TIBCO BusinessConnect™

Interior Server Administration

Version 7.4.0
May 2023

Copyright © 2001-2023. Cloud Software Group, Inc. All Rights Reserved.

Contents

Contents **2**

Introduction **6**

Interior Server Overview. 6

Interior Server Quick Start 7

Installing the Interior Server 7

Configuring Deployment 8

Configuring Private Processes 8

Deploying the Interior Server 8

Deployment Configuration **9**

Overview 9

Step 1. Editing Application Configuration 9

Intercomponent Communication 9

Intercomponent Advanced 11

Intercomponent JMS Settings 12

Intercomponent DMZ-JMS Settings (Optional) 15

Step 2. Configuring Interior Server.par 16

General Tab 16

Fault Tolerance Tab 17

Monitoring Tab 19

Advanced Tab 19

Step 3: Configuring Smart Routing 20

20

Configuring Smart Routing Rules 20

Map Clusters 27

Private Process Configuration **31**

Private Process Communication ... 31
Selecting the Private Process Transport ... 32
JMS Transport ... 33
Configuring JMS Transport ... 33
TIBCO Rendezvous ... 39
Configuring TIBCO Rendezvous ... 39
Intercomponent Communication Rendezvous Settings ... 42
Outbound File Pollers ... 43
Enabling and Configuring Outbound File Poller ... 44

Interior Server Deployment ... 46
Before You Deploy ... 46
46
46
Deploying and Starting the Interior Server ... 47

Manage the Interior Server ... 50
Editing Application Configuration ... 50
BusinessConnect Tab ... 50
General Tab ... 51
Advanced Tab ... 52
Tracing ... 52
Tracing for TIBCO Administrator ... 52
Tracing for the TIBCO BusinessConnect Engine ... 53
TIBCO LogLogic Integration ... 53
Checking the State of the Interior Server Instance ... 56
Starting and Stopping the Server ... 56
Removing TIBCO BusinessConnect ... 57
Undeploying TIBCO BusinessConnect ... 57
Deleting TIBCO BusinessConnect ... 57
Hawk Microagents for Interior Server ... 57

Database Schema Definition **59**
Audit Schema Details 59
BC_TRANSACTIONS 66
BC_MESSAGES 75
BC_MESSAGES_BIN 84
BC_ALERT 87
BC_RESEND_BIN 90
BC_UACLOG 93
BC_LOGACL_TEMP 96
BC_DUP 97
BC_UACLOG_DETAIL 99
BC_POLLER_INFO 101
Non-Repudiation Schema Details 103
BC_NR_BIN_SIGNATURE 106
BC_NR_BIN_EDCRED 108
BC_NR_BIN_SIGCRED 109
BC_NR_TRANSACTIONS 112
BC_NR_MESSAGES 117
BC_NR_BIN_VALICERT 123
BC_NR_BIN 125
Runtime Schema Details 127
BC_SFWS_ATTACHMENTS_BIN 130
BC_SFWS_MESSAGES 135
BC_SCHEDULED_TASK 140
BC_SFWS_TPINFO 145
BC_MDN 150
BC_HIBERNATION 152
BC_ 156
BC_HIBERNATION_BIN 159
BC_LOGQUERYBIN 161
Configuration Store Reporting Schema Details 164

BC_CS_PROTOCOL_PROPERTIES 168
BC_CS_TRANSPORT_PROPERTIES 169
BC_CS_BA_PROTOCOL_PROPERTIES 171
BC_CS_BA_STMS 173
BC_CS_BA_OB_PROPERTIES 174

TIBCO BusinessConnect Palette Projects 177
Deploying TIBCO BusinessConnect Palette Projects 177

Deployment Tuning 178

Deploying the Interior Server on two Subnets 179

Modifying Load Balancing Properties 181

Editing bcengine.tra to Tune Load Balancing 185

TIBCO Documentation and Support Services 186

Legal and Third-Party Notices 188

Introduction

This section introduces TIBCO BusinessConnect Interior Server and explains its functionality.

Interior Server Overview.

TIBCO BusinessConnect Interior Server is located inside the company's firewall and performs the following tasks:

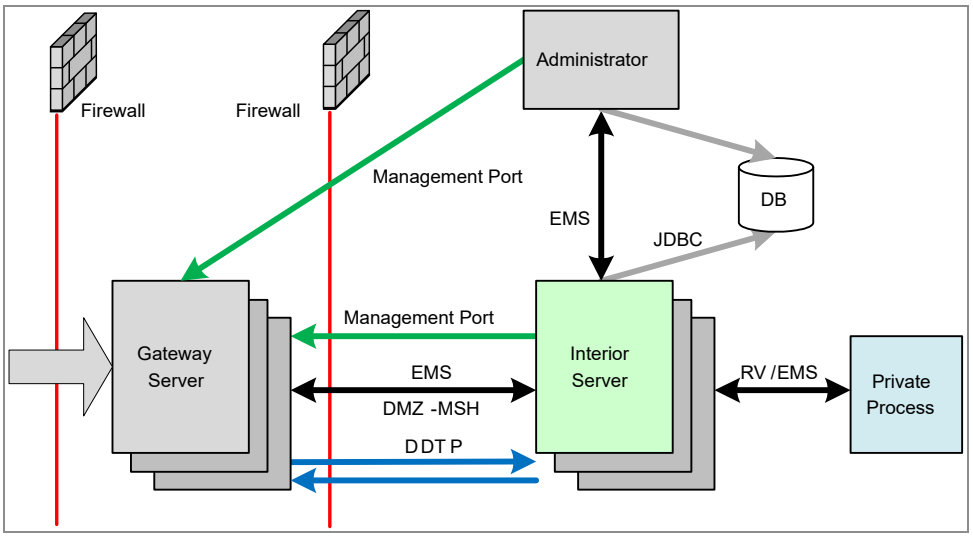
- Handles all messaging level activities such as encryption, decryption, and digital signatures.
- Takes care of business level logic to be executed by an individual protocol, such as document schema validation.

TIBCO BusinessConnect Interior Server is the runtime component that is deployed on top of the other required TIBCO software products.

For the list of all required TIBCO and other Third-party software products, see *TIBCO BusinessConnect ™ Installation and Configuration*, Installation Requirements.

A diagram of the Gateway Server and Interior Server communications is depicted in the following figure.

Figure 1: TIBCO BusinessConnect Interior and Gateway Server Architecture



The Interior Server must be deployed and started before the Gateway Server.

For more details about the Interior Server, see *TIBCO BusinessConnect™ Concepts*, Interior Server and Gateway Server Architecture.

Interior Server Quick Start

To install, configure, deploy, and start the Interior Server, follow these steps:

Installing the Interior Server

1. To install the Interior Server, follow the steps described in *TIBCO BusinessConnect Installation and Configuration,* Installation and Uninstallation
   - Preinstallation Tasks
   - Installation
2. To create an installation on the Interior Server, follow the steps described in *TIBCO BusinessConnect Installation and Configuration,* Postinstallation:
   - Managing Permissions for the Installation
   - Initializing a Database

Configuring Deployment

To prepare the Interior Server for deployment, see:

- Deployment Configuration

Configuring Private Processes

To configure private processes, see:

- Private Process Configuration

Deploying the Interior Server

After the private processes and server are configured, you can deploy and start the Interior Server:

- Interior Server Deployment

Deployment Configuration

This section describes how to configure deployment of the Interior Server using TIBCO Administrator.

Overview

To prepare the Interior Server for deployment, start as follows:

1. Select **BusinessConnect** and click **Manage.**
2. **In the Configuration Repository tab, Create Deployment section, click Create**.
3. When the Success dialog opens, click **OK**.
4. Continue with Step 1. Editing Application Configuration.
5. Add machines as explained in Step 2. Configuring Interior Server.par.
6. Continue with Step 3: Configuring Smart Routing .

Step 1. Editing Application Configuration

To edit an application's configuration:

1. Expand **Application Management > BusinessConnect > Configuration**.
2. Click the **BusinessConnect** link.
3. Select the **Component Settings** tab.

Intercomponent Communication

1. Click the **Intercomponent Communication** link.

   The default values listed in Editing Intercomponent Communication Settings will be used unless you specify other values. To learn more about network default values,

refer to the document *TIBCO Rendezvous Administration*, Default Port and Service Numbers.

Editing Intercomponent Communication Settings

Field	Description
Interior Settings	
Service	Service port specified by the user. Example: 8700 (default is 7500)
Network	IP address on which the service is running. Example: 190.100.0.10
Daemon Host	IP address of the daemon host, which is the same one on which the service is running. Example: 10.100.100.30
Daemon Port	Example: 7500 (default is 7500)
Interior Queue Configuration	
Scheduler Heartbeat (seconds)	Default is 5. The active scheduler sends heartbeat messages at the interval you specify (in seconds). Heartbeat messages inform other members that a member is acting as the scheduler. All members of a group must specify the same scheduler heartbeat interval. This option is not saved (migrated) when exporting an installation configuration and importing the Domain_name.csx file into another installation. See *TIBCO Rendezvous Concepts*, Scheduler Parameters for more details.
Scheduler Activation (seconds)	Default is 15. Defines the amount of time after which the scheduler will be activated.

2. Enter the required values as desired and click **Save**.

Intercomponent Advanced

1. To configure advanced settings, such as to specify locations for shared and temp files other than the default, follow the descriptions in Editing Intercomponent Advanced Settings.

Editing Intercomponent Advanced Settings

Filed	Description
Interior Settings	
Shared Temporary Directory	Pre-populated with the default location: **./** You can enter any valid, available directory location to which TIBCO BusinessConnect can have access to write messages. For example, */BC_HOME*/shared_temp (for Windows: C:\tibco\bc\6.3\shared_temp). Verify that this directory is accessible by all deployment nodes and by Private Processes. It should be accessible to all Interior engines if they are grouped for load balancing. The following files are stored in this directory: • Inbound FTP and Email messages (if their size exceeds the threshold) • Inbound messages (in the form of a file) published to Private Processes Make sure that you have separate locations for: • Shared folder • Temporary folder • Outbound File Poller • Inbound File Poller
Local Temporary Directory	Pre-populated with the default location: **./** You can enter any valid, available directory location to which TIBCO BusinessConnect can have access to write messages. For example, */BC_HOME*/local_temp (for Windows: C:\tibco\bc\6.3\local_

Filed	Description
	temp). Verify that this directory is accessible from all machines in the deployment. If this directory is not defined, temporary files will be stored in the root directory or in the local Java directory. Make sure that you have separate locations for: • Shared folder • Temporary folder • Outbound File Poller • Inbound File Poller
Interior Component Wait Time (seconds)	The default is 7200 seconds. This option specifies the timeout for exchanging internal messages between the Gateway and the Interior Server. If the system is heavily loaded with processing messages and it is not possible to deploy additional load-balancing engines to the domain, this property may be used to adapt to the increased response time of internal processes. In most cases, the default value of 2 hours should be considered as acceptable.

Intercomponent JMS Settings

1. To configure JMS settings and set the connection parameters for the EMS server, follow the descriptions in Editing Intercomponent JMS Settings.

Editing Intercomponent JMS Settings

Field	Enter
Intercomponent JMS Settings	

Field	Enter
Protocol Prefix	`jms://`
JMS User Name	Username to use when logging into the JMS server. If the JMS provider does not require access control, this field can be empty. Not all JMS servers require user names and passwords. Refer to your JMS provider documentation and consult your system administrator to determine if your JMS server requires a username and password.
JMS Password	Password to use when logging into the JMS server. If the JMS provider does not require access control, this field can be empty.
JNDI Context Factory	The initial context factory class for accessing JNDI. (`javax.naming.Context.INITIAL_CONTEXT_FACTORY`). **Note**: TIBCO BusinessConnect attempts to find the class. However, you may need to add the Java file supplied by your JNDI service provider to the `CLASSPATH` environment variable to use JNDI.
JNDI Context URL	This is the URL to the JNDI service provider (`javax.naming.Context.PROVIDER_URL`). See your JNDI provider documentation for the syntax of the URL.
JNDI User Name	Username to use when logging into the JNDI server (`javax.naming.Context.SECURITY_PRINCIPAL`). If the JNDI provider does not require access control, this field can be empty.
JNDI Password	Password to use when logging into the JNDI server (`javax.naming.Context.SECURITY_CREDENTIALS`). If the JNDI provider does not require access control, this field can be empty.

