab to specify general information about the operation that BusinessConnect receives.

Table 4 *Notify Operation Tab*

Field	Description
Name	The name of the operation.
Description	The description of the operation.

Table 4 *Notify Operation Tab (Cont'd)*

Field	Description
SOAP Action	This field specifies the value of the SOAP action HTTP request header field of the SOAP message. It can be used to indicate the intent of the SOAP HTTP request and its value is a URI identifying the intent. See the SOAP 1.1 and 1.2 specifications at http://www.w3.org/TR/soap .

Notify Request Action Tab

Use the **Notify Request Action** tab to specify how BusinessConnect processes a message.

Table 5 *Notify Request Action Tab (Sheet 1 of 3)*

Field	Description
Name	The name of the operation.
Description	The description of the operation.
Direction	Initiator to responder (preset).
Security Configurations	
Require Digital Signature	If selected, you can use this option to sign the outgoing messages and force the incoming messages to be signed.
Require Content Encryption	If selected, you can use this option to encrypt the outgoing messages and force the incoming messages to be encrypted.
Require UsernameToken Authentication	<p>If selected, you can use this option to do the following operations:</p> <ul style="list-style-type: none"> Send the outbound action message with WSS:UsernameToken where the user name and password are provided by the private process, which is the message producer. Require that the inbound message of the same operation must also carry the WSS:UsernameToken element to be used for the inbound message authentication. If the inbound message does not have the WSS:UsernameToken element, the message is rejected with an error message being replied to the trading partner. <p>If your trading partners use WSS:UsernameToken to authenticate the message they send to you, you must define the external users associated with these trading partners.</p> <p>See the information on how to create an external user and associate this user with a particular trading partner in the chapter that explains User Access Management in <i>TIBCO BusinessConnect Trading Partner Administration</i>.</p>

Table 5 Notify Request Action Tab (Sheet 2 of 3)

Field	Description
Validation Configurations	
Validate Message	<p>Validate the message for the following cases:</p> <ul style="list-style-type: none"> • For an initiator, validate the request when it comes from the initiator private process and goes to a trading partner. • For a responder, validate the request when it comes from the initiator trading partner and goes to the responder private process. <p>When this field is selected, at least one of the schemas, body or header must be populated. When a schema is present, the corresponding content, body or header must conform to that schema. If the schema is absent, no validation is performed.</p> <p>See Appendix B, Schema Validation, on page 191.</p>
SOAP Header Validation Schema Name	<p>The XSD file that TIBCO BusinessConnect uses to validate the header part of the SOAP message. This header part contains the customized header elements that both sides of the trading partner agree to put into the SOAP header. All the customized header elements must have a root element.</p> <p>The XSD file must have a namespace declaration.</p> <p>If no file is displayed, you can add a schema by clicking on the change link, browse to the location of another schema, and load it for validation. If an existing schema file is displayed, you can replace it by using the change link, or remove it by using the remove link.</p> <p>Note: This schema is only used for validating the customized SOAP header. The partyInfo, security and WS-Addressing elements are not included.</p>
SOAP Header Root Element Name	<p>The top-level element name for the customized SOAP header element.</p> <p>This name cannot contain spaces. This value is used only for importing the schema into the TIBCO BusinessConnect palette and is not used during schema validation.</p>
SOAP Body Validation Schema Name	<p>The XSD file that TIBCO BusinessConnect uses to validate the body part of the SOAP message. Trading partners use the body element to carry the business document they exchange with each other.</p> <p>The XSD file must have a namespace declaration.</p> <p>If no file is displayed, you can add a schema by clicking on the change link, browse to the location of another schema, and load it for validation. If an existing schema file is displayed, you can replace it by using the change link, or remove it by using the remove link.</p>

Table 5 Notify Request Action Tab (Sheet 3 of 3)

Field	Description
SOAP Body Root Element Name	The top-level element name for your business document. This name cannot contain spaces. This value is used only for importing the schema into the TIBCO BusinessConnect palette and is not used during schema validation.



See [Appendix B, Schema Validation](#), on page 191 for information on validating schemas.

Synchronous Request-Response Operation

You can use three tabs for Synchronous Request-Response operation: **Synchronous Request-Response Operation** tab, **Request Action** tab, and **Response Action** tab.

Synchronous Request-Response Operation Tab

Use the **Synchronous Request-Response Operation** tab to specify general information about the operation.

Table 6 Synchronous Request-Response Operation Tab

Field	Description
Name	The name of the operation.
Description	The description of the operation.
SOAP Action	This field specifies the value of the SOAP action HTTP request header field of the SOAP message. It can be used to indicate the intent of the SOAP HTTP request and its value is a URI identifying the intent. See the SOAP 1.1 and 1.2 specifications at http://www.w3.org/TR/soap .

Request Action Tab

Use the Synchronous Request-Response **Request Action** tab to specify how BusinessConnect processes a request.

Table 7 Request Action Tab (Sheet 1 of 4)

Field	Description
Name	The name of the operation.
Description	The description of the operation.
Direction	Initiator to responder (preset).
Security Configurations	
Require Digital Signature	If selected, you can use this option to sign the outgoing requests and force the incoming requests to be signed.
Require Content Encryption	If selected, you can use this option to encrypt the outgoing requests and force the incoming requests to be encrypted.

Table 7 Request Action Tab (Sheet 2 of 4)

Field	Description
Require UsernameToken Authentication	<p>If selected, you can use this option to do the following operations:</p> <ul style="list-style-type: none"> Send the outbound action message with <code>WSS:UsernameToken</code> where the user name and password are provided by the private process, which is the message producer. Require that the inbound message of the same operation must also carry the <code>WSS:UsernameToken</code> element to be used for the 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 replied to the trading partner. <p>If your trading partners use <code>WSS:UsernameToken</code> to authenticate the message they send to you, you must define the external users associated with these trading partners.</p> <p>See the information on how to create an external user and associate this user with a particular trading partner in the chapter that explains User Access Management in <i>TIBCO BusinessConnect Trading Partner Administration</i>.</p>
NHIN/esMD Authorization Framework	<p>This check box is used only when you want to use the SAML assertion feature in conformance with NHIN/esMD Authorization Framework. CAQH CORE is a standard that uses this feature in the exchange of healthcare documents.</p> <ul style="list-style-type: none"> Select this check box to use SAML 2.0 assertion for request messages in conformance with NHIN/esMD Authorization Framework. You must also select the Require Digital Signature check box, so that the messages are digitally signed with SAML assertion. When this feature is used, provide all the required values in the <code>saml</code> node from the private process, such as the <code>AttributeStatement</code> and <code>AuthzDecisionStatement</code> elements that are required by the NHIN/esMD Authorization Framework. Errors occur if any of the required values is missing. <p>Note: On inbound side, if this check box is selected, all the SAML elements required by the NHIN/esMD Authorization Framework are checked, and errors occur if any of the required elements is missing. If this check box is not selected, the inbound messages are still processed but requirement check is not performed.</p>

Validation Configurations

Table 7 Request Action Tab (Sheet 3 of 4)

Field	Description
Validate Message	<p>Validate the message for the following cases:</p> <ul style="list-style-type: none"> • For an initiator, validate the request when it comes from the initiator private process and goes to a trading partner. • For a responder, validate the request when it comes from the initiator trading partner and goes to the responder private process. <p>When this field is selected, at least one of the schemas, body or header must be populated. When a schema is present, the corresponding content, body or header must conform to that schema. If the schema is absent, no validation is performed.</p> <p>See Appendix B, Schema Validation, on page 191.</p>
SOAP Header Validation Schema Name	<p>The XSD file that TIBCO BusinessConnect uses to validate the header part of the SOAP request. This header part contains the customized header elements that both sides of the trading partner agree to put into the SOAP header. All the customized header elements must have a root element.</p> <p>The XSD file must have a namespace declaration.</p> <p>If no file is displayed, you can add a schema by clicking on the change link, browse to the location of another schema, and load it for validation. If an existing schema file is displayed, you can replace it by using the change link, or remove it by using the remove link.</p> <p>Note: This schema is only used for validating the customized SOAP header. The partyInfo, security and WS-Addressing elements are not included.</p>
SOAP Header Root Element Name	<p>The top-level element name for the customized header elements.</p> <p>This name cannot contain spaces. This value is used only for importing the schema into the TIBCO BusinessConnect palette and is not used during schema validation.</p>
SOAP Body Validation Schema Name	<p>The XSD file that TIBCO BusinessConnect uses to validate the body part of the SOAP request. Trading partners use the body element to carry the business document they exchange with each other.</p> <p>The XSD file must have a namespace declaration.</p> <p>If no file is displayed, you can add a schema by clicking on the change link, browse to the location of another schema, and load it for validation. If an existing schema file is displayed, you can replace it by using the change link, or remove it by using the remove link.</p>

Table 7 Request Action Tab (Sheet 4 of 4)

Field	Description
SOAP Body Root Element Name	The top-level element name for your business document. This name cannot contain spaces. This value is used only for importing the schema into the TIBCO BusinessConnect palette and is not used during schema validation.

Response Action Tab

Use the Synchronous Request-Response **Response Action** tab to specify how BusinessConnect processes a response.

Table 8 Response Action Tab (Sheet 1 of 3)

Field	Description
Name	The name of the operation (response).
Description	The description of the operation.
Direction	Responder to initiator (preset).
Security Configurations	
Require Digital Signature	If selected, you can use this option to sign the outgoing responses and force the incoming responses to be signed.
Require Content Encryption	If selected, you can use this option to encrypt the outgoing responses and force the incoming responses to be encrypted.
Require UsernameToken Authentication	<p>If selected, you can use this option to do the following operations:</p> <ul style="list-style-type: none"> Send the outbound action message with <code>WSS:UsernameToken</code> where the user name and password are provided by the private process, which is the message producer. Require that the inbound message of the same operation must also carry the <code>WSS:UsernameToken</code> element to be used for the 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 replied to the trading partner. <p>If your trading partners use <code>WSS:UsernameToken</code> to authenticate the message they send to you, you must define the external users associated with these trading partners.</p> <p>See the information about how to create an external user and associate this user with a particular trading partner in the chapter that explains User Access Management in <i>TIBCO BusinessConnect Trading Partner Administration</i>.</p>

Table 8 Response Action Tab (Sheet 2 of 3)

Field	Description
NHIN/esMD Authorization Framework	<p>This check box is used only when you want to send a response to a request with SAML assertion in conformance with NHIN/esMD Authorization Framework. The response contains a signature confirmation defined in this framework.</p> <ul style="list-style-type: none"> • Select this check box to send responses with signature confirmation. • When you select this check box, do not select the Require Digital Signature check box. <p>Note: If the NHIN/esMD Authorization Framework check box is cleared, the signature confirmation is not applied to the responses.</p>
Validation Configurations	
Validate Message	<p>Validate the message for the following cases:</p> <ul style="list-style-type: none"> • For an initiator, validate the request when it comes from the initiator private process and goes to a trading partner. • For a responder, validate the request when it comes from the initiator trading partner and goes to the responder private process. <p>When this field is selected, at least one of the schemas, body or header must be populated. When a schema is present, the corresponding content, body or header must conform to that schema. If the schema is absent, no validation is performed.</p> <p>See Appendix B, Schema Validation, on page 191.</p>
SOAP Header Validation Schema Name	<p>The XSD file that TIBCO BusinessConnect uses to validate the header part of the SOAP response. This header part contains the customized header elements that both sides of the trading partner agree to put into the SOAP header. All the customized header elements must have a root element.</p> <p>The XSD file must have a namespace declaration.</p> <p>If no file is displayed, you can add a schema by clicking on the change link, browse to the location of another schema, and load it for validation. If an existing schema file is displayed, you can replace it by using the change link, or remove it by using the remove link.</p> <p>Note: This schema is only used for validating the customized SOAP header. The partyInfo, security and WS-Addressing elements are not included.</p>
SOAP Header Root Element Name	<p>The top-level element name for the customized SOAP header element.</p> <p>This name cannot contain spaces. This value is used only for importing the schema into the TIBCO BusinessConnect palette and is not used during schema validation.</p>

Table 8 Response Action Tab (Sheet 3 of 3)

Field	Description
SOAP Body Validation Schema Name	<p>The XSD file that TIBCO BusinessConnect uses to validate the body part of the SOAP response. Trading partners use the body element to carry the business document they exchange with each other.</p> <p>The XSD file must have a namespace declaration.</p> <p>If no file is displayed, you can add a schema by clicking on the change link, browse to the location of another schema, and load it for validation. If an existing schema file is displayed, you can replace it by using the change link, or remove it by using the remove link.</p>
SOAP Body Root Element Name	<p>The top-level element name for your business document.</p> <p>This name cannot contain spaces. This value is used only for importing the schema into the TIBCO BusinessConnect palette and is not used during schema validation.</p>
Private Process Wait (seconds)	<p>Time to wait for the response from the private process before timeout.</p> <p>The default is 3600 seconds (60 minutes).</p>



See [Appendix B, Schema Validation](#), on page 191 for information on validating schemas.

Importing SOAP Interfaces

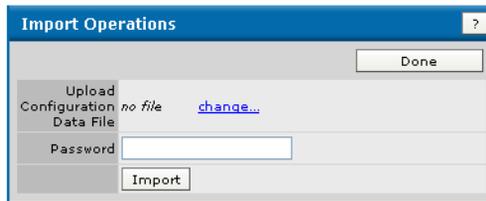
You might want to import one or more interfaces from a trading partner so that you do not have to re-create an interface.

To import an interface, perform the following steps:

1. Expand **BusinessConnect > Operations Editor**.
2. Click **Import**.

The Import Operations window is displayed, as shown in [Figure 17](#).

Figure 17 Importing Operations



3. Click the **Change** link to upload a configuration data file.
4. In the Change File window, click **Browse** to navigate to a directory containing an interface.
5. Select the interface.
6. Click **Open** and then click **OK**.
7. Type a password. This is optional.
8. Click **Import**.
9. Click **Done**.

Sample Interfaces

TIBCO BusinessConnect SOAP Protocol provides the following sample interfaces that can be imported into the BusinessConnect configuration store:

- `TIBCO_HOME/bc/version_number/protocols/soap/samples/server/operations.csx`
- `TIBCO_HOME/bc/version_number/protocols/soap/samples/client/operations.csx`
- `TIBCO_HOME/bc/version_number/protocols/soap/samples/caqh/caqh.csx`

Exporting SOAP Interfaces

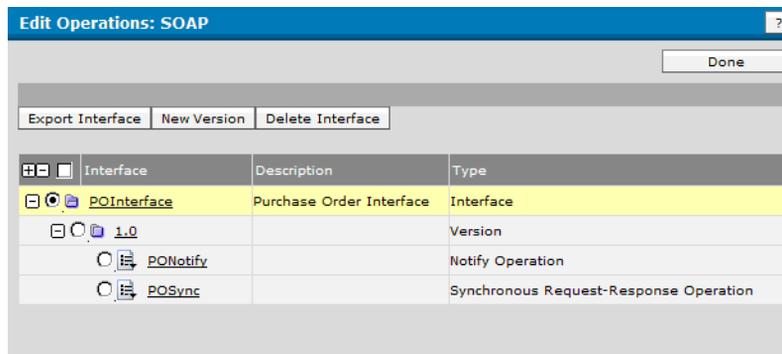
You might want to send an interface to a trading partner so that the partner does not have to re-create an interface.

To export an interface, perform the following steps:

1. Expand **BusinessConnect > Operations Editor**.
2. Click the **SOAP** link from the **Protocol** list in the right panel.
3. Expand **Interface** and **Version**.

The specific version and operations are displayed, as shown in [Figure 18](#).

Figure 18 Exporting SOAP Interface, Version, or Operation



4. Select the **Interface**, **Version** or the specific **operation** you want to export.
5. Click **Export Interface**, **Export Version**, or **Export Operation**, depending on what you selected in [step 4](#).
6. The Export Operations window is displayed for the selected interface, version, or operation.

To save the exported data, set the password. This is optional.

7. Click **Export Data**.
8. The pop-up dialog is displayed with the suggested file name `operations.csx`.
9. Browse to a desired place on your machine and click **Save**.
10. Click **Done** to finish export.

Modifying SOAP Operations

You can modify SOAP operations by defining the SOAP body validation schema name and SOAP body root element name. This must be done on the initiator and responder BusinessConnect installations.