Field	Enter
Topic Connection Factory	The TopicConnectionFactory object stored in JNDI. This object is used to create a topic connection with a JMS application. See your JNDI provider documentation for more information about creating and storing TopicConnectionFactory objects.
Queue Connection Factory	The QueueConnectionFactory object stored in JNDI. This object is used to create a queue connection with a JMS application. See your JNDI provider documentation for more information about creating and storing QueueConnectionFactory objects.
Reconnect Max. Duration (mins)	This is the time during which the TIBCO BusinessConnect server will try to reconnect. After this time, there will be no attempt to reconnect. This duration time does not represent the reconnection frequency. Default is 10 minutes.
Secured	If selected, the transaction will be secured.
Verify JMS Server	If selected, the JMS server's identity (that is, its X509 certificate as well as the specified value in the “Expected JMS Server Host Name” field) will be verified against the data received during the SSL handshake. If either the trusted CA certificates or the expected hostname doesn't match, the transport creation fails. If this verification is not required, BC can establish a JMS connection with any TIBCO Enterprise Message Service, whose credentials are different from the configured properties.
JMS Server Certificate	The certificate credential of the JMS server. To create this certificate, follow the steps described in *TIBCO BusinessConnect Trading Partner Administration*, Adding LDAP/JMS/Email Server Certificates The credential is stored in the TIBCO BusinessConnect keystore and is expected to be configured on the TIBCO Enterprise Message Service server according to the corresponding guidelines.

Field	Enter
Expected JMS Server Host Name	The value of the common name component of the TIBCO Enterprise Message Service server's leaf certificate. This is usually the hostname of the resource, running the TIBCO Enterprise Message Service server. If it is a test system, the common name (CN) value may be any arbitrary string, which must match the value of this field if the “Verify JMS Server” checkbox is checked.

Strong Ciphers Only	If the box is checked, only strong encryption algorithms will be used between the server (or the palette) and the JMS provider. The below cipher suites are offered by the connecting client (either bc or the palette) in this mode:
	`TLS_RSA_WITH_AES_256_CBC_SHA`
	`TLS_RSA_WITH_AES_128_CBC_SHA`
	`TLS_DHE_RSA_WITH_AES_256_CBC_SHA`
	`TLS_DHE_RSA_WITH_AES_128_CBC_SHA`
	`SSL_RSA_WITH_RC4_128_SHA`
	`SSL_RSA_WITH_3DES_EDE_CBC_SHA`
	`SSL_DHE_RSA_WITH_3DES_EDE_CBC_SHA`
	`SSL_DHE_DSS_WITH_3DES_EDE_CBC_SHA`
	`TLS_DHE_DSS_WITH_AES_128_CBC_SHA`
	`TLS_DHE_DSS_WITH_AES_256_CBC_SHA`
	Note: The unlimited strength JCE jurisdiction policy files are pre-installed on the TIBCO Java Runtime Environment (JRE).

Use Trace	See comments in *TIBCO ActiveMatrix BusinessWorks Palette Reference*, JMS Palette section Advanced. When this option is used, the SSL-specific debug tracing for the secure JMS transport will be sent to the engine standard output only.

2. Click **Test Connection**.

Intercomponent DMZ-JMS Settings (Optional)

You can optionally configure an EMS server in the DMZ, which is dedicated for the communication between the Gateway and Interior servers. For more details on configuring

component settings for Intercomponent DMZ-JMS Settings, see Intercomponent JMS Settings

Note The component settings for Intercomponent DMZ-JMS Settings are exactly same as that of Intercomponent JMS Settings.

Step 2. Configuring Interior Server.par

In this step, you will determine which particular machine(s) will be running the Interior Server, which can be assigned to multiple machines to achieve both fault tolerance and load balancing.

1. Expand **Application Management > BusinessConnect > Configuration**.
2. Click the **Interior Server.par** link.

   The Edit Service Configuration: Interior Server.par dialog appears.

General Tab

This tab allows you to add the Interior Server to machines.

1. In the General tab, click **Add to Additional Machines**.

   The Bind to Container(s) dialog appears.
2. Select the checkbox next to the machine on which you want to deploy the Interior component and click **OK.**
3. Repeat step 2. one or more times to add additional machines.

Note If you only add additional machines, they will provide for load balancing. To achieve fault tolerance, you must group two or more machines into groups.

Each time you add a machine, you will define a service instance. These instances can then be grouped to achieve fault tolerance and load balancing.

Enable Service

When this checkbox is checked, it will enable services for the Interior Server.

Fault Tolerance Tab

This tab allows you to group machine to achieve fault tolerance.

1. Select the **Fault Tolerance** tab.

   The Edit Service Configuration: Interior Server.par dialog appears.

2. In the **Group Settings** panel, enter the values for the new fault tolerant group by using Fault Tolerance Configuration.

Fault Tolerance Configuration

Field	Enter
Service	Service used by the fault tolerance daemon
Network	Network used by the fault tolerance daemon
Daemon Host	Host used by the fault tolerance daemon
Daemon Port	Port used by the fault tolerance daemon
Heartbeat Interval (seconds)	The master engine of a fault-tolerant group broadcasts heartbeat messages to inform the other group members that it is still active. The heartbeat interval determines the time (in seconds) between these heartbeat messages. If the master engine fails, the other engine detects the stop in the master's heartbeat and resumes operation in its place. All process starters are restarted on the second machine, and services are restarted to the state of their last checkpoint. Default is 5.
Activation Interval (seconds)	All secondary process engines track heartbeat messages sent from the master engine. This field specifies the amount of time between the last heartbeat from the master engine and the re-starting of the process starters and process engines on the secondary engine. The Heartbeat

Field	Enter
	Interval should be smaller than the Activation Interval. It is recommended that the Activation Interval be slightly over two heartbeats. Activation Interval is a standard TIBCO Rendezvous fault tolerant parameter, documented in *TIBCO Rendezvous Concepts*, Developing Fault Tolerant Programs. Default is 15.
Activation Delay (seconds)	When the master engine resumes operation, the secondary engine shuts down and returns to standby mode. In some situations it may be necessary to ensure that the secondary engine has completely shut down before the master engine resumes operation. This field is used to specify a delay before the master engine restarts. When the time after the last heartbeat from an active member exceeds this value, the ranking inactive member will receive a “hint” so that it can prepare for activation. The Heartbeat Interval should be smaller than the Activation Delay, which should be smaller than the Activation Interval. Default is 10.

Load Balancing

In the **Service Instance Name-Fault Tolerance Group Settings** panel, you will see all machines to which the Interior component was assigned using General Tab.

machine1 - `Interior Server` *domain*.`BCFTGROUP.A`

machine2 - `Interior Server` *domain*.`BCFTGROUP.B`

machine3 - `Interior Server` *domain*.`BCFTGROUP.C`

machine4 - `Interior Server` *domain*.`BCFTGROUP.D`

Machines that belong to one group provide for fault tolerance within that group, while machines in different groups provide for load balancing among these groups.

Fault Tolerance

In order to form groups for fault tolerance, decide which machines you will group together. For example, you will assign `machine1` and `machine2` to the group A and `machine3` and `machine4` to the group B.

3. Click the link for the machine that you want to re-assign to another group.

The Edit FT Node Settings dialog appears.

4. In the **FT Group Name** field, correct the existing entry so that the machine can be re-assigned to another group.

   For example, instead of *doman*.BCFTGROUP.B, make it *doman_name*.BCFTGROUP.A.

5. Click **Save**.

6. Repeat steps for any other machines that need to be re-assigned.

   Once you are finished with re-assigning, verify that you have the appropriate group assignments in the **Service Instance-Fault Tolerance Group Settings** panel, such as

   machine1 - Interior Server *domain*.BCFTGROUP.A

   machine2 - Interior Server *domain*.BCFTGROUP.A

   machine3 - Interior Server *domain*.BCFTGROUP.B

   machine4 - Interior Server *domain*.BCFTGROUP.B

7. Click **Save**.

   The Configuration dialog confirms that the Interior Server component is assigned to multiple machines and is deployable.

Monitoring Tab

The Monitoring tab allows you to manage the Rule Base for load balancing groups as well as events, such as component failures, alerts, email status, and various commands.

For more details about rulebases and events, see *TIBCO Administrator™ User Guide*.

Advanced Tab

Use this tab to enable tracing. For more information, see *TIBCO Administrator™ User Guide* and Enabling Tracing for all TIBCO ActiveMatrix BusinessWorks Tasks.

Step 3: Configuring Smart Routing

Note Configuring smart routing is an optional step.

Note Public smart routing for TIBCO PartnerExpress™ and TIBCO BusinessConnect™ Plug-in for FTP Server is not supported and would always go to the default cluster.

Public smart routing is used to better distribute the workloads and alleviate the likelihood of bottlenecks while receiving inbound documents. Multiple clusters can be introduced to handle a variety of workloads separately. A rule-based routing mechanism based on a combination of configurable conditions and predefined set of criteria is used to make decisions so that it can dispatch the workloads to the best fitting cluster for processing.

In order to configure public transports and use public smart routing, you need to configure clusters of machines based on the transport type, define conditions (rules) for these clusters, and then map the clusters into groups. For more information about rule based routing, see *TIBCO BusinessConnect Concepts*, Public Smart Routing.

Configuring Smart Routing Rules

Note When a rule is changed, the Deployment page will still show as synchronized but the changes to the public smart routing rules will take effect only after both the Interior Server and Gateway Server are restarted.

To enable or disable an inbound transport for the installation, do the following:

1. Expand **Application Management > BusinessConnect > Configuration.**
2. Click the **BusinessConnect** link in the Configuration Builder panel.
3. Select the **Public Process Configuration** tab.

4. Click **Add.**
5. Select one of the transports from the list:
   - AS1_EMAIL
   - AS2_HTTP
   - AS2_HTTPS
   - EMAIL
   - FILE
   - FTP
   - FTPS
   - HTTP
   - HTTPS
   - HTTPSCA
   - SSHFTP
6. Click **OK**.

   The dialog New Rule with the selected transport type appears:
7. Enter data following the explanations in New Cluster Configuration.

New Cluster Configuration

Fields	Enter
Cluster Name	Enter the name of the cluster. This is a logical name of the location where messages are routed. The cluster name must begin with an alphanumeric character and be followed by zero or more alphanumeric characters such as '_' (underscore), '-' (hyphen) or '.' (dot); for example, `CLUSTER_LARGE_MESSAGES, BC_CLUSTER_03, SERVER-POOL-19, C001` The value is *not* case sensitive
Transport Type	Pre-populated with the name of the transport you have selected.
Rule	Expression for the rule is populated from the selection made in the

Fields	Enter
Expression	added conditions for the attribute, operator and operand.
Enabled	Enable or disable the routing mechanism by selecting or clearing this check box.
Add New Condition	Each time you click this button, a new row of attributes will be added. Condition Type can be set to • if all conditions are met: more restrictive rule • if any conditions are met: less restrictive rule A new condition is now displayed, with the configurable options Attribute, Operator, and Operand. For more information about these options, see *TIBCO BusinessConnect™ Concepts*, Attributes, Operators, and Operands.

8. Enter data as explained in the following tables for the respective transports:
   — Rule Options for the HTTP/S, HTTPCA, and AS2_HTTP/S Transports
   — Rule Options for the FTP/S and SSHFTP Transports
   — Rule Options for the File Transport
   — Rule Options for the Email and AS1_Email Transports

The defined rules will be displayed in the field Rule Expression.

⚠ **Warning**	Cluster names are not unique: multiple rules can be assigned to the same cluster name.

Rule Options for the HTTP/S, HTTPCA, and AS2_HTTP/S Transports

Rule Options for the HTTP/S, HTTPCA, and AS2_HTTP/S Transports

Attribute	Operator	Operand1, Operand2	Explanation
HTTP_Host	`matches` `=`	(host name)	Enter the host name
HTTP_Version	`matches` `=`	(HTTP version)	Define whether to use a certain HTTP version.
Large_Content	`is`	`false` `true`	Define whether the file size will be large (`true` or `false`)
Query_String	`matches` `=`	(query)	Define whether to use a certain query.
Request_URI	`matches` `=`	(URI)	Define whether to use a certain URI.
Secure_SSL	`is`	`false` `true`	Define whether the transport will be secure (`true` or `false`)
Client_Auth	`is`	`false` `true`	Define whether client authentication is `true` or `false`

Content_Size	`=`
	`greater_than`
	`less_than`
	`range`
		(value)	Define whether the file size will be equal to, bigger, smaller, or in the range of a certain value.

AS2_From (AS2 only)	`matches` `=`	(partner name)	Enter the AS2_ID of the partner sending the message
AS2_To (AS2 only)	`matches` `=`	(partner name)	Enter the AS2_ID of the partner receiving the message

Defining Rules Using Regular Expressions

The supplied examples refer to TIBCO BusinessConnect™ Services Plug-in (EZComm).

- **`Request_URI matches "/EZComm":`** using the HTTP rule type, create a rule for the Request_URI attribute, use the "matches" operator, specify "/EZComm" for the value.
- **`Query_String matches ".*fromTp=Partner.*"`**: using the HTTP rule type, create a rule for the Query_String attribute, use the "matches" operator, specify ".*fromTp=Partner.*" for the value
- **`Query_String matches ".*operationID=BC%2F1\.0%2FNotify.*"`**: using the HTTP rule type, create a rule for the Query_String attribute, use the "matches" operator, specify ".*operationID=BC%2F1\.0%2FNotify.*" for the value

Rule Options for the FTP/S and SSHFTP Transports

Rule Options for the FTP/S and SSHFTP Transports

Attribute	Operator	Operand1, Operand2	Explanation
File_Name	`matches` `=`	(file name)	Define the name of the file to be sent. The whole path with the file name must be specified.

File_Size	`=`
	`greater_than`
	`less_than`
	`range`
		(value)	Define whether the file size will be equal to, bigger, smaller, or in the range of a certain value.

From_Partner	`matches` `=`	(partner name)	Enter the name of the partner sending the message
To_Partner	`matches` `=`	(partner name)	Enter the name of the partner receiving the message
Large_File	`is`	`false` `true`	Define whether the file size will be large (`true` or `false`)
Protocol	`matches` `=`	(protocol name)	Define whether to use a certain protocol.

Defining Rules Using Regular Expressions

The supplied example refer to TIBCO BusinessConnect Services Plug-in (EZComm).

- **`(Protocol matches "EZComm") and (From_Partner matches "MyPartner")`**: using the SSHFTP rule type, create a rule for the Protocol attribute, use the "matches" operator, specify "/EZComm" for the value, add a new condition for the From_ Partner attribute, use the "matches" operator, specify "MyPartner" for the value

Rule Options for the File Transport

Rule Options for the File Transport

Attribute	Operator	Operand1, Operand2	Explanation
File_Name	`matches` `=`	(file name)	Enter the full path for the file name

File_Size	`=`
	`greater_than`
	`less_than`
	`range`
		(value)	Define whether the file size will be equal to, bigger, smaller, or in the range of a certain value.

Large_File	`is`	`false` `true`	Define whether the file size will be large (`true` or `false`)
Protocol	`matches` `=`	(protocol name)	Define whether to use a certain protocol.

Rule Options for the Email and AS1_Email Transports

Rule Options for the Email and AS1_Email Transports

Attribute	Operator	Operand1, Operand2	Explanation
Sender	`matches` `=`	(host email	Enter the email of the host sending

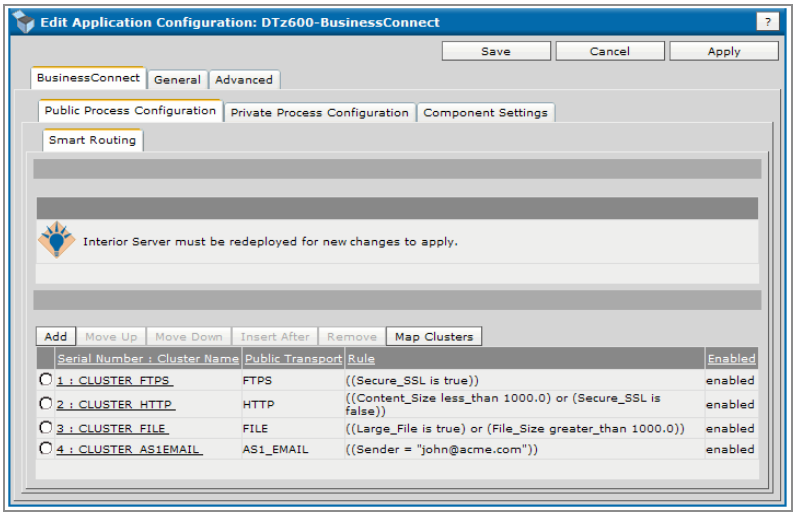
Attribute	Operator	Operand1, Operand2	Explanation
		address)	the message.
Subject	`matches` `=`	(email transport subject)	Define whether to use a certain subject.

Content_Size	`=`
	`greater_than`
	`less_than`
	`range`
		(value)	Define whether the file size will be equal to, bigger, smaller, or in the range of a certain value.

Large_ Content	`is`	`false` `true`	Define whether the file size will be large.
Recipient	`matches` `=`	(partner's email address)	Enter the email address of the partner receiving the message.

9. Add more rules for clusters with defined conditions, as necessary. All added clusters and rules will be listed, together with their corresponding public transports and the routing rules, as in Listed Clusters.

Figure 2: Listed Clusters



The numbers in front of the rules represent their precedence. Once the rules are added and the routing conditions defined, you can group (map) them into fault tolerant groups using the selected criteria.

Map Clusters

Clusters are mapped using the configured rules into fault tolerance groups. The Map Clusters window allows you to list the configured clusters by:

- Load Balanced Cluster Name (Rule Id), as explained in New Cluster Configuration.
- Assigned Fault Tolerance Group, as explained in Fault Tolerance Tab
- Service Instances, as explained in Fault Tolerance Tab

Cluster maps can be edited as follows:

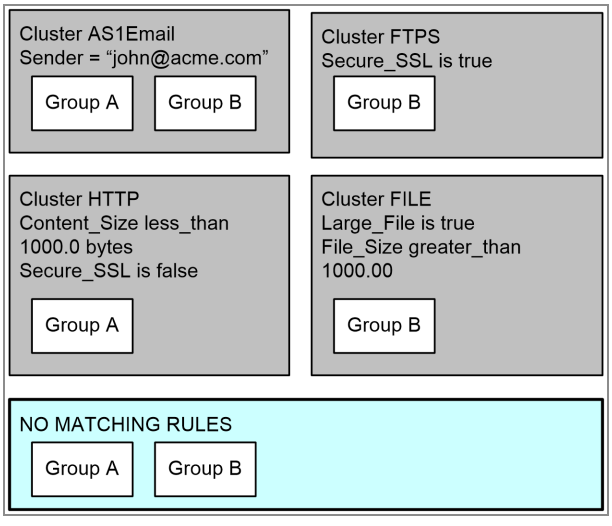
1. Click **Map Clusters**.

   You will notice that, in addition to all the clusters you have defined, there is one additional cluster called NO MATCHING RULES. By default, all service instances are added to this cluster and later can be assigned to another cluster.

Note

The TIBCO BusinessConnect configuration must be saved each time a new server is added. If the configuration is not saved properly, the default cluster will not start and the engines will not be assigned to process unmatched rules (processed otherwise by the NO MATCHING RULES cluster).

Figure 3: Map Clusters Dialog



To read more about this cluster, see *TIBCO BusinessConnect™ Concepts*, NO MATCHING RULES.

2. From the `Group by` list, select the criteria by which you want to group the server clusters:
   - **Load Balanced Cluster Name (Rule Id)** Clusters are grouped by the cluster name defined for load balancing
   - **Assigned Fault Tolerant Group** Clusters are grouped in the previously assigned fault tolerant groups
   - **Service Instance** Clusters are grouped by the existing service instances
   - **None** If selected, all load balanced clusters, fault tolerant groups, and service instances are listed without grouping.

   Regardless of the grouping on the screen, you will be able to add the previously assigned fault tolerant groups of servers to any of the clusters.

Assigning a Fault Tolerance Group to a Cluster

Note After you import a `.csx` file that contains the configured public Smart Routing rules, the clusters are still not mapped to any of the fault tolerant groups. You must add again the fault tolerant group to the cluster.

To assign a fault tolerant group to a cluster (define rules for this group):

3. In the Map Clusters window, click the link c*luster_name*.

   The Edit Cluster Map dialog appears.

4. Check the check boxes next to the fault tolerant group(s) that you want to add to the cluster map.
5. Click **Add to Cluster Map**.

   The fault tolerant groups to which you assigned the specified cluster map are now listed under Assigned Fault Tolerant Groups.

   You can remove any group(s) that you want by checking their check boxes and clicking **Remove From Cluster Map**.

6. Click **Save**.

   This will list the fault tolerant groups you have assigned under the appropriate cluster.

Clusters have been mapped to two fault tolerant groups (*domain*.BCFTGROUP.A and *domain*.BCFTGROUP.B), and also to two different service instances (*machine1* `- Interior Server` and *machine2* `- Interior Server`).

7. View the assigned clusters by the assigned fault tolerant groups (Assigned Fault Tolerant Groups), by the load balanced cluster name (Load Balanced Cluster Name), or by service instance