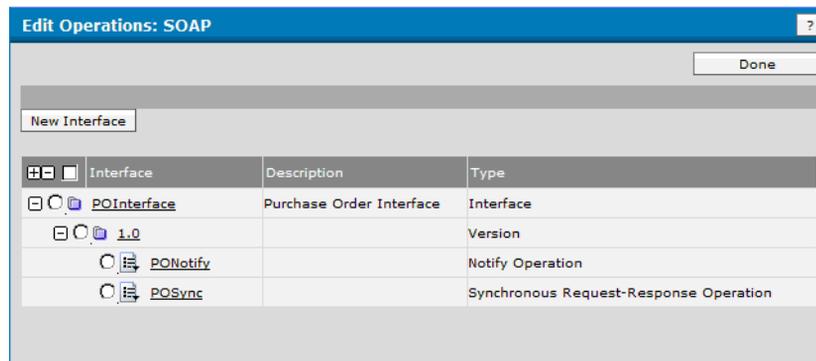
Modifying the Notify Operation

To modify the operation, perform the following steps:

1. Expand **BusinessConnect > Operations Editor**.
2. Click the **SOAP** link from the **Protocol** list in the right panel.
3. Click  to expand the interface tree.

The expanded interface tree is displayed, as shown in [Figure 19](#).

Figure 19 Expanding the Interface



4. Click the *PONotify* link.
5. Click the **Notify Request Action** tab.
6. Next to **SOAP Body Validation Schema Name**, click **change**.
7. Select **Uploaded File** from the **Type** list.
8. Next to the **Uploaded File** field, click **Browse** and select the file to upload, such as the commonpo.xsd file in the `TIBCO_HOME/bc/version_number/protocols/soap/samples/schemas/tutorial` directory.
9. Click **Open** and then click **OK**.

The file is uploaded in the **Notify Request Action** tab of the Notify Operation, as shown in [Figure 20](#).

Figure 20 Notify Operation

10. Type the name in the **SOAP Body Root Element Name** field.
11. Click **Save**.

Modifying the Sync Operation

To modify the operation, perform the following steps:

1. Expand **BusinessConnect > Operations Editor**.
2. Click the **SOAP** link from the **Protocol** list in the right panel.
3. Click  to expand the interface tree.
4. Click the **POSync** link.
5. Click the **Request Action** tab.
6. Next to **SOAP Body Validation Schema Name**, click **change**.
7. Select **Uploaded File** from the **Type** list.
8. Next to the **Uploaded File** field, click **Browse** and select the file to upload, such as the commonpo.xsd file in the `TIBCO_HOME/bc/version_number/protocols/soap/samples/schemas/tutorial` directory.

- Click **Open** and then click **OK**.

The file is uploaded in the **Request Action** tab of the Synchronous Request-Response Operation, as shown in [Figure 21](#).

Figure 21 Synchronous Request-Response Operation: Request Action

The screenshot shows a dialog box titled "Edit Synchronous Request-Response Operation for Version POInterface/1.0 : POSync". It has three tabs: "Synchronous Request-Response Operation", "Request Action", and "Response Action". The "Request Action" tab is selected. The "General" section is expanded, showing the following fields and options:

Name	request
Description	
Direction	Initiator to Responder
Security Configurations	
Require Digital Signature	<input type="checkbox"/>
Require Content Encryption	<input type="checkbox"/>
Require UsernameToken Authentication	<input type="checkbox"/>
NHIN/esMD Authorization Framework	<input type="checkbox"/>
Validation Configurations	
Validate Message	<input checked="" type="checkbox"/>
SOAP Header Validation Schema Name	poheader.xsd change remove
SOAP Header Root Element Name	
SOAP Body Validation Schema Name	commonpo.xsd change remove
SOAP Body Root Element Name	CommonPO

- Type the name in the **SOAP Body Root Element Name** field.
- Click **Save**.
- Click the **POSync** link again.
- Click the **Response Action** tab.
- Next to **SOAP Body Validation Schema Name**, click **change**.
- Select **Uploaded File** from the **Type** list.
- Next to the **Uploaded File** field, click **Browse** and select the file to upload, such as the commoninvoice.xsd file in the `TIBCO_HOME/bc/version_number/protocols/soap/samples/schemas/tutorial` directory.
- Click **Open** and then click **OK**.

The file is uploaded in the **Response Action** tab of the Synchronous Request-Response Operation, as shown in [Figure 22](#).

Figure 22 Synchronous Request-Response Operation: Response Action

The screenshot shows a dialog box titled "Edit Synchronous Request-Response Operation for Version POInterface/1.0 : POSync". It has three tabs: "Synchronous Request-Response Operation", "Request Action", and "Response Action". The "Response Action" tab is selected. Below the tabs is a "General" section with the following fields:

Name	response
Description	
Direction	Responder to Initiator
Security Configurations	
Require Digital Signature	<input type="checkbox"/>
Require Content Encryption	<input type="checkbox"/>
Require UsernameToken Authentication	<input type="checkbox"/>
NHIN/esMD Authorization Framework	<input type="checkbox"/>
Validation Configurations	
Validate Message	<input checked="" type="checkbox"/>
SOAP Header Validation Schema Name	poheader.xsd change remove
SOAP Header Root Element Name	
SOAP Body Validation Schema Name	commoninvoice.xsd change remove
SOAP Body Root Element Name	CommonInvoice
Private Process Wait (seconds)	3600

18. Type the name in the **SOAP Body Root Element Name** field.

19. Click **Save**.

Chapter 7 **Setting Up Trading Hosts and Partners**

This chapter explains how to set up trading hosts and partners in TIBCO BusinessConnect SOAP Protocol.

Topics

- [Configuring the SOAP Protocol for a Host, page 98](#)
- [Configuring the SOAP Protocol for a Partner, page 100](#)

Configuring the SOAP Protocol for a Host

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

1. In TIBCO Administrator, expand **BusinessConnect > Participants**.
2. Click a *host participant* link in the right panel.
3. Click the **Protocols** tab.

If SOAP is not displayed in the list of protocols, perform the following steps:

- a. Click **Enable**.
- b. Select the **SOAP** check box.
- c. Click **OK**.
4. Click the **SOAP** link.
5. In the Edit Enabled Protocol window, set properties for SOAP by using [Table 9](#).

Table 9 Configuring a Host: General Tab

Field	Description
Default Domain Identity	The default domain identity to use for this host. Select from the list of domain identities created in the Domain Identities List window. See on page 98 to add or edit a domain identity.
Valid Email Address List	This field is not used for TIBCO BusinessConnect SOAP Protocol.
Allow Anonymous SOAP Messages	Anonymous SOAP messages can pass through. See Receiving SOAP Messages without TIBCO BusinessConnect-Specific Headers on page 135 .

6. Click **Save**.

Adding a Domain Identity for a Host

To add a domain identity, perform the following steps:

1. In the Edit Enabled Protocol window, click the **Add New** link.
2. In the Domain Identity List window, click **Add New**.
3. Select a domain from the **Domain** list and type an identity in the **Identity** field.
4. Click **Save**.
5. Click **OK**.

Configuring the SOAP Protocol for a Partner

To configure the SOAP protocol for a trading partner, perform the following steps:

1. In TIBCO Administrator, expand **BusinessConnect > Participants**.
2. Click a *partner participant* link in the right panel.
3. Click the **Protocols** tab.

If SOAP is not displayed in the list of protocols, perform the following steps:

- a. Click **Enable**.
 - b. Select the **SOAP** check box.
 - c. Click **OK**.
4. Click the **SOAP** link. The following configuration options are available:
 - **General Tab** See [Configuring General Properties for a Partner on page 100](#).
 - **Transports Tab** See [Configuring Transports for a Partner on page 103](#).
 5. Click **Save**.

Configuring General Properties for a Partner

Use the **General** tab to configure general information about your trading partner.

Table 10 Configuring a Partner: General Tab (Sheet 1 of 3)

Field	Description
Default Domain Identity	The default domain identity to use for this host. Select from the list of domain identities created in the Domain Identities List window. See Adding a Domain Identity for a Partner on page 102 to add or edit a domain identity.
Valid Email Address List	This field is not used for TIBCO BusinessConnect SOAP Protocol.
Duplicate Detection for Outbound	Prevents duplicate messages from being sent to your trading partner and sends an error message to the private process. See Configuring the SOAP Protocol for a Partner on page 100 .
Add BusinessConnect Specific Header	Adds a special header to any messages going to this trading partner. See TIBCO BusinessConnect SOAP Protocol Public Messages on page 140 .

Table 10 Configuring a Partner: General Tab (Sheet 2 of 3)

Field	Description
Allow Anonymous SOAP Messages From This Trading Partner	<p>Anonymous SOAP messages from this trading partner can pass through.</p> <p>See Receiving SOAP Messages without TIBCO BusinessConnect-Specific Headers on page 135.</p>
SOAP Namespace Prefix	<p>You can configure the prefix for the envelope namespace. The string SOAP-ENV is still the default, but you can replace it by any string of your choice. For example, a SOAP message can have the following format:</p> <pre data-bbox="435 494 899 720"> <abc:Envelope xmlns:abc="http://www.w3.org/2003/05/soap-envelope"> <abc:Header> </abc:Header> <abc:Body> </abc:Body> </abc:Envelope> </pre> <p>where <code>abc</code> replaces the previously pre-defined string SOAP-ENV.</p> <p>Note: The value <code>Envelope</code> is not applied to SOAP faults, which always have SOAP-ENV as the default namespace.</p>
SOAP Version	<p>TIBCO BusinessConnect SOAP Protocol supports both SOAP versions 1.1 and 1.2 by selecting SOAP 1.1 Protocol or SOAP 1.2 Protocol in the SOAP Version field. The default value is SOAP 1.1 Protocol. The configuration is used to set the outbound messages.</p> <p>For outbound messages, you can use SOAP protocol to send the request in SOAP version 1.1 or 1.2. You can also use SOAP protocol to send the response in the same version of the request from the trading partner. For inbound messages, SOAP protocol can process the request or response in version 1.1 or 1.2.</p>
Send CAQH Message To EDI Protocol	<p>When you select this check box, BusinessConnect forwards inbound CAQH Batch/Realtime Request/Response to EDI protocol.</p>

Table 10 Configuring a Partner: General Tab (Sheet 3 of 3)

Field	Description
SOAP MTOM Enabled	<p data-bbox="386 232 1272 336">TIBCO BusinessConnect SOAP Protocol supports MTOM by selecting the SOAP MTOM Enabled check box. The configuration is used to set the outbound messages.</p> <p data-bbox="386 354 1272 475">If selected, the outbound messages are processed by using MTOM standard to optimize the transmission of the binary data in XML infoset. The binary data is supposed to be base64 encoded, which is included in the XML document specified in the <code>body</code> node of the request private process message.</p> <p data-bbox="386 493 1272 614">For outbound messages, if the content of the binary data is not in base64 format in the <code>body</code> node of the request private process messages, the messages processed by MTOM might not be decoded correctly by the responder side which is also MTOM enabled.</p> <p data-bbox="386 631 1272 753">For inbound messages, if optimized messages by MTOM are detected, the messages are parsed and the original binary data extracted from the MTOM attachments is encoded into base64 format and put back into SOAP body, before the messages are further processed.</p> <p data-bbox="386 770 1272 805">Note:</p> <ul data-bbox="386 822 1272 1395" style="list-style-type: none"> <li data-bbox="386 822 1272 944">• If the SOAP MTOM Enabled check box is selected, the value under <code>xop > tagName > value</code> of the private process message must be populated according to the binary data element in the <code>body</code> node. If not, the request is rejected and the outbound process is ended. <li data-bbox="386 961 1272 1065">• If the SOAP MTOM Enabled check box is not selected, and the value under <code>xop > tagName > value</code> in the private process is provided, the request is rejected and the outbound process is ended. <li data-bbox="386 1083 1272 1204">• MTOM and attachments from the private process cannot be used together. If the SOAP MTOM Enabled check box is selected, and some attachments are also specified from the private process request message, the request is rejected and the outbound process is ended. <li data-bbox="386 1222 1272 1395">• When selecting the SOAP MTOM Enabled and Require Content Encryption check boxes to send outbound messages, the binary content which refers to the value under <code>xop > tagName > value</code> in the private process message is ignored. The encrypted data in the <code>CipherValue</code> element after encryption is used to construct the MTOM attachment. <p data-bbox="386 1413 1272 1442">See MTOM on page 22.</p>

Adding a Domain Identity for a Partner

To add a domain identity, perform the following steps:

1. In the Edit Enabled Protocol window, click the **Add New** link.
2. In the Domain Identity List window, click **Add New**.
3. Select a domain from the **Domain** list and type an identity in the **Identity** field.
4. Click **Save**.
5. Click **OK**.

Configuring Transports for a Partner

To add the transport for a partner, perform the following steps:

1. Click the **Transports** tab.
2. Click **Add**.

In the New Transport window, type data as explained in [Table 11](#).

Table 11 New Transport for a Partner

Field	Description
Name	Type the name for the transport.
Type	Select HTTP or HTTPS from the Type list.

3. Click **OK**.
4. In the New HTTP Transport window, type information as in [Table 3](#).
5. Click **Save** three times.

See the information about setting up HTTP/S for a trading partner in *TIBCO BusinessConnect Trading Partner Administration*.

Chapter 8 **Configuring Business Agreements**

This chapter explains how to configure business agreements and protocol bindings for TIBCO BusinessConnect SOAP Protocol.

Topics

- [Configuring SOAP Agreement Protocol Binding, page 106](#)
- [Operation Bindings Tab, page 107](#)
- [Document Security Tab, page 115](#)
- [Transports Tab, page 116](#)
- [Overriding Participant Settings, page 117](#)

Configuring SOAP Agreement Protocol Binding

To configure a SOAP agreement protocol binding, perform the following steps:

1. In TIBCO Administrator, expand **BusinessConnect > Business Agreements**.
2. Click **New**.
3. Select a *Host* and a *Partner* from the **Host Party** and **Partner Party** areas.

Be sure that both trading partners have the **SOAP** protocol enabled in the **Protocols** list. If SOAP is not listed, see [Configuring the SOAP Protocol for a Host on page 98](#) or [Configuring the SOAP Protocol for a Partner on page 100](#).

4. Click **OK**.
5. Click the *New Agreement* link just created.
6. Click the **General** tab, verify that the **Valid** check box is selected, which means the current business agreement is valid.

If you want to specify the time period for which this agreement is valid, clear the **Valid** check box, and then use the **Start Date** and **End Date** lists to specify the agreement duration.

7. Click **Add Protocol Binding**.
8. Select **SOAP**.
9. Click **OK**.
10. Click the **SOAP** link from the **Agreement Protocol Binding** list.

11. Configure each of the following tabs:

- **Operation Bindings** See [Operation Bindings Tab on page 107](#).
- **Document Security** See [Document Security Tab on page 115](#).
- **Transports** See [Transports Tab on page 116](#).
- **Show Advanced: Participant Configuration** See [Overriding Participant Settings on page 117](#).

Operation Bindings Tab

Use the **Operation Bindings** tab to configure the SOAP operations that each participant in a business agreement can initiate and respond to.

The **Initiating Operations** and **Responding Operations** areas list the activities that the host and partner can initiate and the partner/host can respond to.

The following properties apply to all the activities that you import in **Operations Editor**, as shown in [Table 12](#).

Table 12 Protocol Bindings Tab

Field	Description
Allow All Operations	<p>Select this check box to enable all operations between participants. You can modify the behavior by binding the operations in the Initiating Operations and Responding Operations areas.</p> <p>When the check box is cleared, it is required that the operation in the Responding Operations area be explicitly bound in the Initiating Operations area also. Otherwise, the following error message is displayed:</p> <p style="padding-left: 20px;">Sending SOAPFault. Fault String: TP not allowed access, reason: Operation not bound for trading partner.</p> <p>This check box is cleared by default for the SOAP protocol.</p>
Non Repudiation Logging	<p>This check box is selected by default for the SOAP protocol.</p> <p>See Non-Repudiation Logs on page 127.</p>

Initiating Operations

Add Operation Binding	<p>The Initiating Operations area lists the activities that the host can initiate and the partner can respond to. To bind an operation in this area, perform the following steps:</p> <ol style="list-style-type: none"> 1. Click Add Operation Bindings. 2. Click  to expand the interface tree and select the <i>operation</i>. 3. Click OK.
-----------------------	---

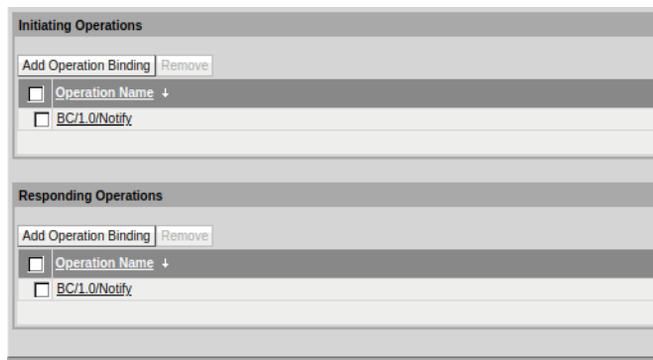
Responding Operations

Table 12 Protocol Bindings Tab (Cont'd)

Field	Description
Add Operation Binding	<p>The Responding Operations area lists the activities that the partner can initiate and the host can respond to. To bind an operation in this area, perform the following steps:</p> <ol style="list-style-type: none"> 1. Click Add Operation Bindings. 2. Click  to expand the interface tree and select the <i>operation</i>. 3. Click OK.

A sample of the operation bindings for the host and partner is displayed, as shown in [Figure 23](#).

Figure 23 Operation Bindings Added for Host and Partner



Editing an Operation Binding for the Host

You can use the **Operation Settings**, **Action Settings**, and **Transports** tabs to override the operation, action, and transport settings of the host.

Operation Settings Tab

To override the operation settings of the host, perform the following steps:

1. Click a *Notify* or a *Synchronous Request-Response* operation binding link in the **Initiating Operations** area.
2. Click the **Operation Settings** tab.

3. Override the operation settings by using information in [Table 13](#).

Table 13 *Override Outbound Settings: Operation Settings Tab*

Field	Description
Override Operation Settings	To override the settings for a SOAP operation, select this check box and then select General from the list. The options selected here override those selected in SOAP Operation Properties on page 80 .
SOAP Action	You can use this function to override the SOAP action that you typed in Operations Editor for this operation.

4. Click **Save**.

Action Settings Tab

To override the action settings of the host, perform the following steps:

1. Click a *Notify* or a *Synchronous Request-Response* operation binding link in the **Initiating Operations** area.
2. Click the **Action Settings** tab.
3. Override the action settings by using information in [Table 14](#).

Table 14 *Override Outbound Settings: Action Settings Tab*

Field	Description
Override Action Settings	Select this check box and then select General from the list to override the action settings originally configured.
Security Configurations	
Require Digital Signature	If selected, this option signs the outgoing requests and forces the incoming requests to be signed. You can use this function to override the signature settings originally configured in Operations Editor .
Require Content Encryption	If selected, this option encrypts the outgoing requests and forces the incoming requests to be encrypted. You can use this function to override the encryption settings originally configured in Operations Editor .

Table 14 Override Outbound Settings: Action Settings Tab (Cont'd)

Field	Description
Require UsernameToken Authentication	<p>If selected, you can use this option to do the following operations:</p> <ul style="list-style-type: none"> • Send the outbound action message with WSS:UsernameToken where the user name and password are provided by the private process, which is the message producer. • Require that the inbound message of the same operation must also carry the WSS:UsernameToken element to be used for the inbound message authentication. If the inbound message does not have the WSS:UsernameToken element, the message is rejected with an error message being replied to the trading partner. <p>If your trading partners use WSS:UsernameToken to authenticate the message they send to you, you must define the external users associated with these trading partners.</p> <p>You can use this function to override the UsernameToken Authentication settings originally configured in Operations Editor.</p> <p>See the information about how to create an external user and associate this user with a particular trading partner in the chapter that explains User Access Management in <i>TIBCO BusinessConnect Trading Partner Administration</i>.</p>
NHIN/esMD Authorization Framework	<p>This check box is used only for synchronous request-response operation.</p> <p>If selected, SAML 2.0 assertion is used for request messages in conformance with NHIN/esMD Authorization Framework. You must also select the Require Digital Signature check box, so that the messages are digitally signed with SAML assertion.</p> <p>On inbound side, if this check box is selected, the incoming requests that contain all the required SAML elements as defined in the framework are checked.</p> <p>You can use this function to override settings of NHIN/esMD Authorization Framework originally configured in Operations Editor.</p>
Validation Configurations	
Validate Message	<p>When selected, any outbound message is validated.</p> <ul style="list-style-type: none"> • When this field is selected, at least one of the schemas must be populated in Operations Editor, such as body or header. • When the schema is present, the corresponding content must conform to that schema, such as body or header. If the schema is absent, no validation is performed. <p>You can use this function to override the validation settings originally configured in Operations Editor.</p> <p>See Appendix B, Schema Validation, on page 191.</p>

4. Click **Save**.

Transports Tab

To override the transports settings of the host, perform the following steps:

1. Click a *Notify* or a *Synchronous Request-Response* operation binding link in the **Initiating Operations** area.
2. Click the **Transports** tab.
3. Override the transport settings by using information in [Table 15](#).

Table 15 Override Outbound Settings: Transports Tab

Field	Description
Override Transports	Select this check box to override the settings for a transport.
Override Outbound Transports	
Primary Transport	Select the primary transport to override from the list: HTTP , HTTPS , or HTTPSCA .

4. Click **Save**.

Editing an Operation Binding for the Partner

You can use the **Operation Settings** and **Action Settings** tabs to override the operation and action settings of the partner.

Operation Settings Tab

To override the operation settings of the partner, perform the following steps:

1. Click a *Notify* or a *Synchronous Request-Response* operation binding link in the **Responding Operations** area.
2. Click the **Operation Settings** tab.

3. Override the operation settings by using information in [Table 16](#).

Table 16 *Override Inbound Settings: Operation Settings Tab*

Field	Description
Override Operation Settings	To override the settings for a SOAP operation, select this check box and then select General from the list. The options selected here override those selected in SOAP Operation Properties on page 80 .
SOAP Action	You can use this function to override the SOAP action that you typed in Operations Editor for this operation.

4. Click **Save**.

Action Settings Tab

To override the action settings of the partner, perform the following steps:

1. Click a *Synchronous Request-Response* operation binding link in the **Responding Operations** area.
2. Click the **Action Settings** tab.
3. Override the action settings by using information in [Table 17](#).

Table 17 *Override Inbound Settings: Action Settings Tab (Sheet 1 of 3)*

Field	Description
Override Action Settings	Select this check box and then select General from the list to override the action settings originally configured.
Security Configurations	
Require Digital Signature	If selected, this option signs the outgoing responses and forces the incoming responses to be signed. You can use this function to override the signature settings originally configured in Operations Editor .
Require Content Encryption	If selected, this option encrypts the outgoing responses and forces the incoming responses to be encrypted. You can use this function to override the encryption settings originally configured in Operations Editor .

Table 17 Override Inbound Settings: Action Settings Tab (Sheet 2 of 3)

Field	Description
Require UsernameToken Authentication	<p>If selected, you can use this option to do the following operations:</p> <ul style="list-style-type: none"> Send the outbound action message with WSS:UsernameToken where the user name and password are provided by the private process, which is the message producer. Require that the inbound message of the same operation must also carry the WSS:UsernameToken element to be used for the inbound message authentication. If the inbound message does not have the WSS:UsernameToken element, the message is rejected with an error message being replied to the trading partner. <p>If your trading partners use WSS:UsernameToken to authenticate the message they send to you, you must define the external users associated with these trading partners.</p> <p>You can use this function to override the UsernameToken Authentication settings originally configured in Operations Editor.</p> <p>See the information about how to create an external user and associate this user with a particular trading partner in the chapter that explains User Access Management in <i>TIBCO BusinessConnect Trading Partner Administration</i>.</p>
NHIN/esMD Authorization Framework	<p>This check box is used only for synchronous request-response operation.</p> <p>If selected, the responder can send a response to a request with SAML assertion in conformance with NHIN/esMD Authorization Framework. The response contains a signature confirmation defined in this framework.</p> <p>You can use this function to override settings of NHIN/esMD Authorization Framework originally configured in Operations Editor.</p>
Validation Configurations	
Validate Message	<p>When selected, any inbound message is validated.</p> <ul style="list-style-type: none"> When this field is selected, at least one of the schemas must be populated in Operations Editor, such as body or header. When the schema is present, the corresponding content must conform to that schema, such as body or header. If the schema is absent, no validation is performed. <p>You can use this function to override the validation settings originally configured in Operations Editor.</p> <p>See Appendix B, Schema Validation, on page 191.</p>

Table 17 *Override Inbound Settings: Action Settings Tab (Sheet 3 of 3)*

Field	Description
Private Process Wait (seconds)	Select to override the wait time originally configured. This is the amount of time the responder waits for the response from the private process. The default value is 3600 seconds (60 minutes). You can use this function to override the time settings originally configured in Operations Editor .

4. Click **Save**.

Document Security Tab

Information about this topic, see Document Security tab in *TIBCO BusinessConnect Trading Partner Administration*.

See also Security in *TIBCO BusinessConnect Concepts*.



TIBCO BusinessConnect SOAP Protocol supports SHA1, SHA256, SHA384 and SHA512 digest algorithm for signatures and DES3, AES-128, AES-192, AES-256, AES-128-GCM, AES-192-GCM, and AES-256-GCM encryption algorithms.



SAML 2.0 assertions only support Secure Hash Algorithm 1 (SHA1) digest algorithm.

Transports Tab

To configure the **Transports** tab, see [Table 18](#).

Table 18 *Transports Tab*

Field	Description
Outbound Transports for Host (Host to Trading Partner)	
Primary Transport	The following options are available: HTTP , HTTPS , HTTPSCA , or None . The primary transport is configured by using the steps described in Configuring Transports for a Partner on page 103 .
Client Authentication Identity for FTPS, HTTPSCA	Select one of the private keys that were assigned to the host. See <i>Managing Host Credentials in TIBCO BusinessConnect Trading Partner Administration</i> .
Client Authentication Identity for SSHFTP	Select one of the SSH private keys assigned to the host, which is used to support the SSHFTP transport. See <i>Managing Host Credentials in TIBCO BusinessConnect Trading Partner Administration</i> .
Allowed Inbound Transports for Partner (Trading Partner to Host)	
HTTP or HTTPS	Transport types are displayed. HTTP or HTTPS is selected in the deployment configuration. See the information about Setting Up HTTP/S for a Trading Partner in TIBCO BusinessConnect Trading Partner Administration .

Overriding Participant Settings

Use the participant configuration tabs to override the general settings for a participant per agreement protocol binding.

- To view the participant configuration tabs, click **Show Advanced**.

Host's Configuration and **Partner's Configuration** tabs are displayed, where *Host* and *Partner* are the participants in the business agreement.

- To hide the participant configuration tabs, click **Hide Advanced**.

Overriding Settings for the Host

To override settings for the host, perform the following steps:

1. Click the **Host's Configuration** tab.
2. Override settings by using information in [Table 19](#).

Table 19 Show Advanced: Host's Configuration

Field	Description
Override Settings	If this check box is selected, the settings specified for the host in Table 9, Configuring a Host: General Tab, on page 98 are overridden.
Default Domain Identity	Select a domain identity from the list. The domain identity is configured as explained in on page 98 .

3. Click **Save**.

Overriding Settings for the Partner

To override settings for the partner, perform the following steps:

1. Click the **Partner's Configuration** tab.
2. Override settings by using information in [Table 20](#).

Table 20 Show Advanced: Partner's Configuration

Field	Description
Override Settings	If this check box is selected, the settings specified for the partner in Table 10, Configuring a Partner: General Tab, on page 100 are overridden.

Table 20 Show Advanced: Partner's Configuration (Cont'd)

Field	Description
Default Domain Identity	Select a domain identity from the list. The domain identity is configured as explained in Adding a Domain Identity for a Partner on page 102 .

3. Click **Save**.

Chapter 9 **Viewing Logs**

This chapter discusses how to view audit, non-repudiation, and resend logs after conducting business transactions.

Topics

- [Overview, page 120](#)
- [Audit Logs, page 121](#)
- [Non-Repudiation Logs, page 127](#)
- [Resend Logs, page 129](#)

Overview

TIBCO BusinessConnect provides logs that are used to store messages processed through the system. These logs are as follows:

- Audit logs
- Resend logs
- Non-repudiation logs

Use the **Log Viewer** to search for SOAP transaction records.

Accessing Logs

To access a log, perform the following steps:

1. In TIBCO Administrator, expand **BusinessConnect** > **Log Viewer** in the left panel.
2. Select **SOAP** from the **Protocol** list.
3. Click one of the following buttons to select which log to view:
 - **Audit** Displays the audit log search options.
 - **Non-repudiation** Displays the non-repudiation log search options.
 - **Resendable Transactions** Displays the resend log search options.
 - **Resend History** Displays the resend history log search options.

See *TIBCO BusinessConnect Trading Partner Administration* for information on searching the logs.

Audit Logs

The audit log is used to store information about the messages and documents processed by BusinessConnect. You can use the audit log to follow the processing states of inbound or outbound documents. Some information types stored in the audit log include the following aspects:

- Sent and received documents
- Document originator
- Trading partner name
- Processing status
- Validation errors

Configuring an Audit Log for SOAP

To configure an audit log for SOAP, perform the following steps:

1. Expand **BusinessConnect > Log Viewer** in the left panel.
2. In the Log Viewer window, click the **SOAP** link.
3. Select the options of the audit log by using information in [Table 21](#).

Table 21 Audit Log: Search Filters

Column Name	Definition
Status	Select a specific status, such as ANY, COMPLETED, COMPLETED_WITH_ERROR, ERROR, or PENDING.
Connection	The database that you connect to. You can switch among multiple audit log databases from this list.
Date Range	Select a specific date range, such as One Day, One Week, One Month, One Year, or --Custom--. If --Custom-- is selected, additional fields for defining exact dates are available.

4. To configure the advanced search filters, click **Add**.

- Configure the advanced search settings by using information in [Table 22](#).

Table 22 *Audit Log: Advanced Search Filters*

Column	Definition
Save as Query	Type the name under which you want to save the query you define in this field.
Gateway Instance Information	Indicates which gateway instance transferred this message. Boolean search: is, contains, is not, is not like.
Trading Partner	The name of the trading partner. Boolean search: is, contains, is not, is not like.
Operation ID	The operation ID. Boolean search: is, contains, is not, is not like.
Transaction ID	The transaction ID. Boolean search: is, contains, is not, is not like.

Audit Log Views

You can view summary, transaction details, and state details information of audit logs.

Summary View

[Table 23](#) lists the columns that are displayed in the audit log.

Table 23 *Audit Log Columns*

Column	Definition
Date Group	Results are grouped according to the day the transactions occurred.
Time Stamp	When each of the transaction states is logged.
Trading Partner	The name of the trading partner.
Operation ID	The operation ID used for this operation. Example: POInterface/1.0/PONotify.
Transaction ID	The transaction ID for this transaction.

Transaction Details View

To view the details of a transaction, click  at the left of a specific time stamp.

The general information for this entry and a table with available information for each event in that transaction are displayed. The table columns are listed in [Table 24](#).

Each transaction is identified with its Process Instance Identifier.

Table 24 Transaction Detail Columns

Column	Definition
Time Stamp	When each of the transaction states is logged.
Status	Status of the message. Valid values: COMPLETED, ERROR, PENDING. These values come from the status field in certain tables.
State	The current state of the message. Example: REQUEST_FROM_PP. See State Field Values on page 123 for a list of the possible values.
Description	Description of the last action recorded for the message. Example: Sending Notification to Trading Partner.

State Details View

To view the details of a specific state, click  at the left of a transaction detail entry.

All information from the summary view and transaction details view for the transaction entry are displayed.

State Field Values

[Table 25](#) provides a list of the **State** field values for the initiator process.

Table 25 State Field Values for Initiator Process

Short Message	Definition	Comments
REQUEST_FROM_PP	Received request from private process.	Received notification from private process. Depending on the message types, different descriptive messages are logged.
REQUEST_SEND_TO_TP	Request or notification is about to be sent to trading partner.	This entry is logged before the request has been sent to the trading partner.