Figure 4: Assigned Fault Tolerant Groups

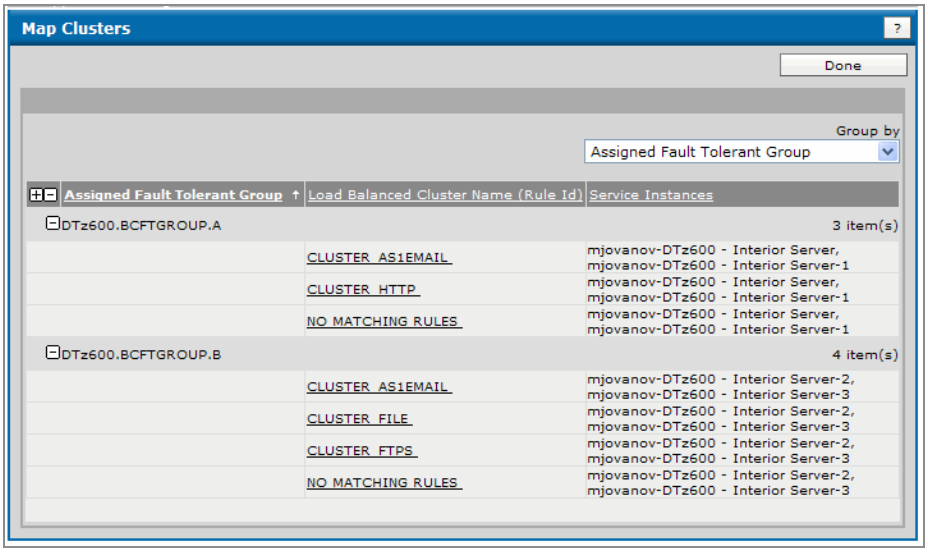


Figure 5: Load Balanced Cluster Name

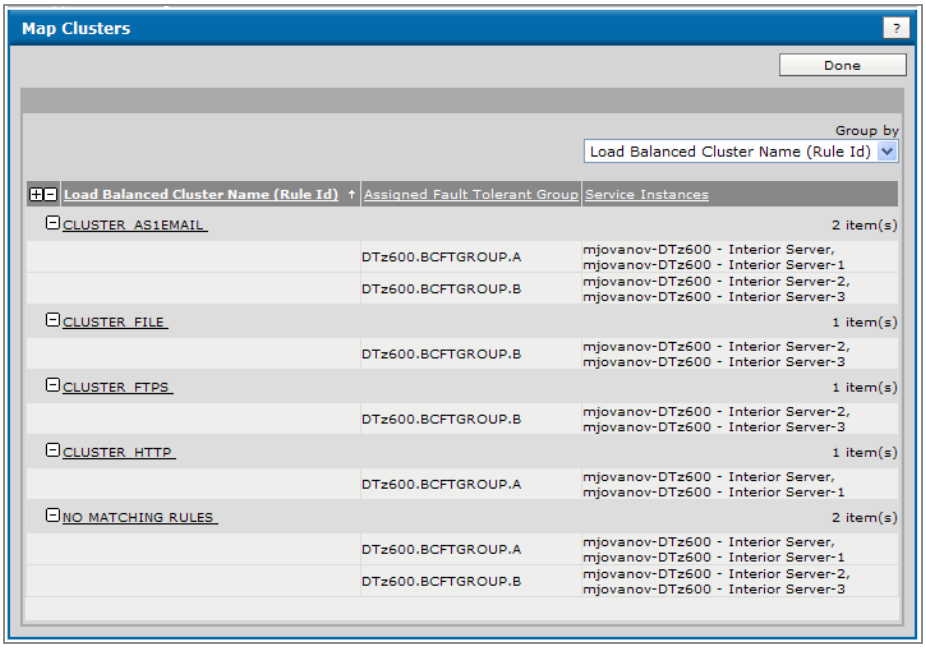
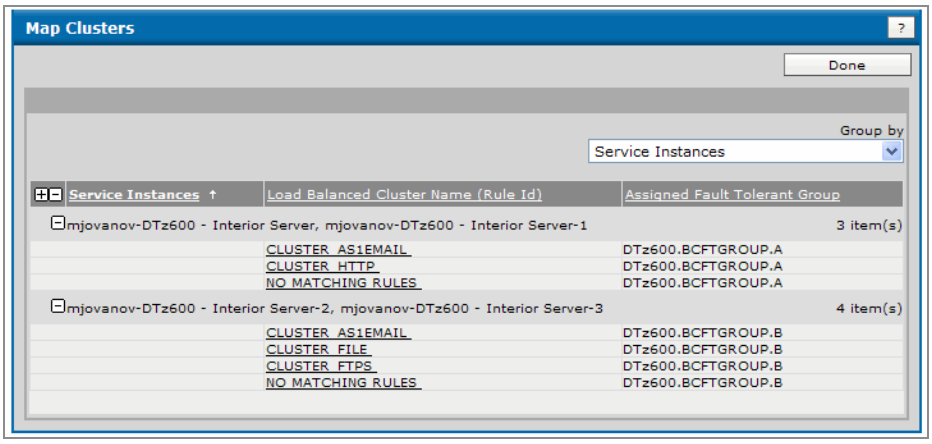


Figure 6: Service Instance



Notice that the cluster NO MATCHING RULES appears in both fault tolerant groups and on both service instances. Notice also the cluster HTTP appears in two fault tolerant groups and on two service instances. This means that ClusterFTPS, AS1Email, and File will be processed only by one of the fault tolerant groups and on one of the instances, while the cluster HTTP will be processed on both the group A and B and on both instances of the Interior Server.

Private Process Configuration

This section explains how to configure Private Processes for TIBCO BusinessConnect, private transports (TIBCO Rendezvous and JMS), and Outbound File pollers.

You can perform the configuration steps explained in this section only after you have created a deployment configuration, as explained in Deployment Configuration.

Private Process Communication

Three ways of transmission are used for communicating between private processes and the TIBCO BusinessConnect server:

- TIBCO Rendezvous and JMS Messages
- Outbound File Pollers

TIBCO Rendezvous and JMS Messages

Private process messages are TIBCO Rendezvous or JMS messages that travel between a private process and the TIBCO BusinessConnect instance.

The private process creates a private process message when it receives a message from an internal application such as SAP. The following series of events then usually occurs:

1. The private process message is sent from the private process to the TIBCO BusinessConnect server, which converts the private message into a public message and sends it over the Internet to a trading partner.
2. The trading partner's TIBCO BusinessConnect server in turn re-converts the public message into a private message and forwards to its private process, which forwards it to its internal application.

To learn more about private processes and TIBCO BusinessConnect architecture, see *TIBCO BusinessConnect™ Concepts*, TIBCO BusinessConnect Architecture.

To learn more about Rendezvous messaging, refer to the TIBCO Rendezvous Certified Messaging documentation.

Outbound File Pollers

Outbound File pollers provides a simple way for private processes to transmit documents to TIBCO BusinessConnect. This contrasts with the other transports, which are used for communication between trading partners.

The outbound File pollers are used by enterprises that do not wish to use Rendezvous or JMS to transfer documents to TIBCO BusinessConnect.

For more information, see *TIBCO BusinessConnect™ Trading Partner Administration*.

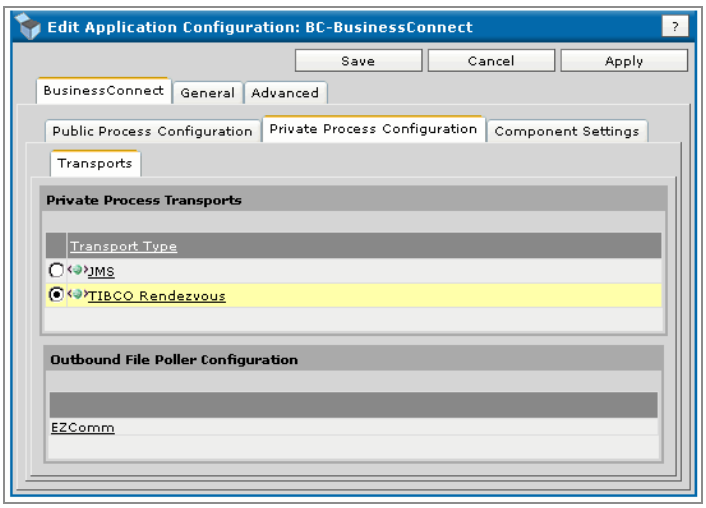
Selecting the Private Process Transport

Before TIBCO BusinessConnect can transport messages between the TIBCO BusinessConnect server and a private process using either TIBCO Rendezvous or JMS, you need to enable one of these two transports and to provide basic configuration information as follows:

1. Expand **Application Management > BusinessConnect > Configuration**.
2. Click the **BusinessConnect** link in the Configuration Builder panel.
3. Click the **Private Process Configuration** tab.

   The Edit Application Configuration dialog appears.

Figure 7: Editing Application Configuration



The Private Process Configuration tab offers an additional Transports tab , which allows you to manage the following:

- JMS Transport
- TIBCO Rendezvous
- Outbound File Pollers

JMS Transport

Note To learn more about the JMS transport, see *TIBCO Enterprise Message Service™ User Guide*, Using the SSL Protocol.

When JMS is selected, TIBCO BusinessConnect will assume that the runtime communication with the private process will take place over a secured or unsecured JMS connection. JMS can only be selected (and saved successfully) if the specified connection factory uses the TIBCO Enterprise Message Service factory string: `com.tibco.tibjms.naming.TibjmsInitialContextFactory`. Otherwise, the transport can only be saved if this button is not selected.

Note BusinessConnect sends some messages on JMS Topics, such as ERROR advisories. You can configure JMS Durable topics to avoid message loss.

Configuring JMS Transport

To configure JMS settings, do the following:

1. Enter information specified in JMS.

JMS

Field	Enter
JMS Settings	
Protocol Prefix	`jms://`

Field	Enter
JMS User Name	Username to use when logging into the JMS server. If the JMS provider does not require access control, this field can be empty. Not all JMS servers require user names and passwords. Refer to your JMS provider documentation and consult your system administrator to determine if your JMS server requires a username and password.
JMS Password	Password to use when logging into the JMS server. If the JMS provider does not require access control, this field can be empty.
JNDI Context Factory	The initial context factory class for accessing JNDI. (`javax.naming.Context.INITIAL_CONTEXT_FACTORY`). **Note**: TIBCO BusinessConnect attempts to find the class. However, you may need to add the Java file supplied by your JNDI service provider to the `CLASSPATH` environment variable to use JNDI.
JNDI Context URL	This is the URL to the JNDI service provider (javax.naming.Context.PROVIDER_URL). See your JNDI provider documentation for the syntax of the URL.
JNDI User Name	Username to use when logging into the JNDI server (`javax.naming.Context.SECURITY_PRINCIPAL`). If the JNDI provider does not require access control, this field can be empty.
JNDI Password	Password to use when logging into the JNDI server (`javax.naming.Context.SECURITY_CREDENTIALS`). If the JNDI provider does not require access control, this field can be empty.
Topic Connection Factory	The TopicConnectionFactory object stored in JNDI. This object is used to create a topic connection with a JMS application.

Field	Enter
	See your JNDI provider documentation for more information about creating and storing TopicConnectionFactory objects.
Queue Connection Factory	The QueueConnectionFactory object stored in JNDI. This object is used to create a queue connection with a JMS application. See your JNDI provider documentation for more information about creating and storing QueueConnectionFactory objects.
Reconnect Max. Duration (mins)	This is the time during which the TIBCO BusinessConnect server will try to reconnect. After this time, there will be no attempt to reconnect. This duration time does not represent the reconnection frequency. Default is 10 minutes.
Secured	If selected, the transaction will be secured.
Verify JMS Server	If selected, the JMS server's identity (that is, its X509 certificate as well as the specified value in the “Expected JMS Server Host Name” field) will be verified against the data received during the SSL handshake. If either the trusted CA certificates or the expected hostname doesn't match, the transport creation fails. If this verification is not required, BC can establish a JMS connection with any TIBCO Enterprise Message Service, whose credentials are different from the configured properties.
JMS Server Certificate	The certificate credential of the JMS server. To create this certificate, follow the steps described in *TIBCO BusinessConnect Trading Partner Administration*, Adding LDAP/JMS/Email Server Certificates The credential is stored in the TIBCO BusinessConnect keystore and is expected to be configured on the TIBCO Enterprise Message Service server according to the corresponding guidelines.
Expected JMS Server Host Name	The value of the common name component of the TIBCO Enterprise Message Service server's leaf certificate. This is usually the hostname

Field	Enter
	of the resource, running the TIBCO Enterprise Message Service server. If it is a test system, the common name (CN) value may be any arbitrary string, which must match the value of this field if the “Verify JMS Server” checkbox is checked.

Strong Ciphers Only	If the box is checked, only strong encryption algorithms will be used between the server (or the palette) and the JMS provider. The below cipher suites are offered by the connecting client (either BusinessConnect or the palette) in this mode:
	`TLS_RSA_WITH_AES_256_CBC_SHA`
	`TLS_RSA_WITH_AES_128_CBC_SHA`
	`TLS_DHE_RSA_WITH_AES_256_CBC_SHA`
	`TLS_DHE_RSA_WITH_AES_128_CBC_SHA`
	`SSL_RSA_WITH_RC4_128_SHA`
	`SSL_RSA_WITH_3DES_EDE_CBC_SHA`
	`SSL_DHE_RSA_WITH_3DES_EDE_CBC_SHA`
	`SSL_DHE_DSS_WITH_3DES_EDE_CBC_SHA`
	`TLS_DHE_DSS_WITH_AES_128_CBC_SHA`
	`TLS_DHE_DSS_WITH_AES_256_CBC_SHA`
	Note: The unlimited strength JCE jurisdiction policy files are pre-

Field	Enter
	installed on the TIBCO Java Runtime Environment (JRE).
Use Trace	See comments in *TIBCO ActiveMatrix BusinessWorks Palette Reference*, JMS Palette section Advanced. When this option is used, the SSL-specific debug tracing for the secure JMS transport will be sent to the engine standard output only.
Queues/Topics (Settings in this section are not configurable)	
Initiator Request Queue	*installation_prefix*.*installation_name*.INITIATOR.REQUEST This is the queue on which TIBCO BusinessConnect listens for requests from the Private Process to initiate requests to its partners. *installation_prefix* and *installation_name* are configured under **BusinessConnect > System Settings > General** tab.
Initiator Response Queue	*installation_prefix*.*installation_name*.INITIATOR.RESPONSE This is the destination (queue) on which the Private Process listens for a response from the Initiator's TIBCO BusinessConnect server. **Note**: If a user configures the private process to receive an INITIATOR.RESPONSE message from the TIBCO BusinessConnect server synchronously over the JMS transport, this activity will be successfully completed as soon as the expected response is received. If the TIBCO BusinessConnect server then resends the same INITIATOR.RESPONSE message for any reason, such as when the user triggers the resend, such message will no longer be requested by the private process and may need to be manually removed. Leaving these messages in the queue won't cause any problems unless such retries occur in large numbers: if they do, the user should consider manual maintenance of the JMS queue for the INITIATOR.RESPONSE messages.
Responder Request Queue	*installation_prefix*.*installation_name*.RESPONDER.REQUEST This is the destination (queue) on which the Responder TIBCO BusinessConnect server sends requests to the Private Process.