Table 25 State Field Values for Initiator Process (Cont'd)

Short Message	Definition	Comments
REQUEST_SENT_TO_TP	Request or notification sent to trading partner; received acknowledgment.	This entry is logged only after the request has been successfully posted to the trading partner. Either this entry or the next one is the last one logged for notification.
RESPONSE_FROM_TP	Received response from trading partner.	This entry is logged only after a response is received from the trading partner.
RESPONSE_TO_PP	Response sent to private process.	Logging is done after a response, such as an error message, was handed off to the private process. In case an error is returned, a descriptive message is logged instead.
PACKAGE_MSG	SOAP Message packaged successfully.	Logging is done after the SOAP message is packaged. For example, after the header elements, body elements, and attachments have been added. If security is enabled, the SOAP body is signed, encrypted, and authenticated.
UNPACKAGE_MSG	Message from the <trading partner> parsed to a SOAP message successfully.	Logging is done after the received HTTP message (request/response) is converted to a SOAP message and decrypted (if encrypted previously).
RESPONSE_DECRYPTED	Encrypted response message successfully decrypted.	Logging is done after the incoming message is decrypted successfully.
RESPONSE_VERIFIED	Signed response from the trading partner verified successfully.	Logging is done after the signed response message is verified successfully.
DOCUMENT_DENIED	Received request document is denied because of encryption, signing, or authentication permissions.	Logging is done when the incoming request or response is denied because of security permissions for this operation.

Table 25 State Field Values for Initiator Process (Cont'd)

Short Message	Definition	Comments
VALIDATE_REQUEST	Incoming request (<header/Body>) is validated.	Logging is done after the validation of the incoming or outgoing request. The description specifies if the validation was a success or failure.
VALIDATE_RESPONSE	Incoming response (<header/Body>) is validated.	Logging is done after the validation of the incoming or outgoing response. Description specifies if validation was a success or failure.

Table 26 provides a list of the **State** field values for the responder process:

Table 26 State Field Values for Responder Process

Short Message	Description	Comments
REQUEST_FROM_TP	Received request from trading partner.	Received notification from trading partner. Logging is done after having successfully received a notification or request.
REQUEST_TO_PP	Request sent to private process. Notification sent to private process.	Logging is done just before sending notification or request to the private process. This is the last entry in case of notification.
RESPONSE_FROM_PP	Received response from private process.	Logging is done after having received response from the private process.
RESPONSE_TO_TP	Response sent to trading partner.	Logging is done after having the response was successfully posted to the trading partner.
PACKAGE_MSG	SOAP Message packaged successfully.	Logging is done after the SOAP message is packaged. For example, after the header elements, body elements, and attachments have been added. If security is enabled, the SOAP body is signed, encrypted, and authenticated.
UNPACKAGE_MSG	Message from the <trading partner> parsed to a SOAP message successfully.	Logging is done after the received HTTP message (request/response) is converted to a SOAP message and decrypted (if encrypted previously).

Table 26 State Field Values for Responder Process (Cont'd)

Short Message	Description	Comments
REQUEST_DECRYPTED	Encrypted request message successfully decrypted.	Logging is done after the incoming message is decrypted successfully.
REQUEST_VERIFIED	Signed request from Trading Partner verified successfully.	Logging is done after the signed request message is verified.
DOCUMENT_DENIED	Received request document is denied because of encryption, signing, or authentication permissions.	Logging is done when the incoming request or response is denied because of security permissions for this operation.
VALIDATE_REQUEST	Incoming request (<header/Body>) is validated.	Logging is done after the validation of the incoming or outgoing request. The description specifies if the validation was a success or failure.
VALIDATE_RESPONSE	Incoming response (<header/Body>) is validated.	Logging is done after the validation of the incoming or outgoing response. Description specifies if validation was a success or failure.
COMPLETED_WITH_ERROR	The response cannot be sent to the partner.	This state is logged when responder fails to send a response to the initiator.

Non-Repudiation Logs

TIBCO BusinessConnect SOAP Protocol logs the signed incoming messages to the non-repudiation tables: the incoming notify or synchronous request messages are logged on the responder side, and the synchronous response messages are logged on the initiator side.

You can use non-repudiation logging in a business agreement and then apply it to all operations. It extracts the SOAP envelope from an incoming SOAP message and logs it when the message verification succeeds.

Non-repudiation logging stores the SOAP envelope without attachments, and also stores the verification certificates and decryption identities for encrypted messages.

The incoming SOAP messages are logged as follows:

- If messages are signed and encrypted, both the verification certificate and encryption identities are logged.
- If messages are signed, but not encrypted, only the verification certificate is logged.

Configuring a Non-Repudiation Log for SOAP

To configure a non-repudiation log for SOAP, perform the following steps:

1. Expand **BusinessConnect > Log Viewer** in the left panel.
2. In the Log Viewer window, select **SOAP**.
3. Click **Non-Repudiation** to display the Non-Repudiation log search options.
4. Select the options of the Non-Repudiation log by using information in [Table 27](#).

Table 27 Non-Repudiation Log: Search Filters

Column Name	Definition
Status	Select a specific status, such as ANY, COMPLETED, COMPLETED_WITH_ERROR, ERROR, or PENDING.
Connection	The database that you connect to. You can switch among multiple databases from this list.
Date Range	Select a specific date range, such as One Day, One Week, One Month, One Year, or --Custom--. If --Custom-- is selected, additional fields for defining exact dates are available.

5. To configure the advanced search filters, click **Add**.

6. Configure the advanced search settings by using information in [Table 28](#).

Table 28 Non-Repudiation Log: Advanced Search Filters

Column	Definition
Save as Query	Type the name under which you want to save the query you define in this field.
Operation ID	The operation ID. Boolean search: is, contains, is not, is not like.
Document ID	The document ID. Boolean search: is, contains, is not, is not like.
Trading Partner	The name of the trading partner. Boolean search: is, contains, is not, is not like.

Details for the transactions in the non-repudiation logs view cannot be opened for viewing.

Resend Logs

You can use the resend log to search for and view both resent transactions and resendable transactions. You can also easily initiate a resend.

- **Resendable Transactions** Resend a transaction.
- **Resend History** View messages that have been resent.

See Resend Logs in *TIBCO BusinessConnect Trading Partner Administration*.

Resendable Transactions

To configure a resendable transaction for SOAP, perform the following steps:

1. Expand **BusinessConnect > Log Viewer** in the left panel.
2. In the Log Viewer window, select **SOAP**.
3. Click **Resendable Transactions** to display the resendable transactions log search options.
4. Configure the search for resendable transactions.

[Table 29](#) lists the options to select in the search transactions log.

Table 29 Resend Log: Resendable Transactions

Column Name	Definition
Status	Select a specific status, such as ANY, COMPLETED, COMPLETED_WITH_ERROR, ERROR, or PENDING.
State	Select the transaction state: REQUEST_FROM_PP, REQUEST_TO_PP, and RESPONSE_TO_PP. See Table 30, State Values for Resendable Transactions , on page 130.
Connection	The database that you connect to. You can switch among multiple databases from this list.
Date Range	Select a specific date range, such as One Day, One Week, One Month, One Year, or --Custom--. If --Custom-- is selected, additional fields for defining exact dates are available.

Table 30 gives the explanation of transaction states for resendable transactions.

Table 30 *State Values for Resendable Transactions*

Short Message	Description	Comments
REQUEST_FROM_PP	Received request from private process.	The initiator message that was originally received is resent.
REQUEST_TO_PP	Request sent to private process.	The responder request messages are resent. This state cannot be resent for synchronous transactions.
RESPONSE_TO_PP	Response sent to private process.	The initiator response is resent.

5. To configure the advanced search filters, click **Add**.
6. Configure the advanced search settings by using information in [Table 31](#).

Table 31 *Resend Log: Advanced Search Filters*

Column Name	Definition
Trading Partner	The name of the trading partner. Boolean search: is, contains, is not, is not like.
Operation ID	The operation ID. Boolean search: is, contains, is not, is not like.
Transaction ID	The transaction ID. Boolean search: is, contains, is not, is not like.

Resend History

To configure a resend history log for SOAP, perform the following steps:

1. Expand **BusinessConnect > Log Viewer** in the left panel.
2. In the Log Viewer window, select **SOAP**.
3. Click **Resend History** to display the resend history log search options.
4. Configure the search for resend history.

Table 32 lists the options to select in the resend history log.

Table 32 Resend Log: Resend History

Column Name	Definition
Status	Select a specific status, such as ANY, COMPLETED, COMPLETED_WITH_ERROR, ERROR, or PENDING.
Connection	The database that you connect to. You can switch among multiple databases from this list.
Date Range	Select a specific date range, such as One Day, One Week, One Month, One Year, or --Custom--. If --Custom-- is selected, additional fields for defining exact dates are available.

5. To configure the advanced search filters, click **Add**.
6. Configure the advanced search settings by using the list presented in Table 33.

Table 33 Resend Log: Advanced Search Filters

Column	Definition
Trading Partner	The name of the trading partner. Boolean search: is, contains, is not, is not like.
Operation ID	The operation ID. Boolean search: is, contains, is not, is not like.
Transaction ID	The transaction ID. Boolean search: is, contains, is not, is not like.

Chapter 10 **Advanced Topics**

This chapter covers several advanced topics, such as trading with a third-party SOAP implementation, using the passthrough feature, and so on.

Topics

- [TIBCO-Specific Header: PartyInfo, page 134](#)
- [Trading with a Third-Party SOAP Implementation, page 135](#)
- [Using the Passthrough Feature, page 137](#)
- [TIBCO BusinessConnect SOAP Protocol Public Messages, page 140](#)
- [SOAP Message with a TIBCO BusinessConnect-Specific Header, page 142](#)
- [SOAP Envelope Attributes and Namespaces, page 144](#)

TIBCO-Specific Header: PartyInfo

TIBCO BusinessConnect specific header are the SOAP header elements defined by TIBCO for the interactions between business-to-business Gateway of two trading partners both implemented by TIBCO BusinessConnect SOAP Protocol. These header elements are the meta-data of the SOAP messages describing attributes of these messages: from which this message was sent, to whom this message is targeted, and the operation ID of the message, so the receiver knows how to process the message, and do other operations. The receiver of the SOAP message with this TIBCO BusinessConnect-specific header, which is also implemented by TIBCO BusinessConnect SOAP Protocol, understands these header elements.

The TIBCO-specific header data is defined by the schema, and is available in the `partyinfo.xsd` file in the `TIBCO_HOME/bc/version_number/protocols/soap/samples/schemas` directory.

This schema describes information such as:

- From (partner)
- To (partner)
- Transaction ID
- Operation ID
- Operation type

The `from` and `to` partner identification in the outbound messages, which are in the `partyInfo` header element, are from the Domain and ID configurations of the partner and host. Values received in the request messages from private process always take precedent.

This results in the following outcomes:

- If the Domain and ID values are not available from the configuration or the private process message, the `name` sub-element of the `from` or `to` element is populated with the host or partner name.
- If the Domain and ID values are available from either configuration or the private process message, the Domain and ID values are used for the `from` or `to` elements.
- If the host and partner configuration and private process messages both provide the same type of identifications (both are partner names, or both are Domain and ID values), the values from private process messages take precedence.

Trading with a Third-Party SOAP Implementation

This section describes how a SOAP message from a third-party SOAP implementation must look like to be processed by a TIBCO BusinessConnect SOAP Protocol implementation. The BusinessConnect-specific header elements, while useful for the messages exchange between two TIBCO BusinessConnect SOAP Protocol implementations, might not be adopted by other third-party SOAP implementations. So rather than using TIBCO BusinessConnect-specific headers, some other approaches must be used to make seamless interactions with trading partners.

Receiving SOAP Messages without TIBCO BusinessConnect-Specific Headers

As an alternative to including the TIBCO BusinessConnect-specific header in the SOAP header, information can be encoded in the URL. In such cases, when receiving an incoming SOAP message without any TIBCO BusinessConnect-specific header data, the incoming URL for the following is checked:

- Trading partner
- Host name
- Operation ID
- Transaction ID



Information has to be provided completely either in the `partyInfo` header or in the URL: it cannot be specified partially in the header and partially in the URL. When information is present both in the `partyInfo` header and in the URL, information in the `partyInfo` header takes precedence.

The URL format is as follows:

```
http://host:port/dmz/SOAP
?host=host_name&tpname=trading_partner_name
&opid=operation_ID
&transid=transaction_ID
```

Example:

```
http://www.SOAPServer.com:6700/dmz/SOAP
?host=SOAPServer&tpname=SOAPClient
&opid=Sync/1.0/PORquest
&transid=1232456789
```

None of the URL parameters is a required value.

Sending SOAP Messages without TIBCO BusinessConnect-Specific Headers

To enable TIBCO BusinessConnect SOAP Protocol to send a SOAP message without TIBCO BusinessConnect-specific headers, do not select the **Add BusinessConnect Specific Header** check box in the trading partner setup procedures. See [Configuring General Properties for a Partner on page 100](#).

Without the TIBCO BusinessConnect-specific header in the outbound SOAP messages, the header information is put and encoded in the URL of the outbound HTTP requests. The format of the URL is the same.

Using the Passthrough Feature

In cases that some of the header values are missing from the inbound URL, TIBCO BusinessConnect SOAP Protocol uses the following rules to process the inbound message:

- The `tpName` and `opid` parameter values are the minimum information in the URL for inbound message to be processed end-to-end.

The `tpName` parameter value indicates which trading partner the message is sent from, and the `opid` parameter value indicates which operation is used to process the message. In a synchronous request-response operation type, the response message can be sent back if everything else is correct.

- If either `tpName` or `opid` parameter value is missing, the following two situations occur:
 - If the **Allow Anonymous SOAP Messages** check box is selected, the message is processed in passthrough mode and forwarded directly to private process without validation and parsing. For a synchronous request reply operation, the message is processed as a notify operation without any response being sent back.

The **Allow Anonymous SOAP Messages** check box can be configured in various locations. If the `host` parameter value is present in the URL, the **Allow Anonymous SOAP Messages** check box in the configuration of the local host referred by this `host` parameter value takes effect. If the `tpName` parameter value is present in the URL, the **Allow Anonymous SOAP Messages From This Trading Partner** check box in the configuration of the trading partner referred by this `tpName` parameter value takes effect. If neither `host` or `tpName` value is present, the configuration of the default host takes effect.

- If the **Allow Anonymous SOAP Messages** check box is unselected in either the related host or trading partner configuration, the message is rejected.

Here are the detailed descriptions of some typical scenarios:

Case 1

SOAP header is absent; URL has no parameters.

Example: `http://www.SOAPServer.com:6700/dmz/SOAP`

The SOAP message is rejected by default, and a SOAP fault is generated and sent back to the trading partner. To avoid messages from trading partners being rejected, you must select the **Allow Anonymous SOAP Messages** check box under **BusinessConnect > Participants > Default Host > Protocols > SOAP > General**.

When anonymity is enabled, TIBCO BusinessConnect SOAP Protocol has no context for the inbound message. It can only handle the incoming message as a notify operation. TIBCO BusinessConnect server forwards the inbound SOAP message to the private process through the `ResponderRequest` message. The `request` and `requestHeader` fields of this message contain the incoming SOAP header and body information in XML format. Once the inbound SOAP message is forwarded to the local private process, BusinessConnect server sends an HTTP 204/No Content response back to the trading partner.

Case 2

SOAP header is absent; URL has no `tpName` and `host` parameters.

Example:

```
http://www.SOAPServer.com:6700/dmz/SOAP
?opid=Sync/1.0/POResult
&transid=1232456789
```

TIBCO BusinessConnect SOAP Protocol cannot determine the trading partner context information. The behavior is identical to when no parameters are present.

Case 3

SOAP header is absent; URL has no `opid` parameter, but has the `tpName` parameter.

Example:

```
http://www.SOAPServer.com:6700/dmz/SOAP
?tpname=SOAPClient
&transid=1232456789
```

The SOAP message is rejected by default, and a SOAP fault is generated and sent back to the trading partner. To avoid messages from trading partners being rejected, you must select the **Allow Anonymous SOAP Messages From This Trading Partner** check box under **BusinessConnect > Participants > SOAPClient > Protocols > SOAP > General**.

When anonymity is enabled for this trading partner, TIBCO BusinessConnect SOAP Protocol has no operation-related context for the inbound message. It can only handle the incoming message as a notify operation. TIBCO BusinessConnect SOAP Protocol forwards the inbound SOAP message to the private process through the `ResponderRequest` message. The `request` and `requestHeader` fields of this message contain the incoming SOAP header and body information in XML format. Once the incoming SOAP message is forwarded to the local private process, BusinessConnect server sends an HTTP 204/No Content response back to the trading partner. For this transaction, `host` is the default host of the system.

Case 4

SOAP header is absent; URL has no `transid` parameter, but has the `tpName` and `opid` parameters.

Example:

```
http://www.SOAPServer.com:6700/dmz/SOAP
?tpname=SOAPClient &opid=Sync/1.0/PORquest
```

TIBCO BusinessConnect SOAP Protocol has the complete context of the inbound SOAP message. The transaction is processed as a regular synchronous request response or notify transaction, depending on the type of operation in the incoming message. A unique transaction ID is generated and this ID is passed to the private process.

Case 5

SOAP header is absent; URL has no `opid` parameter, but has the `host` and `tpName` parameters.

Example:

```
http://www.SOAPServer.com:6700/dmz
/SOAP?host=SOAPClient&tpname=SOAPServer
```

The SOAP message is rejected by default and a SOAP fault is generated and sent back to the trading partner, because no operation context is used for processing the inbound message. To avoid messages from trading partners being rejected, you must select the **Allow Anonymous SOAP Messages From This Trading Partner** check box under **BusinessConnect > Participants > SOAPClient > Protocols > SOAP > General**. With the `host` and `tpName` parameters, the configuration of the trading partner that is *SOAPClient* referred by the `tpName` takes precedence. BusinessConnect server forwards the inbound SOAP message to the private process through the `ResponderRequest` message. The `request` and the `requestHeader` fields of this message contain the incoming SOAP header and body information in XML format. Once the incoming SOAP message is forwarded to the local private process, BusinessConnect server sends an HTTP 204/No Content response back to the trading partner.



Passthrough is only available for HTTP and HTTPS transports. The HTTPSCA transport can be used only when the `tpname` parameter is populated in the URL for passthrough messages.

TIBCO BusinessConnect SOAP Protocol Public Messages

The following is an example of a public SOAP request message that TIBCO BusinessConnect SOAP Protocol generates if the **Add BusinessConnect Specific Header** check box is selected in the configuration of the sending trading partner.

See [Configuring General Properties for a Partner on page 100](#) to set up this option for a trading partner.

Descriptions of the messages are available in the following sections:

- [SOAP Header on page 140](#)
- [SOAP Body on page 140](#)

SOAP Header

Header Section 1 - Customized Header Elements

From `<ep:endpoints xmlns:ep="http://user.org/header"> to </prop:properties>`

This is the part of the SOAP header that contains customized header elements and comes from the `requestHeader` field in the `InitiatorRequest` message from the private process.

See a sample SOAP envelope message in [SOAP Message with a TIBCO BusinessConnect-Specific Header on page 142](#).

See [Initiator Outbound Request — Private Process to BusinessConnect on page 149](#) for a description of this private process message.

Header Section 2 - TIBCO-Specific Header Elements

From `<pi:PartyInfo xmlns:pi="http://www.tibco.com/namespaces/bc/2002/04/partyinfo.xsd"> to </pi:PartyInfo>`

This is the BusinessConnect specific part of the SOAP header that is added if the **Add BusinessConnect Specific Header** check box is selected under **BusinessConnect > Participants > Partner > Protocols > SOAP > General**.

SOAP Body

From <cpo:CommonPO xmlns:cpo="http://po.org/body"> to </cpo:CommonPO>

This is the part of the SOAP body that comes from the `request` field in the `InitiatorRequest` message from the private process, which is actually the business document to be exchanged to the responder side.

See [Initiator Outbound Request — Private Process to BusinessConnect](#) on page 149 for a description of this private process message.

SOAP Message with a TIBCO BusinessConnect-Specific Header

This is a sample SOAP 1.1 XML message that has the TIBCO BusinessConnect specific header:

```
<?xml version='1.0' encoding='utf-8'?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <SOAP-ENV:Header>
    <ep:endpoints xmlns:ep="http://user.org/header">
      <ep:to>
        <ep:address>
          <name>Book Orders</name>
          <street>1st Street</street>
          <city>New York</city>
          <zip>1111</zip>
        </ep:address>
      </ep:to>
      <ep:from>
        <ep:address>
          <name>Book Lovers</name>
          <street>1st Street</street>
          <city>Los Angeles</city>
          <zip>90210</zip>
        </ep:address>
      </ep:from>
    </ep:endpoints>
    <prop:properties xmlns:prop="http://user.org/header">
      <identity>uuid:74b9f5d0-33fb-4a81-b02b-5b760641c1d6</identity>
      <sentAt>2000-05-14T03:00:00+08:00</sentAt>
      <expiresAt>2000-05-15T04:00:00+08:00</expiresAt>
      <topic>http://electrocommerce.org/purchase_order</topic>
    </prop:properties>
    <pi:PartyInfo xmlns:pi="http://www.tibco.com/namespaces/bc/2002/04/partyinfo.xsd">
      <from>
        <name>SOAPClient</name>
      </from>
      <to>
        <name>SOAPServer</name>
      </to>
      <operationID>POInterface/1.0/PONotify</operationID>
      <operationType>notify</operationType>
      <transactionID>ZrGSGRWSbWkd0UePKiQefcJubQo</transactionID>
    </pi:PartyInfo>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <cpo:CommonPO xmlns:cpo="http://po.org/body">
      <cpo:POHeader xmlns:cpo="http://po.org/body" CreationDate="2000-06-23" Number="12345" Purpose="PO" Type="EZ" />
      <cpo:BillTo />
      <cpo:ShipTo ContactName="BonifazLuis" ContactNumber="12345" ContactType="CT" />
      <cpo:Item>
        <cpo:ItemHeader xmlns:cpo="http://po.org/body" ExtendedPrice="499.75" Price="99.95" Quantity="5" UnitOfMeasure="EA" />
        <cpo:ItemDescription Description="Word Processing Application" Type="F" />
      </cpo:Item>
    </cpo:CommonPO>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

```
</cpo:Item>  
<cpo:Total LineItemTotal="87" POTotal="544.15" QuantityTotal="34">12.34</cpo:Total>  
<cpo:Description>VGhpcyBpcyBhIHB1cmNoYXNlIG9yZGVyIGRlc2NyaXB0aW9uLg==</cpo:Description>  
</cpo:CommonPO>  
</SOAP-ENV:Body>  
</SOAP-ENV:Envelope>
```

SOAP Envelope Attributes and Namespaces

This section shows how a private process can set SOAP envelope attributes and namespaces. The SOAP message generated by TIBCO BusinessConnect SOAP Protocol has the private process-supplied envelope attributes and namespaces along with the default attributes and namespaces generated by TIBCO BusinessConnect SOAP Protocol.

The SOAP envelope attributes are managed following these rules:

- For outbound messages, all attributes are populated on the SOAP envelope.
- For inbound messages, only the envelope attributes with their associated namespaces are sent to the private process. The unused namespace attributes are not sent.
- The envelope attributes must be bound to a namespace prefix.
- A SOAP envelope namespace, such as <http://www.w3.org/2003/05/soap-envelope>, cannot be sent from the private process. The private process cannot send the namespace for a SOAP envelope prefix that has been defined in TIBCO Administrator for a given trading partner.

Envelope Attributes and Namespaces Example

The `ae/SOAP/InitiatorRequest` and `ae/SOAP/ResponderResponse` SOAP AE class have the **envelopeAttributes** field. This field is of type sequence, `ae/SOAP/Attributes`. `ae/SOAP/Attributes` is a sequence of the class `ae/SOAP/Attribute`.

The `ae/SOAP/Attribute` class contains the following fields:

- **name**
- **value**

The following example is a SOAP 1.2 message:

```

<SOAP-ENV:Envelope
xmlns:SOAP-ENV="http://www.w3.org/2003/05/soap-envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:pro="http://myorg.com/schemas/promotions"
pro:promotionExpiryDate="03/12/2003">
...
</SOAP-ENV:Envelope>

```

This SOAP message has a namespace and a `promotionExpiryDate` attribute in the namespace `http://myorg.com/schemas/promotions`.

Create an `ae/SOAP/Attributes` sequence element. This sequence node has two `ae/SOAP/Attribute` elements.

The field values for the first `ae/SOAP/Attribute`:

- **name:** `xmlns:pro`
- **value:** `"http://myorg.com/schemas/promotions"`

The field values for the second `ae/SOAP/Attribute`:

- **name:** `pro:promotionExpiryDate`
- **value:** `03/12/2003`

Set the `ae/SOAP/Attributes` sequence element to the **envelopeAttributes** field in the `ae/SOAP/InitiatorRequest` OR `ae/SOAP/ResponderResponse` node.

If an attribute is not bound to a namespace, an error occurs and the transaction is cancelled.

Configurable Namespace Prefix for the Envelope Namespace

The prefix for the envelope namespace is now configurable.

For example, a SOAP message can have the following format:

```

<abc:Envelope
xmlns:abc="http://www.w3.org/2003/05/soap-envelope/">
<abc:Header>
.....
</abc:Header>
<abc:Body>
.....
</abc:Body>
</abc:Envelope>

```

where `abc` replaces the previously pre-defined string `SOAP-ENV`.

The string `SOAP-ENV` is still the default, but it can be replaced by any string that you want. This prefix is not used for SOAP faults, which are always sent with the `SOAP-ENV` prefix.

Chapter 11 **Private Messages**

This chapter describes private message formats in TIBCO BusinessConnect SOAP Protocol transactions and elements included in private messages.

Topics

- [Overview](#), page 148
- [Initiator Messages](#), page 149
- [Responder Messages](#), page 157
- [Advisory Message](#), page 166
- [Attachment](#), page 168
- [Attachment Type](#), page 172
- [TradingPartner Type](#), page 173
- [Attributes](#), page 174
- [SOAPFault](#), page 175
- [WS-Addressing](#), page 176
- [XOP](#), page 178
- [AuthenticationToken](#), page 179

Overview

The following sections describe the messages used for private request and response document exchange in TIBCO BusinessConnect SOAP Protocol.

- For RV Messages, message fields are packaged in the `<data>` tag as part of the `aeRvMsg` format.
- For JMS messages, these values are part of the node message.

See `aeRvMsg` Message Format and JMS Message Format in *TIBCO BusinessConnect Concepts*.

Private process messages are described in this chapter:

- [Initiator Messages on page 149](#)
- [Responder Messages on page 157](#)
- [Advisory Message on page 166](#)

The following nodes can be included in private messages:

- [Attachment Type on page 172](#)
- [TradingPartner Type on page 173](#)
- [Attributes on page 174](#)
- [SOAPFault on page 175](#)
- [WS-Addressing on page 176](#)
- [XOP on page 178](#)
- [AuthenticationToken on page 179](#)

Initiator Messages

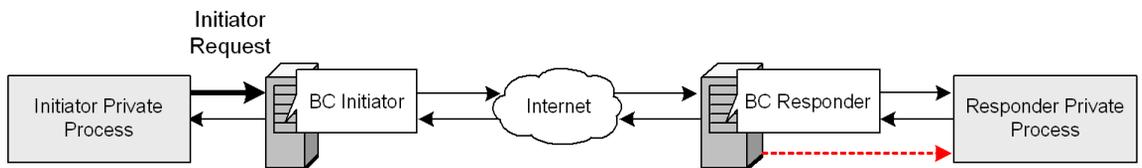
The following initiator messages are described:

- [Initiator Outbound Request — Private Process to BusinessConnect on page 149](#)
- [Initiator Inbound Response — BusinessConnect to Private Process on page 154](#)

Initiator Outbound Request — Private Process to BusinessConnect

The initiator private process uses this message to handle outbound requests, as shown in [Figure 24](#).

Figure 24 Initiator Outbound Request



Subject Name `prefix.installation.SOAP.INITIATOR.REQUEST`

Example RV Subject: `AX.BC.BC-ACME.SOAP.INITIATOR.REQUEST`

Example JMS queue: `AX.BC.BC-ACME.INITIATOR.REQUEST`

Table 34 Initiator Request (Sheet 1 of 5)

Field	Type	Required	Description
standardID	String	Yes	The name of the protocol: SOAP.
transactionID	String	No	A unique ID within the initiator private process. This value is sent across the public boundary to the trading partner in the BusinessConnect specific SOAP header. See TIBCO BusinessConnect SOAP Protocol Public Messages on page 140 .
operationID	String	Yes	A three-part ID of the form: <code>interface/version/operation_name</code> . See Adding an Operation to a Version on page 79 .

Table 34 Initiator Request (Sheet 2 of 5)

Field	Type	Required	Description
closure	String	No	<p>The private process generates a closure message and sends it to BusinessConnect. BusinessConnect is required to return this closure contents back in the InitiatorResponse message to ensure that the private process can match it with the original InitiatorRequest message.</p> <p>This field is not available for JMS messages. In JMS header fields, closure is populated as JMSCorrelationID.</p>
hostName	String	No	<p>Identifies the local host in a multi-host environment. If the value is not present, the default host is used.</p> <p>This field can either have the actual name of the host, or the domain ID of the host.</p>
hostDomain	String	No	<p>Identifies the local host domain in a multi-host environment.</p> <p>If this field is present, the hostName field must contain the domain identity.</p>
tpName	String	Yes	<p>Name of the trading partner.</p> <p>This field can either have the actual name of the partner, or the domain ID of the partner.</p>
tpDomain	String	No	<p>Trading partner domain.</p> <p>If this field is not present, the default domain TPname is used.</p>

Table 34 Initiator Request (Sheet 3 of 5)

Field	Type	Required	Description
xop	N/A	No	<p>Identifies the specific value under xop > tagName > value if MTOM is selected when sending outbound messages.</p> <p>Note:</p> <ul style="list-style-type: none"> • If the SOAP MTOM Enabled check box is selected, the value under xop > tagName > value of the private process message must be populated according to the binary data element in the <code>body</code> node. If not, the request is rejected and the outbound process is ended. • If the SOAP MTOM Enabled check box is not selected, and the value under xop > tagName > value in the private process is provided, the request is rejected and the outbound process is ended. • MTOM and attachments from the private process cannot be used together. If the SOAP MTOM Enabled check box is selected, and some attachments are also specified from the private process request message, the request is rejected and the outbound process is ended. • When selecting the SOAP MTOM Enabled and Require Content Encryption check boxes to send outbound messages, the binary content which refers to the value under xop > tagName > value in the private process message is ignored. The encrypted data in the <code>CipherValue</code> element after encryption is used to construct the MTOM attachment. <p>See Configuring General Properties for a Partner on page 100 on how to select SOAP MTOM.</p>

Table 34 Initiator Request (Sheet 4 of 5)

Field	Type	Required	Description
saml	N/A	No	<p>Specifies the required values in the <code>saml</code> node if SAML assertions are used when sending outbound messages.</p> <p>If you use SAML assertions in conformance with NHIN/esMD Authorization Framework, provide all the required values in the <code>saml</code> node from the private process. Errors occur if any of the required values is missing.</p> <p>See Request Action Tab on page 84 on how to configure SAML assertions in conformance with NHIN/esMD Authorization Framework.</p>
authentication Token	N/A	No	<p>Includes the user name and password information that is used for WSS UsernameToken authentication, which is an alternative to the WSS signature authentication.</p> <p>See UsernameToken for Authentication on page 17.</p>
attachment	Sequence of Attachment	No	<p>A sequence of attachments.</p> <p>See Attachment on page 168.</p>
request	String	No	<p>The body of the public SOAP message in XML format.</p>
requestHeader	String	No	<p>The header of the public SOAP message in XML format.</p>
WS-Addressing	N/A	No	<p>Specifies the WS-Addressing parameters.</p> <p>See WS-Addressing on page 176.</p>
httpAttributes	Sequence of SOAP/ Attribute	No	<p>Contains the HTTP header attributes, which are populated in the HTTP header of the outbound HTTP request.</p> <p>These optional attribute values might be useful for the receiving server.</p> <p>See Attributes on page 174.</p>

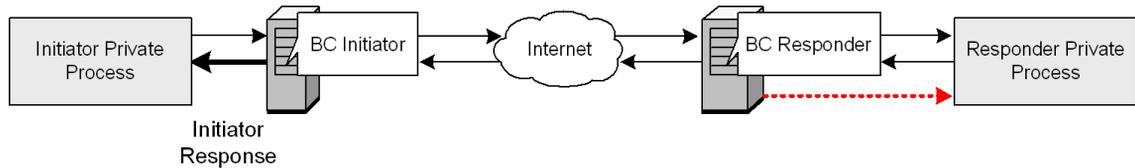
Table 34 Initiator Request (Sheet 5 of 5)

Field	Type	Required	Description
envelope Attributes	Sequence of SOAP/ Attribute	No	Contains the SOAP envelope attributes. These optional attribute values might be useful for the receiving server. See Attributes on page 174 and SOAP Envelope Attributes and Namespaces on page 144 .
enableMTOM	Boolean	No	If you provide value for this field, this value will override the value given in Trading Partner's settings for SOAP MTOM Enabled filed.
requestFile	String	No	Specify the reference file for request.

Initiator Inbound Response — BusinessConnect to Private Process

A synchronous response is sent from the responder BusinessConnect to the initiator BusinessConnect and forwarded to the local private process, as shown in [Figure 25](#).

Figure 25 Initiator Inbound Response



Subject Name `prefix.installation.SOAP.INITIATOR.RESPONSE`

Example RV Subject: `AX.BC.BC-ACME.SOAP.INITIATOR.RESPONSE`

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

Table 35 Initiator Response (Sheet 1 of 3)

Field	Type	Required	Description
response	String	Yes	The body of the SOAP response from the responder in XML format. The body element is modified to have appropriate namespaces; for example, if a body element is in a default namespace, it is added to the body element before converting to XML format.
statusCode	Integer	Yes	See statusCode and statusMsg Field Reference on page 182.
statusMsg	String	Yes	OK, or the string representing the cause of the error. See statusCode and statusMsg Field Reference on page 182.
operationID	String	Yes	A three-part ID of the form: <i>interface/version/operation_name</i> See Adding an Operation to a Version on page 79.

Table 35 Initiator Response (Sheet 2 of 3)

Field	Type	Required	Description
transactionID	String	No	The same value that was in the transactionID field of the initiator request.
closure	String	Yes	<p>The private process generates a closure message and sends it to BusinessConnect. BusinessConnect is required to return this closure contents back in the InitiatorResponse message to ensure that the private process can match it with the original InitiatorRequest message.</p> <p>This field is not available for JMS messages. In JMS header fields, closure is populated as JMSCorrelationID.</p>
attachment	Sequence of Attachment	No	<p>Sequence of attachments.</p> <p>The class definition name for this message is ac/BC/Attachment. See Attachment on page 168.</p>
responseHeader	String	No	<p>The header of the response SOAP message in XML format.</p> <p>The header element is modified to have appropriate namespaces; for example, if a header element is in a default namespace, it is added to the header element before converting it to XML format.</p>
saml	N/A	No	Contains the SAML parameters.
WS-Addressing	N/A	No	<p>Contains the WS-Addressing parameters.</p> <p>See WS-Addressing on page 176.</p>
envelopeAttributes	Sequence of SOAP/Attribute	No	<p>Contains the SOAP envelope attributes. These optional attribute values might be useful for the receiving server.</p> <p>See Attributes on page 174 and SOAP Envelope Attributes and Namespaces on page 144.</p>
resent	Boolean	No	This field is set to true when the message is a resent transaction.
standardID	String	Yes	The name of the protocol: SOAP.

Table 35 Initiator Response (Sheet 3 of 3)

Field	Type	Required	Description
duplicate	Boolean	No	This field is set to <code>true</code> when this is a duplicate response.
responsefile	String	No	Specify the reference file for response.



The Receive Response activity can receive a SOAP fault or any valid XML document as a response. Therefore, if the **Parse XML Payload** check box in the **Configuration** tab in private process is selected, the `body` element can have either the schema for the response, or the `SOAPFault` element.

To get an appropriate response, the root element has to be configured in TIBCO Administrator, and also the schema has to be present for that operation. If no schema is defined for the response, only the `SOAPFault` element is displayed. This behavior is the same as in the case of the Send Response activity when sending a response.

See [SOAPFault](#) on page 175.

Responder Messages

The following responder messages are described:

- [Responder Inbound Request](#) — [BusinessConnect to Private Process](#) on page 157
- [Responder Outbound Response](#) — [Private Process to BusinessConnect](#) on page 161
- [Responder Acknowledgement](#) — [BusinessConnect to Private Process](#) on page 165

Responder Inbound Request — BusinessConnect to Private Process

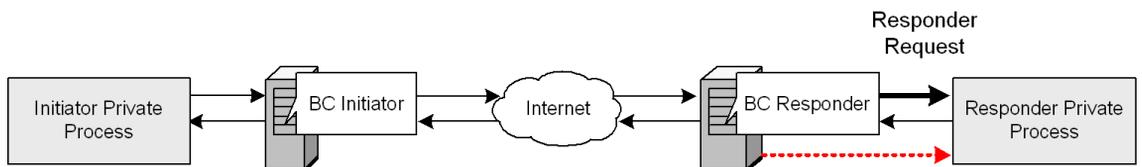
The responder private process uses this message to handle inbound requests, as shown in [Responder Inbound Request](#).



Even if the incoming message size exceeds the threshold set under **BusinessConnect > Gateway > Gateway Services > HTTP > Transport > Advanced > Data Streaming Threshold (KB)** in TIBCO Administrator, the full message, not a file reference, is sent to the private process through RV or JMS.

A new property `bc.soap.pp.fileref.threshold` is added under **BusinessConnect > System Settings > Activated Protocol Plug-ins and properties > SOAP** in TIBCO Administrator. The default value is 5MB.

Figure 26 Responder Inbound Request



Subject Name `prefix.installation.SOAP.RESPONDER.REQUEST`

Example RV Subject: `AX.BC.BC-ACME.SOAP.RESPONDER.REQUEST`

Example JMS queue: `AX.BC.BC-ACME.RESPONDER.REQUEST`

Table 36 Responder Inbound Request (Sheet 1 of 3)

Field	Type	Required	Description
request	String	Yes	<p>The body of the public SOAP message in XML format.</p> <p>The <code>body</code> element is modified to have appropriate namespaces; for example, if a <code>body</code> element is in a default namespace, it is added to the <code>body</code> element before converting to XML format.</p>
sourceTP	String	Yes	<p>The name of the initiator of this message.</p> <p>This field contains name or domain identity value.</p>
sourceTPDomain	String	No	The domain of the initiator of this message.
destinationTP	String	Yes	<p>The name of the responder to this message.</p> <p>This field contains name or domain identity value.</p>
destinationTPDomain	String	No	The domain of the responder to this message.
operationID	String	Yes	<p>A three-part ID of the form: <i>interface/version/operation_Name</i>.</p> <p>See Adding an Operation to a Version on page 79.</p>
transactionID	String	No	<p>The value of the transaction ID from TIBCO BusinessConnect SOAP Protocol specified HTTP URL. If no transaction ID is specified, a unique one with the name <code>bcsoap:guid</code> is generated.</p> <p>See TIBCO BusinessConnect SOAP Protocol Public Messages on page 140.</p>

Table 36 Responder Inbound Request (Sheet 2 of 3)

Field	Type	Required	Description
closure	String	No	<p>Used for synchronous request-response operation. BusinessConnect generates a closure message and sends it to the local private process.</p> <p>If this is a request of a synchronous request-response operation, the private process is required to return this closure contents back in the ResponderResponse message to ensure that BusinessConnect can match it with the original ResponderRequest message.</p> <p>This field is not available for JMS messages. In JMS header fields, closure is populated as JMSCorrelationID.</p>
standardID	String	Yes	The name of the protocol: SOAP.
attachment	Sequence of Attachment	No	<p>A sequence of attachments.</p> <p>See Attachment on page 168.</p>
requestHeader	String	No	<p>The header of the public SOAP message in XML format.</p> <p>The header element is modified to have appropriate namespaces; for example, if a header element is in a default namespace, it is added to the header element before converting to XML format.</p>
saml	N/A	No	Contains the SAML elements received from inbound requests.
WS-Addressing	N/A	No	<p>Contains the WS-Addressing parameters.</p> <p>See WS-Addressing on page 176.</p>
envelopeAttributes	Sequence of SOAP/Attribute	No	<p>Contains the SOAP envelope attributes. These optional attribute values might be useful for the receiving server.</p> <p>See Attributes on page 174 and SOAP Envelope Attributes and Namespaces on page 144.</p>

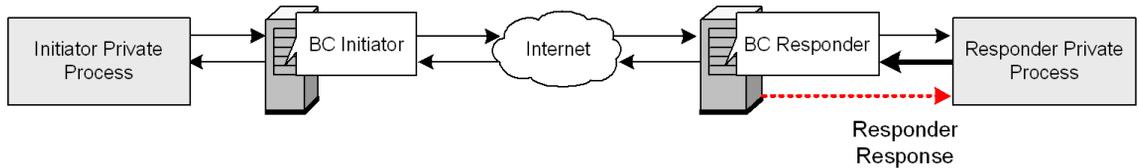
Table 36 Responder Inbound Request (Sheet 3 of 3)

Field	Type	Required	Description
resend	Boolean	No	This field is set to <code>true</code> when the message is a resent transaction.
duplicate	Boolean	No	This field is set to <code>true</code> when this is a duplicate request.
operationType	String	Yes	Type of operation: notify or synchronous request-response.