Field	Enter
Responder Response Queue	*installation_prefix*.*installation_name*.RESPONDER.RESPONSE This is the destination (queue) on which the Responder TIBCO BusinessConnect server listens to responses from the Private Process.
Responder Acknowledgement Confirm Queue	This is the queue on which Responder TIBCO BusinessConnect sends acknowledgement messages to private processes. Example: *installation_prefix*.*installation_name*.RESPONDER.ACK

2. Once you have entered all required data in the section JMS Settings, click **Test Connection** to verify that the connection works.
3. If necessary, change and verify the data again.

JMS Auto Reconnect for the TIBCO BusinessConnect Server

If the JMS server is down or the network connection is down when TIBCO BusinessConnect engine starts up, the engine will try to reconnect to the JMS server for a specified period of time (as set in the field Reconnect Max Duration). If the connection could not be established within this time, the engine will stop.

However, if the connection between the engine and the JMS server is established within the specified period of time (as set in the field Reconnect Max Duration), the engine will continue to execute.

If the connection between the TIBCO BusinessConnect engine and the JMS server is terminated during runtime, the engine will try to establish connection. During this time, messages from private process to TIBCO BusinessConnect will not be received. If the protocols are trying to send message to private process, the engine will hold the message for a specified period of time (as set in the field Reconnect Max Duration), configured in the JMS transport, to check if the connection is established. If the connection is established then the message is sent to private process. If the connection is not established with this period of time the engine will throw an appropriate error message.

TIBCO Rendezvous

When using TIBCO Rendezvous, messaging is performed through RVCM (TIBCO Rendezvous Certified Messaging) using the TIBCO BusinessConnect `aeRvMsg` standard message format (see *TIBCO BusinessConnect™ Concepts*, aeRvMsg Message Format).

The `request` or `response` field in a message that a private process sends or receives contains an unparsed message.

An unparsed XML string has a single message attribute and looks like this:

```
<?xml version="1.0" encoding="UTF-8"?>...rest of document in XML.
```

An unparsed non-XML message can have any form. For example, in an `aeRvMsg` *request* message, the `^data^` tag contains the following fields:

- `transactionID` Describes the transaction ID.
- `operationID` Describes the operation ID.
- `request` Describes the request itself. This is usually an XML string representing the message body or a TIBCO Rendezvous representation of an XML file. This is the original message that was first generated by the private process when it was contacted by the Initiator's application.
- `closure` Describes a closure object to be copied from the private process to the response.

Configuring TIBCO Rendezvous

To configure TIBCO Rendezvous settings for private processes, do the following:

1. Enter information specified in TIBCO Rendezvous Settings.

TIBCO Rendezvous Settings

Field	Enter
TIBCO Rendezvous Settings	

Field	Enter
Service	Name of the service
Network	Network on which the service is running
Daemon Host	Host used by the RV daemon
Daemon Port	Port used by the RV daemon
BusinessConnect to Private Process: RVCM Settings	
CM Application Name Prefix	BC-*domain_name*.`PP_CM` (pre-populated)
Ledger File Location	Enter the location of the ledger file.
CM Ledger File Name Postfix	`PP_CM.ldg` (pre-populated
Initiator/Responder Listeners (Specify a comma separated list of listeners)	Enter listeners separated by a comma.
Private Process to BusinessConnect: RVCMQ Settings	
CMQ Name (pre-populated)	BC-*domain_name*.pp-cmq
Scheduler Heartbeat (seconds)	The active scheduler sends heartbeat messages at the interval you specify (in seconds). Heartbeat messages inform other members that a member is acting as the scheduler. All members of a group must specify the same scheduler heartbeat interval. Default is 5. See *TIBCO Rendezvous Concepts*, Scheduler Parameters for more details.

Field	Enter
Scheduler Activation (seconds)	All members of a group must specify the same scheduler activation interval. When the heartbeat signal from the scheduler has been silent for this interval (in seconds), the worker with the greatest scheduler weight takes its place as the new scheduler. Default is 15. See *TIBCO Rendezvous Concepts*, Scheduler Parameters for more details.
Subjects (Settings in this section are not configurable)	
Initiator Request Subject	Subject of the TIBCO Rendezvous request from the Initiator's private process to the local TIBCO BusinessConnect. *installation_prefix*.*installation_name*.*standardID*.INITIATOR.REQUEST *installation_prefix* and *installation_name* are configured under **BusinessConnect > System Settings > General** tab. The standard ID is the protocol ID, such as EZComm, SOAP, and so on.
Initiator Response Subject	Subject of the TIBCO Rendezvous response from the Initiator's TIBCO BusinessConnect to the local private process. *installation_prefix*.*installation_name*.*standardID*.INITIATOR.RESPONSE
Responder Request Subject	Subject of the TIBCO Rendezvous request from the Responder's TIBCO BusinessConnect to the local private process. *installation_prefix*.*installation_name*.*standardID*.RESPONDER.REQUEST
Responder Response Subject	Subject of the TIBCO Rendezvous response from the Responder's private process to the local TIBCO BusinessConnect. *installation_prefix*.*installation_name*.*standardID*.RESPONDER.RESPONSE

2. Once you have entered and viewed all required data, click **Save**.

Intercomponent Communication Rendezvous Settings

Intercomponent Communication Rendezvous settings are used within the Interior Server for its various sub-components, such as process starters, poller triggers, and intracomponent communications.

Settings are described in:

- Gateway Engine
- Interior Server
- TIBCO Administrator to Interior Server Peer Update Settings

Gateway Engine

Communication between the Gateway Engine and Interior Server is configured at the **BusinessConnect > Gateway > Gateway Tokens** tab in the Advanced section, where each file carries the intercommunication JMS.

Interior Server

If there are differences in the network part of the network parameter between TIBCO Administrator and the Interior Server machines, the following properties must be added along with multicast group to each of the Interior Server engine `.tra` files.

The values entered should follow the same rules used for creating the Rendezvous transport. This is used when other than the default is being set for the network part, and an appropriate value is set for the network parameter.

Following properties must be added and configured appropriately for the Interior Server Engine .tra files.

```
tibco.clientVar.gatewayProperties/transports/Intercomponent/msh/network=<nic|hostname|ip>;<multicast group>
tibco.clientVar.gatewayProperties/transports/Intercomponent/msh/service=<value>
tibco.clientVar.gatewayProperties/transports/Intercomponent/msh/daemon=<value>
tibco.clientVar.gatewayProperties/transports/Intercomponent/bmh/network=<nic|hostname|ip>;<multicast group>
```

```
tibco.clientVar.gatewayProperties/transports/Intercomponent/bmh/service=<
value>
tibco.clientVar.gatewayProperties/transports/Intercomponent/bmh/daemon=<v
alue>
tibco.clientVar.gatewayProperties/transports/Intercomponent/dmz/network=<
nic|hostname|ip>;<multicast group>
tibco.clientVar.gatewayProperties/transports/Intercomponent/dmz/service=<
value>
tibco.clientVar.gatewayProperties/transports/Intercomponent/dmz/daemon=<v
alue>
```

TIBCO Administrator to Interior Server Peer Update Settings

TIBCO Administrator is not required to run on the same physical machine as the Interior Server and, in certain other usage scenarios, TIBCO Administrator could be running on a separate subnet not related to the subnets in which the Interior Servers are running (this includes TIBCO BusinessConnect Gateway and TIBCO BusinessConnect Interior Servers).

The Intercomponent JMS settings in the deployment UI will be used by TIBCO Administrator machine to send peer change to the TIBCO BusinessConnect Interior Server engines whenever there are changes made in the Administrator UI.

TIBCO Hawk, fault tolerant or default TIBCO Rendezvous parameters might need valid values set for the TIBCO BusinessConnect Server to start. Below are the parameters:

- `Bus.User.<rv parameters>`
- `Hawk.<rv parameters>`
- `Bus.FtDefault.<rv parameters>`
- Bus.Default.<rv parameters>

Outbound File Pollers

Outbound File pollers are protocol specific. This section provides global information on their configuration, while the specific information is explained for each of the protocols.

Note Directories for Inbound and Outbound File pollers should not be the same ones that are used for storing shared or local files.

Note By default, the Outbound File poller will pick up existing files when the engine starts up.

Enabling and Configuring Outbound File Poller

To enable an Outbound File poller, perform these steps:

1. In TIBCO Administrator, expand **Application Management > BusinessConnect > Configuration**.
2. Click the **BusinessConnect** link in the Configuration Builder panel.
3. Select the **Private Process Configuration** tab.
4. Click the protocol link in the Outbound File Poller Configuration area.
5. Edit the options listed in Table 43.

Outbound File Poller Configuration

Field	Description
Enable	Enable to Outbound File poller.
Directory to Monitor (e.g. /.../*.*)	The name of the file `fileName` and directory location, if desired, to monitor. Either provide the `fileName` or use the asterisk (*.*) character as a wildcard to specify a collection of files. Do not provide a directory location only. TIBCO BusinessConnect searches subdirectories recursively. The directory C:\ will not be taken as a base directory: specify `C:\`*BaseDir* instead. A better configuration is `C:\`*BaseDir*`\*.*`, which specifies the directory for the Outbound File poller. **Note**: Directories for Inbound and Outbound File pollers should not be the same ones that are used for storing shared or local files.
Directory to Place Error Files	Designate a directory where the files will be placed if an error occurs during the processing of the outgoing files.

Field	Description
Polling Interval (seconds)	How often polling occurs. Default is 300.
Delete File	Enable files to be deleted after processing. In order to avoid that the same files are picked up on the subsequent startup of the TIBCO BusinessConnect engine, it is advised to select this checkbox and have the files removed after processing has been completed.

Interior Server Deployment

This section explains how to deploy the Interior Server.

Before You Deploy

⚠ **Warning**	Before you deploy the installed TIBCO BusinessConnect application, verify that TIBCO Hawk agents are running on all target machines.

⚠ **Warning**	If you deploy multiple TIBCO BusinessConnect components, all must be on the same platform — either Windows or UNIX. Also, verify that the machines on which you plan to deploy belong to the TIBCO Administrator domain. If they don't, use TIBCO Domain Utility to add the machines to the domain. Read about the Domain Utility in *TIBCO Runtime Agent Domain Utility User Guide*. The domain *must* be configured to use UTF-8 encoding.

⚠ **Warning**	Never disable `BCBootstrap.serviceagent`. If it is disabled, no TIBCO BusinessConnect component will work at runtime.

When deploying the TIBCO BusinessConnect application, keep in mind that the ledger files will be affected differently depending on their location:

- If you use the default ledger file location and you redeploy TIBCO BusinessConnect after undeploying it, the ledger file(s) will be removed and recreated again (which is the same as losing any undelivered messages).

- If you do *not* use the default ledger file location, the ledgers will be left as is assuming that the location has not changed.

The ledger file location can be set using the private process transport settings, by expanding **Application Management > BusinessConnect > Configuration > BusinessConnect** link **> Private Process Configuration > TIBCO Rendezvous > BusinessConnect to Private Process:RVCM Settings: Ledger File Location**.

Deploying and Starting the Interior Server

Warning

When you are deploying TIBCO BusinessConnect you should not use a terminal or command window and change directory to either of these directories: *TIBCO_HOME*`\tra\domain\`*domain_name*`\application\BusinessConnect` and *TIBCO_HOME*`\tra\domain\`*domain_name*`\datafiles`.

Doing so would cause deployment to fail. Also, you should not try to modify any files under these directories. If any of the files were accidentally open, close these files and use the option **Force redeploy**.

1. Expand **Application Management > BusinessConnect > Configuration**.

   The Interior Server.par must have the deployability status of `Deployable, (new)`
   and TIBCO BusinessConnect must have the deployability status of `Deployable, services require deployment.`

2. Click **Deploy**.

   The Deploy Configuration screen appears with several options that you can configure. Deploy Configuration Options shows the list of configurable options on the screen.

Deploy Configuration Options

Option	Description
Stop running services before	Select Stop running services before deployment to stop all running services before deploying the service. All services that should be

Option	Description
deployment	redeployed are stopped.
Kill services that haven't stopped after (seconds)	Indicate how many seconds can elapse before a service is stopped, using the Stop command.
Start successfully deployed services	Select Start successfully deployed services to stop and restart the services in the application after they have been successfully deployed. If you do not select this option, you can explicitly start the services later.
Force redeployment of all services	Select Force redeployment of all services to redeploy all services even if a service is in a synchronized state.
Description	Describe the deployment configuration (optional)
BusinessConnect	
Administrator Tasks To Perform	Lists the tasks that TIBCO Administrator will perform for this server if you choose to deploy by selecting **OK**.
Remote Tasks To Perform	Lists the tasks to perform on the selected machine (which could actually be the local machine) in the following fields: • **Software** — Required software for this application (for example, an adapter or TIBCO ActiveMatrix BusinessWorks). • **Deployability** — Shows whether the application is deployable and whether it's been deployed before. • **Machine** — Computer on which the application is scheduled to be deployed. • **Machine Status** — Machine state. • **Tasks**— The actions that the deployment process will perform on the target machine(s).