requestFile	String	No	Specify the reference file for request if request data is too big to embed in the message directly.

Responder Outbound Response — Private Process to BusinessConnect

The responder private process uses this message to handle outbound responses, as shown in Figure 27.

Figure 27 Responder Outbound Response



Subject Name *prefix.installation.SOAP.RESPONDER.RESPONSE*

Example RV Subject: AX.BC.BC-ACME.SOAP.RESPONDER.RESPONSE

Example JMS queue: AX.BC.BC-ACME.RESPONDER.RESPONSE

Table 37 Responder Outbound Response (Sheet 1 of 3)

Field	Type	Required	Description
statusCode	Integer	No	All status codes below 200 and above 299 are considered to be SOAP fault. When the value in the <code>statusCode</code> field is set to under 200 or above 299, the <code>SOAPFault</code> element is sent to the trading partner and the body is ignored even if it contains data. To send the content of the body, the value in the <code>statusCode</code> field must range between 200 and 299. See statusCode and statusMsg Field Reference on page 182.
statusMsg	String	No	OK, or the string representing the cause of the error. See statusCode and statusMsg Field Reference on page 182.
response	String	Yes	The response business document in XML format.

Table 37 Responder Outbound Response (Sheet 2 of 3)

Field	Type	Required	Description
xop	N/A	No	<p>Identifies the specific value under xop > tagName > value if MTOM is selected when sending outbound messages.</p> <p>Note:</p> <ul style="list-style-type: none"> If the SOAP MTOM Enabled check box is selected, the value under xop > tagName > value of the private process message must be populated according to the binary data element in the <code>body</code> node. If not, the request is rejected and the outbound process is ended. If the SOAP MTOM Enabled check box is not selected, and the value under xop > tagName > value in the private process is provided, the request is rejected and the outbound process is ended. MTOM and attachments from the private process cannot be used together. If the SOAP MTOM Enabled check box is selected, and some attachments are also specified from the private process request message, the request is rejected and the outbound process is ended. When selecting the SOAP MTOM Enabled and Require Content Encryption check boxes to send outbound messages, the binary content which refers to the value under xop > tagName > value in the private process message is ignored. The encrypted data in the <code>CipherValue</code> element after encryption is used to construct the MTOM attachment. <p>See Configuring General Properties for a Partner on page 100 on how to select SOAP MTOM.</p>
authentication Token	N/A	No	<p>Includes the user name and password information that is used for WSS UsernameToken authentication, which is an alternative to the WSS signature authentication.</p> <p>See UsernameToken for Authentication on page 17.</p>

Table 37 Responder Outbound Response (Sheet 3 of 3)

Field	Type	Required	Description
attachment	Sequence of Attachment	No	A sequence of attachments. See Attachment on page 168 .
closure	String	Yes	BusinessConnect generates a closure message, and sends it to the local private process in the ResponderRequest message upon receiving a request for a synchronous request-response operation. The private process must return this closure contents in the ResponderResponse message to ensure that BusinessConnect can match it with the original ResponderRequest message. This field is not available for JMS messages. In JMS header fields, closure is populated as JMSCorrelationID.
operationID	String	Yes	A three-part ID of the form: <i>interface/version/operation_Name</i> .
responseHeader	String	No	The header of the response. This becomes the header of the public SOAP message.
WS-Addressing	N/A	No	Specifies the WS-Addressing parameters. See WS-Addressing on page 176 .
envelope Attributes	Sequence of SOAP/Attribute	No	Contains the SOAP envelope attributes. These optional attribute values might be useful for the receiving server. See Attributes on page 174 and SOAP Envelope Attributes and Namespaces on page 144 .
saml	N/A	No	Not used for Responder.Response messages.
soapFault	N/A	No	See SOAPFault on page 175 .
standardID	String	Yes	The name of the protocol: SOAP.
responseFile	String	No	Specify the reference file for response.



The Send Response activity has the SOAPFault element as a part of a body element, and also of the BCResponderResponse element.

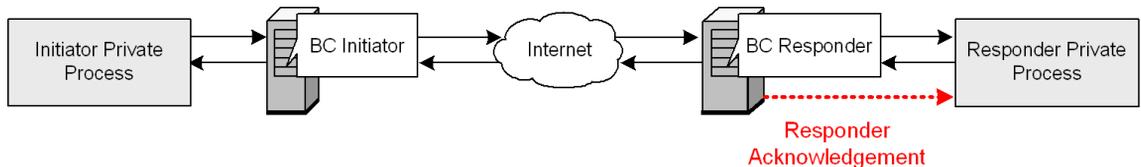
When sending a SOAP fault, you can populate it in the SOAPFault element under the BCResponderResponse element or in the SOAPFault element under the body element. Any SOAP fault values specified under the body element are ignored when the values under the BCResponderResponse element are provided.

See [SOAPFault](#) on page 175.

Responder Acknowledgement — BusinessConnect to Private Process

When the responder BusinessConnect sends a response, the responder BusinessConnect sends an acknowledgement to the local private process if the initiator BusinessConnect returns no error, as shown in Figure 28.

Figure 28 Responder Acknowledgement



Subject Name *prefix.installation.SOAP.RESPONDER.ACK*

Example RV Subject: AX.BC.BC-ACME.SOAP.RESPONDER.ACK

Example JMS queue: AX.BC.BC-ACME.RESPONDER.ACK

Table 38 ResponderAck

Field	Type	Required	Description
statusCode	Integer	Yes	See statusCode and statusMsg Field Reference on page 182.
statusMsg	String	Yes	OK, or the string representing the cause of the error. See statusCode and statusMsg Field Reference on page 182.
operationType	String	Yes	ACK.
closure	String	Yes	BusinessConnect generates a closure message, and sends it to the local private process in the <code>ResponderRequest</code> message upon receiving a request for a synchronous request-response operation. This field has the same value as in the <code>ResponderRequest</code> message. This field is not available for JMS messages. In JMS header fields, closure is populated as <code>JMSCorrelationID</code> .

Advisory Message

This message is used to publish status information.

Subject Name *prefix.installation.SOAP.ERROR*

Example RV Subject: AX.BC.BC-ACME.SOAP.ERROR

Example JMS topic: AX.BC.BC-ACME.ERROR

Table 39 Advisory Messages

Field	Type	Required	Description
statusCode	Integer	Yes	One of the error codes. See statusCode and statusMsg Field Reference on page 182.
statusMsg	String	Yes	Represents the cause of the error. In this case, the response might be an error document. See statusCode and statusMsg Field Reference on page 182.
details	String	No	More detailed description of the error. This can be an analysis of the problem.
operationID	String	Yes	A three-part ID of the form: <i>interface/version/operation_Name</i> .
transactionID	String	No	Request-response ID for which the error message is generated.
closure	String	No	If the error occurred on the initiating side, it contains a closure sent as part of the request. Otherwise, it contains a closure passed to the responder private process. It might be null. This field is not available for JMS messages. In JMS header fields, <i>closure</i> is populated as <i>JMSCorrelationID</i> .
host	soap/ Trading Partner	No	Host information. See TradingPartner Type on page 173.
tpName	String	Yes	Name of the trading partner.

Table 39 Advisory Messages (Cont'd)

Field	Type	Required	Description
tradingPartner	soap/ Trading Partner	No	Trading partner information. This field is populated only if TIBCO BusinessConnect SOAP Protocol is able to extract the domain and ID information for the trading partner. See TradingPartner Type on page 173.
Code	N/A	No	Specifies the value and subcode to check detailed fault information.
Reason	N/A	No	Specifies the cause of the fault.
Node	String	No	Identifies the node in which the fault occurred.
Role	String	No	Identifies the role of the node in the SOAP message that caused the fault to happen.
Detail	String	No	Detailed message of the SOAP fault in the SOAP body. The content format must be well-formed XML.
msgDirection	String	No	Currently not used.
standardID	String	No	The name of the protocol: SOAP.
timestamp	String	No	Currently not used.
extraInfo	String	No	Currently not used.

Attachment

You can use TIBCO BusinessConnect SOAP Protocol to send and receive attachments in binary format, such as .jpg, .zip, or .doc, or in the text format. TIBCO BusinessConnect SOAP Protocol private process sends the attachments to BusinessConnect, or receives the attachments from BusinessConnect.

The attachment class in the messages has the following fields:

- **name** The name of the attachment
- **content-type** The type of the content associated with this attachment, such as text of binary
- **content-id** The content ID for the attachment
- **content** Content of the attachment
- **originalContentType** The original received content type

This field is only available for the `ResponderRequest` and `InitiatorResponse` private process messages. It has the same value as the **content-type** field, unless the **content-type** field cannot be inferred by Business Connect. In that case, the **content-type** field is changed to `application/octet-stream` and the **originalContentType** field is set to the actual value that is received.

All attachment fields are defined as strings, which require that the binary content be specified as base64 encoded.



MTOM and attachments from the private process cannot be used together. If the **SOAP MTOM Enabled** check box is selected, meanwhile some attachments are also specified from the private process request message, the request is rejected and the outbound process is ended.

See [Configuring General Properties for a Partner on page 100](#) on how to select SOAP MTOM.

Multipart MIME Messages

When attachments are present, SOAP messages are formulated as multipart Multipurpose Internet Mail Extensions (MIME) messages and TIBCO BusinessConnect SOAP Protocol depends on the parameters in the **content-type** field for packaging.

MIME messages are configured as follows:

- The first message part carries the SOAP envelope.

- Other message parts carry the attachments. Each message part carries the following header information:
 - **Content-Type** Indicates the type of the content embedded in the part.
 - **Content-Disposition** Indicates the presentation style used for the MIME part.
 - **Content-Transfer-Encoding** Indicates the encoding used for the MIME part.
 - **Content-ID** Indicates the content from anywhere in the multipart SOAP message.

Determining the Content Type of an Attachment

The value in the **content-type** field must be defined in the `type/subtype` format, such as `text/xml`, `application/zip`, and so on.

TIBCO BusinessConnect SOAP Protocol processes the attachment content based on the value in the **content-type** field as follows:

- Content is processed as plain text and the content type is referred to as `text` if it is in the `text/*` or `application/*xml*` formats.
- Content is processed as binary and the content type is referred to as `binary` if it is in the `application/binary`, `application/octet-stream`, `application/zip`, `application/pdf`, `application/doc`, `image/*`, `audio/*`, or `video/*` formats.
- Content type that is not in text or in binary format is processed as `unknown` and referred to as `unknown`. Content is packaged or interpreted as either text or binary, depending on the situation.
- If the value in the **content-type** field does not conform to the Request for Comments (RFC) standard, it is changed to `application/octet-stream` and the content is processed as `binary`.

The only exception to this rule is when the value in the **content-type** field is defined as `binary` and is subsequently changed to `application/binary`. This change is made to maintain backward compatibility with TIBCO BusinessConnect SOAP Protocol 5.0.

The value in the **content-type** field for attachments must conform to MIME [RFC2045] standards, for example, it has to be in the `type/subtype` format. For complete syntax of this property, see RFC 2045 at <http://www.ietf.org/rfc/rfc2045.txt>.

Outbound Attachments

Outbound attachments are populated for the outbound request or response messages such as `InitiatorRequest` or `ResponderResponse`.

If the value in the **content-type** field is not specified, it is determined based on the type of content:

- If content is of type `String`, the value in the **content-type** field defaults to `text/plain`.
- If content is of type `byte Array`, the value in the **content-type** field defaults to `application/octet-stream`.

Content for the outbound attachments is packaged as a byte array as follows:

- If the value in the **content-type** field is `text`, content is converted to bytes and the attachment is populated.
- If the value in the **content-type** field is `binary`, content is base64 decoded if it was previously base64 encoded. Otherwise, content is converted to bytes and the attachment is populated.
- If the value in the **content-type** field is `unknown`, the content type is changed to `application/octet-stream`, content is converted to bytes, and the attachment is populated.

Inbound Attachments

Inbound attachments are received from public process multipart MIME SOAP messages, which can carry the attachment content either in text or binary format.

Since the private process has to receive the attachment content as a string, TIBCO BusinessConnect SOAP Protocol handles the attachments with the different content types as follows:

- If the attachment content type is `text`, content is sent to the private process as text.
- If the attachment content type is `binary`, content is base64 encoded and sent to the private process.
- If the attachment content type is `unknown`, content is base64 encoded and sent to the private process. In this case, the value in the **content-type** field is changed to `application/octet-stream` and the content is processed as binary. For example, even if the plain text is received with the value in the **content-type** field described as `unknown`, this content is base64 encoded.

See [Converting Attachments to Base64 Strings on page 170](#).

Converting Attachments to Base64 Strings

Attachments are converted to `base64` string as follows:

1. Content-Transfer-Encoding is inspected. If it is `base64`, content is sent as `base64` to the private process.
2. Content itself is inspected for `base64` characters. If all the characters are in the `base64` character range, content is sent as `base64`. Otherwise, content is base64 encoded and sent.

OriginalContentType

For inbound attachments, if the content type is unknown, it is changed to `application/octet-stream`. The same value is passed on to the private process in the **content-type** field. The original value received is populated in the **originalContentType** field.

For text and binary content types, the **content-type** and **originalContentType** fields for the private process have the same value. The **originalContentType** field is populated even when the attachment property from the release 5.0 is preserved.

Preserving the SOAP 5.0 Attachment Functionality

TIBCO BusinessConnect SOAP Protocol version 5.0 processes the attachments as plain strings. For binary attachments, this could cause a problem.

From 5.1 release and onward, the behavior has been changed so that the binary attachments are converted to base64 encoding before they are sent to private process if they are not in base64 encoding originally.

However, you can still keep version 5.0 behavior for a particular partner by setting the TIBCO BusinessConnect SOAP Protocol Boolean property:

```
bc.soap.<tpname>.enable50Attachment = true
```

where `<tpname>` is the name of the trading partner. When enabled, the original content type that was received is sent to the private process.

Attachment Type

The attachment node (`ac/BC/attachment`) has the following fields:

Table 40 Attachment Type

Field	Type	Description
name	String	<p>Name of the attachment file.</p> <p>If the name of the attachment is not provided, the attachment name is auto-generated as <code>attachmentindex.content-type</code>.</p> <p>Example: If the second attachment file name is missing and <code>content-type=image/jpeg</code>, and then the name generated for that attachment is <code>attachment2.jpeg</code>.</p>
content-type	String	<p>Content type of attachment.</p> <p>Examples: <code>plain/text</code> and <code>application/binary</code>.</p>
content-id	String	<p>The unique identifier used to identify an attachment in outgoing document.</p>
content	String	<p>Actual attachment.</p> <p>All text-based payloads and attachments from the private process to BusinessConnect must be encoded in UTF-8.</p> <p>Binary attachments must be base64 encoded.</p>
originalContentType	String	<p>The original received content type is populated in this field.</p> <p>The field is available only for the <code>ResponderRequest</code> and <code>InitiatorResponse</code> private process messages.</p>

TradingPartner Type

The tradingPartner node (`ac/BC/tradingPartner`) has the following fields:

Table 41 Trading Partner Type

Field	Type	Description
name	String	Trading partner name.
domain	String	Trading partner domain.
id	String	Trading partner ID.

Attributes

The Attributes node (ae/SOAP/Attribute) has the following fields:

Table 42 Attribute Node

Field	Type	Description
name	String	The name of the attribute.
value	String	The value associated with the name.

SOAPFault

The `soapFault` node (`ac/SOAP/soapFault`) has the following fields:

Table 43 SOAP Fault Node

Field	Type	Description	
Code			
Value	String	Specifies the value from the set of SOAP fault: VersionMismatch, MustUnderstand, DataEncodingUnknown, Sender, and Receiver.	
SubCode	String	Specifies the value of the subcode to check detailed fault information.	
Reason			
ReasonText			
	lang	String	Specifies the language code used for the fault reason.
	content	String	Specifies the cause of the fault.
Node	String	Identifies the node in which the fault occurred.	
Role	String	Identifies the role of the node in the SOAP message that caused the fault to happen.	
Detail	String	A detailed message of SOAP fault in the SOAP body. The content format must be well-formed XML.	



For TIBCO BusinessConnect SOAP Protocol, the TIBCO ActiveMatrix BusinessWorks private process supports the SOAP fault only in version 1.2. If you want a SOAP fault in version 1.1, perform the following steps:

1. Populate the value in the `soapFault > Code > Value` field as your `faultCode`.
2. Populate the value in the `soapFault > Reason > ReasonText > content` field as your `faultString`.
3. Populate the value in the `soapFault > Role` field as your `faultActor`.

After performing these steps, BusinessConnect converts the SOAP fault from version 1.2 to 1.1.

WS-Addressing

The WS-Addressing node (`ac/SOAP/WS-Addressing`) has the following fields:

Table 44 *WS-Addressing Node*

Field	Type	Description
MessageID	String	The unique URI of the message.
To	String	Specifies the receiver URI of the message.
From	N/A	Specifies the endpoint of the service that dispatched the message.
ReplyTo	N/A	Specifies the endpoint of the receiver for reply messages.
FaultTo	N/A	Specifies the endpoint of the receiver for fault messages.
Action	String	The specific semantic operation of the message.
RelatesTo	N/A	A pair of values that specify the relationship between this message and another message. The original request MessageID must correspond with this property.



- If MessageID is not specified from the private process, or from any inbound request or response, all the properties are ignored.
- In the response message replied back to the initiator, the To property is set to the same as the ReplyTo property of the original request message. However, if the private process specifies the WS-Addressing To property for the response message, it is used to override the ReplyTo property in the original request message.
- The RelatesTo property of the response message replied to the initiator is set to the same as the MessageID property of the original request message.

The From, ReplyTo, and FaultTo properties has the following fields, as shown in Table 45.

Table 45 *The From, ReplyTo, and FaultTo Properties*

Field	Type	Description
Address	String	Specifies the URI address of the endpoint.
ReferenceProperties	String	Specifies the required properties for identifying the transferred entity or resource.

Table 45 The From, ReplyTo, and FaultTo Properties (Cont'd)

Field	Type	Description	
ReferenceParameters	String	Specifies the elements used to associate with the endpoint to boost an interaction.	
PortType	String	Specifies the primary PortType of the endpoint being conveyed.	
ServiceName			
	PortName	String	Specifies the name of the <wsdl:port> definition that interacts with the referenced endpoint.
	value	String	Specifies the <wsdl:service> definition that contains a WSDL description of the referenced endpoint.

The RelatesTo property has the following fields, as shown in Table 46.

Table 46 The RelatesTo Property

Field	Type	Description
RelationshipType	String	Specifies the relationship type between this message and another message.
value	String	The specific content of the RelateTo property.

XOP

The `xop` node contains the **tagName** field, and you can type the specific value in the **value** field when expanding the **tagName** field.

The `xop` node (`ac/SOAP/xop`) has the following fields:

Table 47 XOP Node

Field	Type	Description
tagName	Sequence of tagName	Specifies the tag names of the element node that is optimized with MTOM.

See [Configuring General Properties for a Partner on page 100](#) on how to select SOAP MTOM, and see also [MTOM on page 22](#) for more information.

AuthenticationToken

The authenticationToken node (ac/SOAP/authenticationToken) has the following fields:

Table 48 AuthenticationToken Node

Field	Type	Description
userName	String	The user name of the token for authentication.
password	base64Binary	The password of the token for authentication.



If you select the **Require UsernameToken Authentication** check box, the user name and password must be typed in the private process.

Appendix A **Status Codes**

This appendix describes the possible values in the **statusCode** and **statusMsg** fields. It also explains schema validation errors.

Topics

- [statusCode and statusMsg Field Reference, page 182](#)

statusCode and statusMsg Field Reference

The **statusCode** and **statusMsg** fields are used in private messages that are sent in response to a request. See [Responder Inbound Request — BusinessConnect to Private Process on page 157](#) and [Responder Outbound Response — Private Process to BusinessConnect on page 161](#) for examples of this.

The following values might be displayed in the **statusCode** and **statusMsg** fields:

Table 49 *statusCode and statusMsg Fields (Sheet 1 of 8)*

Code (statusCode)	Description (statusMsg)	Role	Category	Possible Resolution
The following status codes (200-500) are standard HTTP codes:				
200	OK			All status codes below 200 and above 299 are considered to be SOAP fault. When the value in the <code>statusCode</code> field is set to under 200 or above 299, the <code>SOAPFault</code> element is sent to the trading partner and the body is ignored even if it contains data. To send the content of the body, the value in the <code>statusCode</code> field must range between 200 and 299.
201-299	HTTP(S) OK codes			
300 - 499	HTTP(S) error codes	Error	System	
500	Internal Server error. SOAP fault occurred.	Error	System	Check fault code, fault string, and fault reason.
510	Socket closed.	Error	System	
The 600-699 status codes are error related to Web Services Security:				
601	Generic errors occurred when processing the message for security.	Error	Security	
650	Message encryption failed.	Error	Security	Check the certificates or encryption algorithm. One of them might be invalid.

Table 49 statusCode and statusMsg Fields (Sheet 2 of 8)

Code (statusCode)	Description (statusMsg)	Role	Category	Possible Resolution
651	Message signing failed.	Error	Security	Check the identity used for signature or the signing algorithm. One of them might be invalid.
652	Signature verification failed.	Error	Security	Check the certificates used for signature verification. It might be wrong.
653	Message decryption failed.	Error	Security	Check the identity used for decryption. It might be wrong.
654	WSSecurity process upon inbound message failed.	Error	Security	Either the decryption or verification of the inbound message has failed. For more details, look in the status message.
656	UsernameToken authentication failed.	Error	Security	
657	User name or password is not supplied by private process.	Error	Security	
659	Unknown errors	Error	Security	Errors occurred while processing the message for security. For more details, look in the status message.
The 800-899 status codes are for errors trapped locally on the initiator side:				
800	Access Denied: <i>reason</i>	Error	Configuration	
801	Duplicate outgoing message detected	Error	Configuration	This is detected by the initiator process. Make sure unique IDs are used.
802	Transport <i>transport</i> is not supported by SOAP.	Error	Configuration	SOAP supports HTTP and HTTPS only.
803	Cannot process a synchronous EMAIL request	Error	Configuration	

Table 49 *statusCode* and *statusMsg* Fields (Sheet 3 of 8)

Code (status Code)	Description (statusMsg)	Role	Category	Possible Resolution
804	Cannot send <i>operation type</i> request to multiple trading partners	Error	Configuration	
805	Notification did not reach all trading partners.	Error	Configuration	
806	Received SOAP Fault. Fault String: <i>fault_string</i>	Error		
807	Errors occurred when reading a response.	Error		
820	Schema validation error: missing validating schema (request)	Error	Schema validation and message contents	Depending on what caused the error to occur, load either header, body schema or both.
821	Schema validation error: missing validating schema (response)	Error	Schema validation and message contents	Depending on what caused the error to occur, load either header, body schema or both.
822	Schema validation error: request not conforming to schema. <i>reason</i>	Error	Schema validation and message contents	This is detected by the initiator process. Make sure there is an agreement between the message and the schema registered in the Operation > Request Action tab of the Operations Editor . Or do not select Validate Message check box in the Operation > Request Action tab of the Operations Editor .

Table 49 statusCode and statusMsg Fields (Sheet 4 of 8)

Code (statusCode)	Description (statusMsg)	Role	Category	Possible Resolution
823	Schema validation error: response not conforming to schema. <i>reason</i>	Error	Schema validation and message contents	This is detected by the initiator process. Make sure there is an agreement between the message and the schema registered in the Operation > Response Action tab of the Operations Editor . Or do not select Validate Message check box in the Operation > Response Action tab of the Operations Editor .
824	Attachment processing error, msg: <i>message</i> (request from private process)	Error	Schema validation and message contents	
825	Attachment processing error, msg: <i>message</i> (response from private process)	Error	Schema validation and message contents	
827	Failed to set HTTP basic authorization header	Error	HTTP error	
831	Partner is not configured:	Error	Configuration	
841	Posting error: <i>reason</i>	Error	HTTP error	
842	Invalid transport URI	Error	HTTP error	The transport URL for the trading partner is not correct: check and correct the URL.
855	Errors occurred when processing the attachments.	Error	Schema validation and message contents	
856	Errors occurred when extracting the envelope attributes.	Error	Package Message	

Table 49 *statusCode and statusMsg Fields (Sheet 5 of 8)*

Code (statusCode)	Description (statusMsg)	Role	Category	Possible Resolution
857	Errors occurred when processing MTOM configuration: <i>reason</i>	Error	Configuration	
858	Errors occurred when processing SAML configuration: <i>reason</i>	Error	Configuration	
866	Received response document is denied because of encryption/signing permissions.	Error	Security	Check the Operations Editor settings.
867	Errors occurred when setting envelope attributes.	Error	Package Message	Verify that envelope attributes are correctly specified as described in the documentation.
872	Errors occurred when verifying or decrypting the response.	Error	Security	
888	Errors occurred when serializing SOAP message.	Error	Package Message	A problem happened when packaging the SOAP message. Verify the SOAP body and header elements.
890	No matching transaction found for ID <i>ID</i> . The request might have timed out.			
899	Errors occurred. No available detailed information.	Error	System	

The 900 - 999 status codes are for errors trapped by the responder and sent back to the initiator:

Table 49 statusCode and statusMsg Fields (Sheet 6 of 8)

Code (statusCode)	Description (statusMsg)	Role	Category	Possible Resolution
900	Trading partners are not permitted to access: <i>reason</i>	Error	Configuration	
901	Access Denied: <i>reason</i>	Error	Passthrough	Verify that your default host accepts anonymous messages by selecting the Allow Anonymous SOAP Messages check box for the default host.
910	Inbound transport is not enabled.			
920	Request parsing error	Error	Schema validation and message contents	
921	Missing validating schema	Error	Schema validation and message contents	Depending on what caused the error to occur, load either header, body schema or both.
922	Request not conforming to schema	Error	Schema validation and message contents	Depending on what caused the error to occur, load either header, body schema or both. This is detected by the responder process. Make sure there is an agreement between the message and the schema registered in the Operation > Request Action tab of the Operations Editor . Or do not select Validate Message check box in the Operation > Request Action tab of the Operations Editor .
923	Attachment processing error, msg: (request from trading partner)	Error	Schema validation and message contents	

Table 49 *statusCode and statusMsg Fields (Sheet 7 of 8)*

Code (statusCode)	Description (statusMsg)	Role	Category	Possible Resolution
924	Attachment processing error in response from private process, detail:	Error	Schema validation and message contents	
925	Errors occurred when decrypting or verifying the SOAP message.	Error	Security	Check the Operations Editor settings.
926	Non-Repudiation logging of signed request failed.			
927	Errors occurred when processing the response, detail:	Error		
928	Errors occurred when processing MTOM configuration: <i>reason</i>	Error	Configuration	
929	Errors occurred when processing SAML configuration:	Error	Configuration	
940	Client authentication failed.	Error	HTTP/Comm	
941	Errors occurred when receiving a response from private process. Detail:	Error	HTTP/Comm	
942	Failed to send reply to DMZ	Error	HTTP/Comm	Check the DMZ timeout parameters.
943	Errors occurred when sending a request to private process.	Error	System	Could not save context information in the database. Check database connection status.

Table 49 statusCode and statusMsg Fields (Sheet 8 of 8)

Code (statusCode)	Description (statusMsg)	Role	Category	Possible Resolution
944	Response not conforming to schema	Error	Schema validation and message contents	<p>Depending on what caused the error to occur, load either header, body schema or both.</p> <p>This is detected by the responder process. Make sure there is an agreement between the message and the schema registered in the</p> <p>Operation > Response Action tab of the Operations Editor. Or do not select Validate Message check box in the Operation > Response Action tab of the Operations Editor.</p>
990	No matching transaction found for ID <i>ID</i> . The request might have timed out.			
999	Errors occurred. No available detailed information.	Error	System	<p>This is detected by the responder process. Turn on tracing to capture the traces. Contact TIBCO Support and forward the traces to them.</p>
1000 - 1999	Private-party-defined error codes	Error		

Appendix B **Schema Validation**

This appendix explains how to perform schema validation with TIBCO BusinessConnect SOAP Protocol.

Topics

- [Overview, page 192](#)
- [Schema Validation Errors, page 193](#)

Overview

When schema validation is selected, it is performed on the SOAP header, SOAP body, or both.

When validation is used, at least one of the schemas, body or header must be populated. If schema validation is selected and the schema is not uploaded, the schema validation for body or header is ignored. Schema validation fails if schema is uploaded for body or header in TIBCO Administrator and message does not have that part.

Schema Validation Errors

This section explains SOAP fault and SOAP advisory messages that are displayed if schema validation fails.

Schema validation errors are accessed through **Log Viewer** by using TIBCO Administrator.

- If schema validation fails on the initiator side for an outbound SOAP message, the SOAP advisory message on the ERROR subject on the initiator side has the complete details of the schema validation errors. These details are posted in the `statusMsg` field of the SOAP/Advisory AE message.
- If the schema validation fails on the responder side for an inbound SOAP message, which means the SOAP message from the public process, a SOAP fault is sent to the trading partner. The SOAP fault detail has the complete schema validation error.

A sample 1.1 SOAP fault message structure with fault details looks as follows:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<SOAP-ENV:Body>
<SOAP-ENV:Fault>
<faultcode>SOAP-ENV:Client</faultcode>
<faultstring>Request not conforming to schema</faultstring>
<faultactor>SOAPServer</faultactor>
<detail>
<ei:ErrorInfo xmlns:ei="http://www.tibco.com/namespaces/bc/2003/10/errorinfo.xsd">
<code>922</code>
<description>Request not conforming to schema</description>
</ei:ErrorInfo>
</detail>
</SOAP-ENV:Fault>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

The schema for the `ErrorInfo` element is in the `errorinfo.xsd` file in the `TIBCO_HOME/bc/version_number/protocols/soap/samples/schemas` directory.

The SOAP advisory message is published both on the initiator and responder sides, with the `statusMsg` element containing the schema validation errors.

- If the schema validation fails when processing a response message from the private process on the responder side, a SOAP fault is sent with the fault string, `Response not conforming to schema`. This SOAP fault message does not have fault details.

A SOAP/Advisory AE message is published on the ERROR subject with the schema validation errors in the `statusMsg` element.

- If the schema validation fails when processing a response message on the initiator side, a SOAP/Advisory AE message is published on the ERROR subject with schema validation errors in the `statusMsg` element.

Appendix C **TIBCO BusinessConnect SOAP Protocol WSDL Tool**

This appendix gives an introduction to TIBCO BusinessConnect SOAP Protocol WSDL tool and shows how to use it to export and import WSDL files.

Topics

- [Overview, page 196](#)
- [WSDL Import, page 197](#)
- [WSDL Export, page 199](#)

Overview

You can use TIBCO BusinessConnect SOAP Protocol WSDL tool to import the WSDL files into TIBCO BusinessConnect configuration or export the TIBCO BusinessConnect configuration into the WSDL files.



The TIBCO BusinessConnect SOAP Protocol WSDL tool does not automatically set certificates for participants. After importing the WSDL files, you must explicitly set certificates for the HTTPS and HTTPSCA transports.

WSDL Import

To import the WSDL files into TIBCO BusinessConnect configuration, navigate to the `TIBCO_HOME/bc/version_number/protocols/soap/tools/wsdli` directory and run the following commands:

- **Windows** `wsdliimport.exe`
- **Unix** `./wsdliimport`



After importing the WSDL files, log off from TIBCO Administrator, and then log in to TIBCO Administrator to view the imported WSDL operations.

When importing the WSDL files with the WSDL tool, you must note the following:

- Operation bindings are created for the operations imported through the WSDL by selecting **BusinessConnect > Business Agreement > Agreement Name > SOAP > Host X can initiate** of that particular host and partner mentioned in the `wsdliimport.tra` file. Business agreement is created if it is not present before the import process. The **Override Transports** check box in the **Transports** tab of these operation bindings is selected, and primary transport is set to the transport created under partner for this particular operation.
- The **Require Digital Signature**, **Require Content Encryption**, and **Require UsernameToken Authentication** check boxes for that operation in **Operations Editor** are always unselected after the import process if at least one schema for that operation is in WSDL. The **Validate Message** check box for that operation in **Operations Editor** is always selected.

Importing WSDL in nested Schemas

If nested schemas are configured as file references, their location should not change compared to the configured location in the operations editor. This is due to the fact that the content of the schemas components are loaded into the BusinessConnect schema cache on-demand from the specified location. In addition, if the BC palette-based private process imports operations that have been configured with such referenced schema, the original location - with the same path - must be accessible from the importing project only for the duration of the update. After the update on the palette-based project is complete there is no further correlation between the original schema file resources and the imported schemas.

When you use nested schemas in BusinessConnect, use the file reference option and make sure that all dependant schemas are present within the same directory. This way the BusinessConnect palette will import the mentioned schemas to the Designer.

1. Set up the schema as a file reference in BusinessConnect.

- 2.Import it into BusinessWorks.
- 3.Import the rest of the nested schemas into a subdirectory in your BusinessWorks project.
- 4.Open the root schema in BusinessWorks. When it asks for the reference to a sub-schema, select the appropriate file in the imported BusinessWorks directory.

WSDL Import TRA File Configuration

TIBCO BusinessConnect SOAP Protocol WSDL import tool uses the `wsdlimport.tra` file in the `TIBCO_HOME/bc/version_number/protocols/soap/tools/wsdl` directory for configuration and command line arguments.

Table 50 and Table 51 describe the properties in the `wsdlimport.tra` file that you can use to modify for WSDL import process.

Table 50 BusinessConnect Database Configuration Properties for WSDL Import Process

Properties	Description
<code>java.property.bc.repo.db.user=database user</code> <code>java.property.bc.repo.db.password=database password</code>	Specify the user name and password of the database.
<code>java.property.bc.repo.db.url=JDBC URL</code> <code>java.property.bc.repo.db.driver=JDBC Driver</code>	Specify JDBC URL and driver of the database.

Table 51 WSDL File, Trading Partner, Local Host Name, and SOAP Version Properties for WSDL Import Process

Properties	Description
<code>java.property.wsdlfilename=WSDL filename</code>	Specify the WSDL file name to be imported. If the WSDL filename is a directory, all the WSDL files under this directory are imported.
<code>java.property.tradingpartner=trading partner name</code>	Specify the name of the trading partner.
<code>java.property.hostname=host name</code>	Specify the local host name. Host name to be used in a multi-host configuration. This value is used to create business agreement between this trading partner and host. If this property is not specified, the default host is used.
<code>java.property.wsdl.soap.version=SOAP version</code>	Specify the SOAP version of the imported WSDL file, such as SOAP1.1 or SOAP1.2. You can use this parameter to update the SOAP version of a trading partner in configuration store. The default SOAP version is SOAP1.1.

WSDL Export

To export TIBCO BusinessConnect configuration into the WSDL files, navigate to the `TIBCO_HOME/bc/version_number/protocols/soap/tools/wsd` directory and run the following commands:

- **Windows** `wsdlexport.exe`
- **Unix** `./wsdlexport`



To export schemas into a WSDL file, the root element has to be available. If no root element is available, schemas are not exported even if the schemas are uploaded in TIBCO Administrator.

WSDL Export TRA File Configuration

TIBCO BusinessConnect SOAP Protocol WSDL export tool uses the `wsdlexport.tra` file in the `TIBCO_HOME/bc/version_number/protocols/soap/tools/wsd` directory for configuration and command line arguments.

[Table 52](#) and [Table 53](#) describe the properties in the `wsdlexport.tra` file that you can use to modify for WSDL export process.

Table 52 BusinessConnect Database Configuration Properties for WSDL Export Process

Properties	Description
<code>java.property.bc.repo.db.user=database user</code> <code>java.property.bc.repo.db.password=database password</code>	Specify the user name and password of the database.
<code>java.property.bc.repo.db.url=JDBC URL</code> <code>java.property.bc.repo.db.driver=JDBC Driver</code>	Specify JDBC URL and driver of the database.

Table 53 WSDL File, Endpoint, SOAP Version, Service, Namespace, and List Properties for WSDL Export Process

Properties	Description
<code>java.property.wsd.filename=wsdl file name</code>	Specify the WSDL file name.
<code>java.property.wsd.url=url</code>	Specify WSDL endpoint URL. For example: <code>http://SOAPServer:6700/dmz/SOAP</code>
<code>java.property.wsd.soap.version=SOAP version</code>	Specify the SOAP version of the exported WSDL file, such as SOAP1.1 or SOAP1.2. The default SOAP version is SOAP1.1.

Table 53 WSDL File, Endpoint, SOAP Version, Service, Namespace, and List Properties for WSDL Export Process (Cont'd)

Properties	Description
java.property.wsdl.service= <i>service name</i>	Specify the WSDL service name.
java.property.wsdl.namespace= <i>wsdl namespace</i>	Specify the WSDL namespace.
java.property.wsdl.operationlist= <i>export list filename</i>	<p>Specify the WSDL export list.</p> <p>This file contains list of interface / version that can be exported into the WSDL file.</p> <p>Each line of this file must have one interface / version.</p> <p>Format of each line is as follows:</p> <p><i>interface name / version name</i></p> <p>Interface and version name separated by '/'.</p>

Index

A

- adding a domain identity 98
- adding a domain identity for a partner 102
- adding a version to an interface 78
- adding an operation to a version 79
- advisory message 166
- anonymous SOAP messages 98
- attachment
 - determine the Content-Type property 169
 - formats 168
 - object fields 172
 - preserve SOAP 5.0 functionality 171
- attachments
 - inbound 170
 - outbound 169
- audit logs 121
- authenticationToken 179

B

- BusinessConnect SOAP Protocol features 4
- BusinessWorks private process tutorial
 - configuring private processes 54
 - modify the PNotify operation 92
 - modify the POSync operation 93
 - overview 50
 - prerequisites 53
 - responder process definitions 60
 - running private processes 65
 - running the tutorial 65
 - setting up 53
 - viewing the audit logs 67

C

- configure a non-repudiation log 127
- configuring the SOAP protocol 98
- configuring the SOAP protocol for a partner 100
- configuring transports for partners 103
- creating a SOAP interface 76

E

- edit an operation binding for the partner 111
- editing an operation binding for the host 108
 - action settings tab 109
 - operation settings tab 108
 - transports tab 111
- editing an operation binding for the partner
 - action settings tab 112
 - operation settings tab 111
- exchanging identity information 72
- exchanging URI definitions 71
- exporting the SOAP interface 91

I

- import the SOAP interface 76
- importing interfaces 90
- initiator inbound response — BusinessConnect to private process 154
- initiator outbound request — private process to BusinessConnect 149

M

managing SOAP envelope attributes 144
 MTOM 22, 102
 multipart MIME messages 168

N

non-repudiation logs 127
 notify operation 10
 notify transaction process flow 10

O

objects included in private messages 168
 operation
 notify 10
 request-response 10
 operation bindings tab 107
 operation properties 80
 operation properties, action tab 81, 84
 operation properties, general tab 80
 operation types and process flows 10
 OriginalContentType 171

P

private messages and processes 8
 public messages 9

R

request-response operation 10
 resend logs 129
 resendable transactions 129
 responder acknowledgement — BusinessConnect to private
 process 165

responder inbound request — BusinessConnect to private
 process 157
 responder outbound response — private process to
 BusinessConnect 161

S

SAML 19
 schema validation errors 147
 setting general properties 98
 setting transports for the business agreement 116
 SOAP envelope attributes and namespaces 144
 SOAP fault 23
 SOAP overview 2
 SOAP version 101
 SOAP 1.1 protocol 101
 SOAP 1.2 protocol 101
 SOAPFault 175
 SOAPFault object fields 175
 standalone tutorial
 configuring the business agreement 35, 40
 configuring the initiator 30
 configuring the responder 36
 initiator machine 28
 operations 26
 overview 26
 prerequisites 29
 responder machine 28
 running the tutorial 41
 setting up initiator server 33
 setting up responder server 39
 setting up the responder as a trading partner 33
 trading partners 26
 transports 26
 viewing the audit logs 47
 state field values for logs 123
 statusCode and statusMsg field reference 182
 synchronous request-response operations 11

T

TIBCO_HOME [xvi](#)

trading with a third-party SOAP implementation [134](#), [134](#)

U

UsernameToken for authentication [17](#)

W

WS-Addressing [20](#)