Option	Description
Service Instance	Name of the service instance (*machine_name*-`Interior Server`)
Service Configuration	`Interior Server.par`

Note Although the TIBCO BusinessConnect instance can be started from the command line, starting from TIBCO Administrator is the *recommended* method.

3. Click **OK**.

   If you leave the `Start successfully deployed services` checkbox selected, the TIBCO BusinessConnect engine will be deployed and then started.

4. After a successful deployment, Deployability is now `Synchronized` and Deployment Status is `Success`, indicating that the instance is now deployed.
5. After all engines are deployed, they will be started if the Start option was selected.

Manage the Interior Server

This section explains how to manage the Interior Server.

Editing Application Configuration

To edit an application's configuration before or after deployment, do the following:

1. Expand **Application Management > BusinessConnect > Configuration.**
2. Click the **BusinessConnect** link in the Configuration Builder panel.

   The Edit Application Configuration dialog contains the following tabs:

   — BusinessConnect Tab

   — General Tab

   — Advanced Tab

Note When BusinessConnect is in **Deployable (Configuration Update)** state, `BusinessConnect application is in Requires-Deployment state` warning message is displayed which indicates that you can save changes under the deployment configuration.

BusinessConnect Tab

The BusinessConnect tab contains these additional sub-tabs:

- Public Process Configuration tab
- Private Process Configuration Tab
- Component Settings Tab

Public Process Configuration tab

Using this tab, you can configure public smart routing.

For more information, see Step 3: Configuring Smart Routing .

Private Process Configuration Tab

Using this tab, you can select the transport type for communications between the TIBCO BusinessConnect server and private processes: TIBCO Rendezvous or JMS. Using this tab you can also configure the outbound File poller.

For more information, see Private Process Configuration.

Component Settings Tab

Using this tab, you can configure TIBCO BusinessConnect components and configure some other advanced features, such as shared and large file locations.

From the Component Settings tab, you can open the following links:

- Intercomponent Communication
- Intercomponent Advanced

General Tab

This tab displays information about:

- **Application archive** Package Name, Package Version, Package Description, Package Creation Date, and Package Owner.
- **Application Parameters** Name, Deployment Name, Description, Contact, Max Deployment Revision.

You can also use this tab to upload a new EAR archive file that contains a TIBCO ActiveMatrix BusinessWorks project.

- Click **Upload New EAR File**, and browse to the file location, and select it to upload.

Advanced Tab

This tab shows TIBCO BusinessWorks and Adapters Deployment Repository Instance.

TIBCO BusinessWorks and Adapters Deployment Repository Instance includes the following information:

- **Transport** Transport is always `local`. No changes are required.
- **Message Encoding** Select UTF-8 or ISO.

⚠ **Warning**	The Warning on this screen reminds you about the required version of the TRA agents on the target machines: “Require TRA 5.3 (or higher) Agents on all target machines.”

Tracing

This section describes how to enable tracing and view tracing output in TIBCO Administrator, the TIBCO BusinessConnect engine, and the TIBCO BusinessConnect palette.

Tracing for TIBCO Administrator

Enabling Tracing

To enable tracing, access the file *TIBCO_HOME*`\tra\domain\`*domain_name*`\AdministrationDomain.properties` and set the property `LogDebug=true`.

Viewing the Log

View tracing output in the log *TIBCO_HOME*`\tra\domain\`*domain_name*`\logs\Administrator.log`

Tracing for the TIBCO BusinessConnect Engine

Enabling Tracing for all TIBCO ActiveMatrix BusinessWorks Tasks

Note Tracing affects the performance of a running TIBCO BusinessConnect engine.

To enable tracing for all TIBCO ActiveMatrix BusinessWorks tasks:

1. Locate the `BusinessConnect-Interior_Server.tra` file in the *TIBCO_HOME*`\tra\domain\`*domain_name*`\application\BusinessConnect` directory.
2. Enable the task by editing the file as follows:

   `Trace.Task.*=True`.
3. Click **Save**.
4. Deploy TIBCO BusinessConnect to apply this change.

Set log4j.logger.bw.logger property

5. locate the file *BC_HOME*`\lib\is\log4j.xml`
6. Find the following line:

```
<logger name="bw.logger">
   <level value ="INFO"/>
 <appender-ref ref="tibco_bw_log"/>
 </logger>
```

   `INFO` is the default setting you can change it to `DEBUG`.

TIBCO LogLogic Integration

When you select DEBUG from log level, the details of the logs related to the transaction processing are traced. These logs are in a format that can be easily parsed and analyzed through TIBCO LogLogic. When these detailed logs are combined with the Gateway Server logs and the audit database, they provide an end to end view of the processing of a transaction. Below is a sample entry:

2017 Jun 30 13:02:51:577 GMT +0530 BW.BusinessConnect-Interior_Server Debug [bw.logger]

BW-EXT-LOG-300002 Job-4030.4030.B910EE33-F6C9-4EF3-BB3A-B5F188A89805.<machine

name>-t460-Interior-Server IBHTTPHandler.createAndSetSyncReply: HTTP message received. Returning HTTP 200

OK.

View the Log

You can view tracing output in the logs *TIBCO_HOME*\tra\domain*domain_name*\application\logs\BusinessConnect-Interior_Server.log

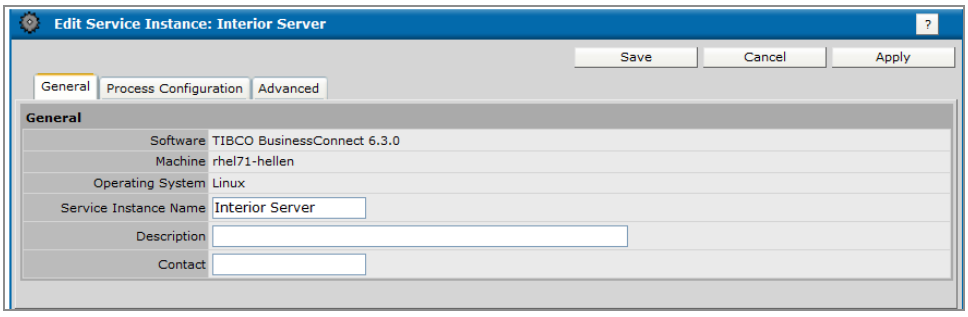
Enabling Verbose Tracing for TIBCO ActiveMatrix BusinessWorks Service Instances

To enable verbose tracing for TIBCO ActiveMatrix BusinessWorks service instances follow these steps.

1. In the Configuration Builder window, click the *hostname*-**Interior_Server** link (where *hostname* is the name of the machine on which the TIBCO BusinessConnect server component is deployed)

   The Dialog Edit Service Instance: Interior Server appears with three tabs: General, Process Configuration, and Advanced.

Figure 8: Editing Service Instance Dialog



2. Click the **Process Configuration** tab.
3. In the General section, select the **Enable Verbose Tracing** check box.
4. Click **Save**.

5. Redeploy and restart TIBCO BusinessConnect.

For more information, see *TIBCO ActiveMatrix BusinessWorks Administration*, section Changing Server Settings.

Enabling Tracing for the TIBCO BusinessConnect Palette

To enable tracing in TIBCO BusinessConnect palette, create a global variable `bc.palette.tracing` and specify one of the following values: `debug`, `error`, `warn`, or `info`. In `debug`, you'll see every message on transport coming and going before dispatching to TIBCO ActiveMatrix BusinessWorks or before sending from TIBCO ActiveMatrix BusinessWorks.

On high volume and/or large messages in debug mode, performance will be heavily affected.

Production environments *should* set this property's value to `none` or remove this global variable altogether. For verbose tracing (most of the comments are such), the `debug` value is recommended.

Tracing is also self-describing as the palette reports the status of its tracing when initialized either from the designer or loaded runtime. If the property is not configured, the TIBCO BusinessConnect palette displays the name of the tracing property as well as the possible values that you can specify.

View the Log

There are two different contexts when using the TIBCO BusinessConnect palette:

- Design-time
- Runtime

During design-time, tracing output goes to the `stdout.log` file of the folder set in **Edit > Preferences > General > User Directories > User Log Directory**.

During runtime, tracing output goes to the following locations:

- The project’s log file wherever the project was deployed
- The project’s log in the folder specified in the design-time setting when the project is run in the test-engine.

Checking the State of the Interior Server Instance

1. Expand **Application Management > BusinessConnect > Service Instances**.

   The Service Instances window appears.

2. View the state of the Service Instance.

   - If the state is `Running` and Status is `OK`, your service instance is up and running.
   - If the `Start successfully deployed services` checkbox was not selected in Checking the State of the Interior Server Instance, you can start the `TIBCO BusinessConnect` instance by selecting the instance checkbox and clicking **Start**.
   - To start, restart, stop, or kill any of the instances, select the checkbox next to the instance and click the appropriate button.

Starting and Stopping the Server

In most cases you will start or stop the server as part of the deployment process.

To start or stop the server outside the deployment process, do the following:

1. Expand **Application Management > BusinessConnect > Services Instances**.
2. View the state of the Service Instance.
3. Select the checkbox next to the instance you want to start or stop.
4. Click **Start** or **Stop**.

Note Although the TIBCO BusinessConnect instance can be started from the command line, starting from within TIBCO Administrator is the *recommended* method.

Removing TIBCO BusinessConnect

Undeploying TIBCO BusinessConnect

To undeploy the TIBCO BusinessConnect application perform these steps:

1. Expand **Application Management > All Applications**.
2. Select the checkbox next to **BusinessConnect**.

   Undeploy becomes active.
3. Click **Undeploy**.
4. Click **OK**.

⚠ **Warning**	Make sure that TIBCO Hawk agents are running on all target machines.

Deleting TIBCO BusinessConnect

To delete the TIBCO BusinessConnect application perform these steps:

1. Expand **Application Management > All Applications**.
2. Select the checkbox next to **BusinessConnect**.

   Delete becomes active.
3. Click **Delete**.

   A dialog appears to verify that you want to delete the application.

Hawk Microagents for Interior Server

The BusinessConnect engine implements a set of microagents to monitor and manage the various polling operations that happen inside the BusinessConnect engine.

Below are the microagent names and functions:

Server Name	Description
SCHDTASKPOLLER	Poller to look for scheduled tasks that need to be fired.
RESENDPOLLER	Poller to look for transactions that need to be resent.
QUEUEACTIONPOLLER	Poller to look for messages to send off the message queue (primarily batch transactions)
MDNPOLLER	Poller to look for overdue MDNs with AS2.
InteriorController	The main BC engine
HIBERPOLLER	Poller to look up transactions that have hibernated too long
FTPPOLLER	Poller that triggers inbound FTP inbound sessions on a periodic basis. This poller is vulnerable to hanging, so it would be good to monitor it.
CREDPOLLER	Expired credential poller. Checks certificate store for expired certificateses on a regular basis.
CANCELPOLLER	Poller that looks at audit log for transaction that have been marked for cancellation. Primarily for RosettaNet and ebXML.

The most common reason for monitoring TIBCO BusinessConnect is for the FTP pollers. In an environment where there may not be reliable data connections to trading partners, there is a tendency for this poller to "hang" because of a bad connection to a SFTP trading partner. Using microagents, you can eliminate this issue with a Hawk rule and rulebase.

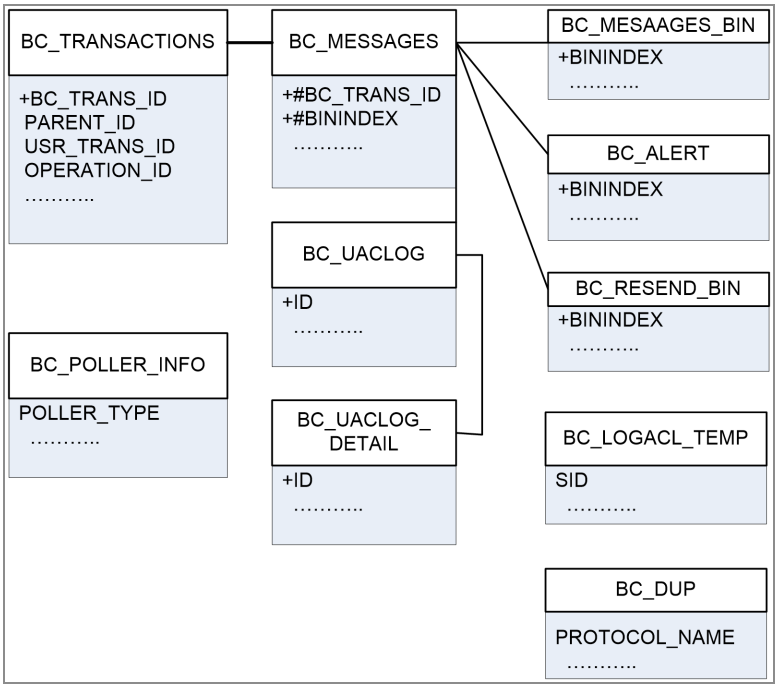
Database Schema Definition

To replace the deprecated Archiver Tool from the previous releases, details about Audit, Non-Repudiation and Runtime schemas are supplied in this section for users to create their own archiving strategy.

Audit Schema Details

Tables that are part of the audit schema are depicted in Audit_RelationShip.

Figure 9: Audit_RelationShip



For the schema above, the values are as shown in Audit_RelationShip Values.

Audit_RelationShip Values

Name	Value
Name	Audit_RelationShip
Data Model	Physical

Table summaries for the audit schema are explained in Audit Schema Tables Summary.

Audit Schema Tables Summary

Name	Documentation
BC_TRANSACTIONS	This is the main table that holds all the summary rows for auditing. An audit trail happens when: • TIBCO BusinessConnect sends or receives messages from a trading partner. • TIBCO BusinessConnect Gateway engine bootstraps and starts up the event sources, as well as when BusinessConnect PartnerExpress or BusinessConnect FTP Server users log in. • TIBCO BusinessConnect PartnerExpress or TIBCO BusinessConnect FTP Server users, or other business protocol partners, transfer messages or receive messages to TIBCO BusinessConnect. The BC_TRANS_ID column is a GUID generated by TIBCO BusinessConnect that holds the latest transaction STATUS from the partner or from the messages sent from the private process. The STARTDATE column is stamped when the first entry is inserted, and the TS column is updated every time a child row is inserted in the BC_MESSAGES table. Depending on the business protocol type (such as TIBCO BusinessConnect EDI Protocol powered by Instream, TIBCO BusinessConnect Services Plug-in, TIBCO BusinessConnect ebXML Protocol, TIBCO BusinessConnect RosettaNet Protocol, TIBCO BusinessConnect SOAP Protocol) or the startup of the TIBCO BusinessConnect Gateway engines, various column values are populated differently in order to show the views differently when looked from the

Name	Documentation
	TIBCO Administrator UI. Usage of the AUX columns is completely governed by individual business protocols or Gateway instances, while the log viewer for individual plug-ins generates a screen layout based on the metadata of these plug-ins. Users who archive the BC_TRANSACTIONS data should query based on the STATUS column and the TS column, which would give all the transactions between the dates specified for certain STATUSes. **Note**: This table will grow based on the exchanged transactions, as well as the number of times the Gateway instances are restarted. This table should be added as a part of the archiving process; for example, a search for `Select * from BC_TRANSACTIONS` where STATUS is such as %COMPLETED% OR %ERROR% and TS is <from Date> .
	A typical sample archive query for Oracle can be such as follows: **Select * from BC_TRANSACTIONS where STATUS in ('COMPLETED','ERROR') AND TO_TIMESTAMP(TO_CHAR(TS,'YYYY-MM-DD HH24:MI:SS.FF'),'YYYY-MM-DD HH24:MI:SS.FF') <= TO_TIMESTAMP('<to date>', 'YYYY-MM-DD HH24:MI:SS.FF') AND TO_TIMESTAMP(TO_CHAR(TS,'YYYY-MM-DD HH24:MI:SS.FF'),'YYYY-MM-DD HH24:MI:SS.FF') >= TO_TIMESTAMP('<from date>','YYYY-MM-DD HH24:MI:SS.FF')** A typical sample archive query for Microsoft SQL database can be such as follows: **Select * from BC_TRANSACTIONS where STATUS in ('COMPLETED','ERROR') AND convert(datetime,convert(varchar, TS, 121), 121) >= convert (datetime, '<from_date>', 121) AND convert(datetime,convert(varchar, TS, 121), 121) <= convert (datetime, '<to_date>', 121)** (The query varies based on the database type)
BC_MESSAGES	This table provides an audit trail for each transaction summary row. The different state depends on the type of TIBCO BusinessConnect transaction, protocol, and plug-in variations that are used; it also varies based on the stage of transition a transaction enters, such as

Name	Documentation
	authentication, decryption, and so on: • TIBCO BusinessConnect audit trails, startup activities, and transitions of each Gateway service in the Gateway Service Instance of the log in the TIBCO Administrator UI. It also audit trails the transaction exchange between partners and TIBCO BusinessConnect in the appropriate protocol level audit trail. • Gateway Service Session login for user level activities for FTP Server and PartnerExpress are logged and an audit trail gets generated with transitions. • Business protocol level audits are captured and can be viewed in the appropriate Protocol Log Viewer section; for example, TIBCO BusinessConnect Services Plug-in has its own Log Viewer screen that audits the different transitions the transactions go through when a partner sends or receives a message. Various audit trail activities keep adding up and a table can grow significantly. This table, along with its BIN table (`BC_MESSAGES_BIN`, `BC_RESEND_BIN`) should also be archived. `Usage of the AUX` columns is completely ruled by individual business protocols or Gateway instances, while the log viewer for individual plug-ins generates a screen layout based on the metadata of these plug-ins. **Note**: This table would grow based on the exchanged transactions, as well as the number of times the Gateway instances are restarted. This table should be added as a part of the archiving process.
	A typical sample archive query for Oracle can be such as follows: Select * from BC_MESSAGES where BC_TRANS_ID IN (SELECT BC_TRANS_ID FROM BC_TRANSACTIONS where TO_TIMESTAMP(TO_CHAR (TS,'YYYY-MM-DD HH24:MI:SS.FF'),'YYYY-MM-DD HH24:MI:SS.FF') <= TO_TIMESTAMP('<to_date>', 'YYYY-MM-DD HH24:MI:SS.FF') AND TO_TIMESTAMP(TO_CHAR(TS,'YYYY-MM-DD HH24:MI:SS.FF'),'YYYY-MM-DD HH24:MI:SS.FF') >= TO_TIMESTAMP('<from_date>','YYYY-MM-DD HH24:MI:SS.FF') AND STATUS IN ('COMPLETED', 'ERROR'))

Name	Documentation
	A typical sample archive query for Microsoft SQL database can be such as follows: Select * from BC_MESSAGES where BC_TRANS_ID IN (select BC_TRANS_ID FROM BC_TRANSACTIONS where convert(datetime,convert(varchar, TS, 121), 121) >= convert(datetime, '<from_date>', 121) AND convert (datetime,convert(varchar, TS, 121), 121) <= convert(datetime, '<to_date>', 121) AND STATUS IN ('COMPLETED', 'ERROR')) The query varies based on the database type.
BC_MESSAGES_BIN	This table has a foreign key to the BC_MESSAGES table. It contains certain transition details that TIBCO BusinessConnect logs, such as failure descriptions or payloads written by TIBCO BusinessConnect protocols when the option Include Message in Log is enabled. Rows in this table need to be archived to avoid the table to grow too big. **Note**: This table will grow based on the exchanged transactions for business protocols and should be added as part of the archiving process. A typical sample archive query for Oracle varies based on the database type and can be such as follows: **Select * from BC_MESSAGES_BIN where BININDEX IN (SELECT BININDEX FROM BC_MESSAGES where BC_TRANS_ID IN (SELECT BC_TRANS_ID FROM BC_TRANSACTIONS where TO_TIMESTAMP (TO_CHAR(TS,'YYYY-MM-DD HH24:MI:SS.FF'),'YYYY-MM-DD HH24:MI:SS.FF') <= TO_TIMESTAMP('<to_date>', 'YYYY-MM-DD HH24:MI:SS.FF') AND TO_TIMESTAMP(TO_CHAR(TS,'YYYY-MM-DD HH24:MI:SS.FF'),'YYYY-MM-DD HH24:MI:SS.FF') >= TO_TIMESTAMP('<from_date>','YYYY-MM-DD HH24:MI:SS.FF') AND STATUS IN ('COMPLETED', 'ERROR')))** A typical sample archive query for Microsoft SQL database can be as follows: **Select * from BC_MESSAGES_BIN where BININDEX IN (SELECT BININDEX FROM BC_MESSAGES A, BC_TRANSACTIONS B where A.BC_**

Name	Documentation
	TRANS_ID = B.BC_TRANS_ID AND convert(datetime,convert(varchar, B.TS, 121), 121) >= convert(datetime, '<from_date>', 121) AND convert(datetime,convert(varchar, B.TS, 121), 121) <= convert (datetime, '<to_date>', 121) AND B.STATUS IN ('COMPLETED', 'ERROR'))
BC_ALERT	This table has a foreign key to the BC_MESSAGES table and contains Alert details that TIBCO BusinessConnect logs, such as expired certificates, and so on. Rows in this table need to be archived to avoid the table to grow too big. A typical sample archive query for Oracle varies based on the database type and can be such as follows: **Select * from BC_ALERT where BININDEX IN (SELECT BININDEX FROM BC_MESSAGES where BC_TRANS_ID IN (SELECT BC_TRANS_ID FROM BC_TRANSACTIONS where TO_TIMESTAMP (TO_CHAR(TS,'YYYY-MM-DD HH24:MI:SS.FF'),'YYYY-MM-DD HH24:MI:SS.FF') <= TO_TIMESTAMP(<to_date>, 'YYYY-MM-DD HH24:MI:SS.FF') AND TO_ TIMESTAMP(TO_CHAR(TS,'YYYY-MM-DD HH24:MI:SS.FF'),'YYYY-MM-DD HH24:MI:SS.FF') >= TO_TIMESTAMP(<from_date>,'YYYY-MM-DD HH24:MI:SS.FF') AND STATUS IN ('COMPLETED', 'ERROR')))** A typical sample archive query for Microsoft SQL database can be as follows: **Select * from BC_ALERT where BININDEX IN (SELECT BININDEX FROM BC_MESSAGES A, BC_TRANSACTIONS B where A.BC_TRANS_ID = B.BC_ TRANS_ID AND convert(datetime,convert(varchar, B.TS, 121), 121) >= convert(datetime, '<from_date>', 121) AND convert (datetime,convert(varchar, B.TS, 121), 121) <= convert (datetime, '<to_date>', 121) AND B.STATUS IN ('COMPLETED', 'ERROR'))**
BC_RESEND_BIN	This table stores all resendable information to be used by TIBCO BusinessConnect. The stored messages are not in a readable format since it is proprietary information. **Note**: This table will grow based on the exchanged transactions for

Name	Documentation
	business protocols and should be added as part of the archiving process. A typical sample archive query for Oracle varies based on the database type and can be such as follows: **Select * from BC_RESEND_BIN where BININDEX IN (SELECT BININDEX from BC_MESSAGES where BC_TRANS_ID IN (SELECT BC_TRANS_ID from BC_TRANSACTIONS where TO_TIMESTAMP (TO_CHAR(TS,'YYYY-MM-DD HH24:MI:SS.FF'),'YYYY-MM-DD HH24:MI:SS.FF') <= TO_TIMESTAMP('<to_date>', 'YYYY-MM-DD HH24:MI:SS.FF') AND TO_TIMESTAMP(TO_CHAR(TS,'YYYY-MM-DD HH24:MI:SS.FF'),'YYYY-MM-DD HH24:MI:SS.FF') >= TO_TIMESTAMP('<from_date>','YYYY-MM-DD HH24:MI:SS.FF') AND STATUS IN ('COMPLETED', 'ERROR')))** A typical sample archive query for Microsoft SQL database can be as follows: **Select * from BC_RESEND_BIN where BININDEX IN (SELECT BININDEX from BC_MESSAGES A, BC_TRANSACTIONS B where A.BC_TRANS_ID = B.BC_TRANS_ID AND convert(datetime,convert(varchar, B.TS, 121), 121) >= convert (datetime, '<from_date>', 121) AND convert(datetime,convert (varchar, B.TS, 121), 121) <= convert(datetime, '<to_date>', 121) AND B.STATUS IN ('COMPLETED', 'ERROR'))**
BC_UACLOG	The table BC_UACLOG logs any edits that happen during the TIBCO BusinessConnect configuration dome with TIBCO Administrator. All changes made by the user logged into the TIBCO Administrator UI are captured in this table. Any changes to participants, business agreements and operation editor are also part of the audit trail captured in this table. The tables BC_UACLOG and BC_UACLOG_DETAIL can be purged and archived based on the column OPERATION_TIME to be a part of the selection query. **Note**: This table would grow based on the modifications to participants, business agreements, and operations. It should be added as part of the archiving process.

Name	Documentation
BC_LOGACL_TEMP	This table holds the User-to-Partner information and is only transient until the user is logged and querying the log viewer.
BC_DUP	This table stores the transactions hash values, which are used to indicate whether an incoming or outgoing transaction is a duplicate. The information stored is completely controlled by individual business protocols. **Note**: This table will grow as transactions are exchanged. It should be added as a part of the archiving process.
BC_UACLOG_DETAIL	**Note**: This table will grow based on the modifications to participants, business agreements, and operations. It should be added as a part of the archiving process.
BC_POLLER_INFO	This table is used to store transient information populated by the TIBCO Administrator UI, such as Resend Actions or Queue Action. This table does not grow and need not be a part of the archiving process.

Note The Relationship diagrams describe all the tables that are part of the Audit Log in TIBCO BusinessConnect. Not all entity tables are used at this time, but are created so as to be prepared for future functionality.

BC_TRANSACTIONS

Details for the table BC_TRANSACTIONS are explained in BC_TRANSACTIONS: Details.

BC_TRANSACTIONS: Details

Name	Value
Data Model	Physical

Name	Value
Documentation	This table is the main table that holds all the summary rows for auditing. An audit trail happens when: • TIBCO BusinessConnect sends or receives messages from the trading partner. • The Gateway engine bootstraps and starts up the event sources, as well as when PartnerExpress or FTP Server users log in. • PartnerExpress users, FTP Server users, or other business protocol partners transfer messages or receive messages to TIBCO BusinessConnect. The column BC_TRANS_ID is a GUID generated by TIBCO BusinessConnect that holds the last STATUS of the transaction from the partner or from the messages sent from the private process. The column STARTDATE is stamped when the first entry is inserted, and TS column is updated every time a child row is inserted in the BC_MESSAGES table. Depending on the business protocol type (such as TIBCO BusinessConnect™ EDI Protocol powered by Instream®, TIBCO BusinessConnect Services Plug-in, TIBCO BusinessConnect™ ebXML Protocol, TIBCO BusinessConnect™ RosettaNet Protocol, TIBCO BusinessConnect™ SOAP Protocol) or the startup of the TIBCO BusinessConnect Gateway engines, various column values are populated differently in order to show the views differently when looked from the TIBCO Administrator UI. Usage of the AUX columns is completely governed by individual business protocols or Gateway instances, while the log viewer for individual plug-ins generates a screen layout based on the metadata of these plug-ins. Users who archive the BC_TRANSACTIONS data should query based on the STATUS column and the TS column, which would give all the transactions between the dates specified for certain STATUSes. **Note**: This table will grow based on the exchanged transactions, as well as the number of times the Gateway instances are restarted. This table should be added as a part of the archiving process; for example, a search for `Select * from BC_TRANSACTIONS` where STATUS is

Name	Value
	such as %COMPLETED% OR %ERROR% and TS is <from Date> . A typical sample archive query for an Oracle database can be as follows: **Select * from BC_TRANSACTIONS where STATUS IN ('COMPLETED','ERROR') AND TO_TIMESTAMP(TO_CHAR(TS,'YYYY-MM-DD HH24:MI:SS.FF'),'YYYY-MM-DD HH24:MI:SS.FF') <= TO_TIMESTAMP('<to date>', 'YYYY-MM-DD HH24:MI:SS.FF') AND TO_TIMESTAMP(TO_CHAR(TS,'YYYY-MM-DD HH24:MI:SS.FF'),'YYYY-MM-DD HH24:MI:SS.FF') >= TO_TIMESTAMP('<from date>','YYYY-MM-DD HH24:MI:SS.FF')** A typical sample archive query for Microsoft SQL database can be as follows: **Select * from BC_TRANSACTIONS where STATUS IN ('COMPLETED','ERROR') AND convert(datetime,convert(varchar, TS, 121), 121) >= convert(datetime, '<from_date>', 121) AND convert (datetime,convert(varchar, TS, 121), 121) <= convert(datetime, '<to_date>', 121)** (Queries vary depending on the database type.)
User IDLast Numeric Value	0
Records	N/A

DDLClauses	`CREATE TABLE BC_TRANSACTIONS (`
	` BC_TRANS_ID VARCHAR2(512) primary key,`
	` PARENT_ID varchar2(512),`
	` USER_TRANS_ID VARCHAR2(512) null,`
	` OPERATION_ID VARCHAR2(512) null,`
	` TPNAME VARCHAR2(128) NULL,`
	` TPDOMAIN VARCHAR2(32) NULL,`
	` TPID VARCHAR2(32) NULL,`
	` HOSTNAME VARCHAR2(128) NULL,`
	` HOSTDOMAIN VARCHAR2(32) NULL,`
	` HOSTID VARCHAR2(32) NULL,`
	` PROTOCOL_VERSION VARCHAR2(128) NULL,`
	` PROTOCOL_NAME VARCHAR2(32) NULL,`
	` INSTALLATION_NAME VARCHAR2(32) NULL,`

Name	Value

```
  STATUS VARCHAR2(64) NULL,
  HOST_INITIATES VARCHAR2(5) NULL,
  USAGE_MODE VARCHAR2(16) NULL,
  RESEND_USERNAME VARCHAR2(128) NULL,
  GS_TYPE VARCHAR2(128) NULL,
  GS_INSTANCE_INFO VARCHAR2(128) NULL,
  EXTERNAL_USER VARCHAR2(128) NULL,
  AUX1 VARCHAR2(512) NULL,
  AUX2 VARCHAR2(512) NULL,
  AUX3 VARCHAR2(512) NULL,
  AUX4 VARCHAR2(512) NULL,
  AUX5 VARCHAR2(512) NULL,
  AUX6 VARCHAR2(512) NULL,
  AUX7 VARCHAR2(512) NULL,
  AUX8 VARCHAR2(512) NULL,
  AUX9 VARCHAR2(512) NULL,
  AUX10 VARCHAR2(512) NULL,
  AUX11 VARCHAR2(512) NULL,
  AUX12 VARCHAR2(512) NULL,
  AUX13 VARCHAR2(512) NULL,
  AUX14 VARCHAR2(512) NULL,
  AUX15 VARCHAR2(512) NULL,
  STARTDATE TIMESTAMP DEFAULT SYSTIMESTAMP,
  TS TIMESTAMP DEFAULT SYSTIMESTAMP
 );
```

Columns summary for the table `BC_TRANSACTIONS` is shown in BC_TRANSACTIONS: Columns Summary.

BC_TRANSACTIONS: Columns Summary

Name	Data Type	Constraints	Nullable	Documentation
BC_TRANS_ID	varchar2(512)	PKUnique	No	Internal Unique ID generated by the TIBCO BusinessConnect Server
PARENT_ID	varchar2(512)		No	This is used when Resend Options are used with the TIBCO BusinessConnect Log Viewer
USER_TRANS_ID	varchar2(512)		Yes	User Specific Transaction

Name	Data Type	Constraints	Nullable	Documentation
				ID populated from the TIBCO ActiveMatrix BusinessWorks private process
OPERATION_ID	varchar2(512)		Yes	Operation ID provided from the private process, or inferred when receiving a request from the partner.
TPNAME	varchar2(128)		Yes	The column where the partner name is populated by TIBCO BusinessConnect.
TPDOMAIN	varchar2(32)		Yes	The column where the partner domain is populated by TIBCO BusinessConnect and is specific for a business protocol.
TPID	varchar2(32)		Yes	The column where the partner identity is populated by TIBCO BusinessConnect and is specific for a business protocol.
HOSTNAME	varchar2(128)		Yes	The column where the host name is populated by TIBCO BusinessConnect.
HOSTDOMAIN	varchar2(32)		Yes	The column where the host domain is

Name	Data Type	Constraints	Nullable	Documentation
				populated by TIBCO BusinessConnect and is specific for a business protocol.
HOSTID	varchar2(32)		Yes	The column where the host identity is populated by TIBCO BusinessConnect and is specific for a business protocol.
PROTOCOL_ VERSION	varchar2(128)		Yes	The column where the business protocol version is populated by TIBCO BusinessConnect.
PROTOCOL_NAME	varchar2(32)		Yes	Specifies the actual business protocol, or one of the Gateway Services that are used to log during bootstrapping of the Gateway engines.
INSTALLATION_ NAME	varchar2(32)		Yes	The column where the business installation name is populated by TIBCO BusinessConnect.
STATUS	varchar2(64)		Yes	The status value specific to a protocol or Gateway Service type. This is the last status value set to indicate the state at which the trail is maintained.

Name	Data Type	Constraints	Nullable	Documentation
HOST_INITIATES	varchar2(5)		Yes	Indicates whether the transaction is being initiated by TIBCO BusinessConnect or by the partner.
USAGE_MODE	varchar2(16)		Yes	
RESEND_ USERNAME	varchar2(128)		Yes	This column is populated when a resend action happens after the Administrator logged in a user.
GS_TYPE	varchar2(128)		Yes	The column populated by TIBCO BusinessConnect, where this transaction entry will be available for the PartnerExpress History view for a partner.
GS_INSTANCE_ INFO	varchar2(128)		Yes	Specifies which Gateway Instance transferred this inbound message.
EXTERNAL_USER	varchar2(128)		Yes	Not used at this point
AUX1	varchar2(512)		Yes	
AUX2	varchar2(512)		Yes	
AUX3	varchar2(512)		Yes	
AUX4	varchar2(512)		Yes	
AUX5	varchar2(512)		Yes	

Name	Data Type	Constraints	Nullable	Documentation
AUX6	varchar2(512)		Yes	
AUX7	varchar2(512)		Yes	
AUX8	varchar2(512)		Yes	
AUX9	varchar2(512)		Yes	
AUX10	varchar2(512)		Yes	
AUX11	varchar2(512)		Yes	
AUX12	varchar2(512)		Yes	
AUX13	varchar2(512)		Yes	
AUX14	varchar2(512)		Yes	
AUX15	varchar2(512)		Yes	
STARTDATE	timestamp		Yes	Start date and time of a transaction that occurred.
TS	timestamp		Yes	The column indicating latest updates to the row.

Indices information for the table `BC_TRANSACTIONS` is shown in BC_TRANSACTIONS Indices: BC_TRANSACTIONS_UTXID and BC_TRANSACTIONS Indices: BC_TRANSACTIONS_LOGVIEWER.

BC_TRANSACTIONS Indices: BC_TRANSACTIONS_UTXID

BC_TRANSACTIONS_UTXID	
User IDLast Numeric Value	0

BC_TRANSACTIONS_UTXID	
Unique	false
Clustered	Unspecified

BC_TRANSACTIONS Indices: BC_TRANSACTIONS_LOGVIEWER

BC_TRANSACTIONS_LOGVIEWER	
User IDLast Numeric Value	0
Unique	false
Clustered	Unspecified

Relationship information for the table `BC_TRANSACTIONS` is shown in *BC_TRANSACTIONS Relationships*.

BC_TRANSACTIONS Relationships

BC_NR_BIN_MID: Relationship	
To	BC_MESSAGES
User IDLast Numeric Value	0
Identifying	true
Subtype	false
On Delete	Cascade
To Multiplicity	1..*
From Multiplicity	1

BC_NR_BIN_MID: Relationship	
Sync To Association	Yes
Data Model	Physical

BC_MESSAGES

Details for the table `BC_MESSAGES` are explained in BC_MESSAGES: Details.

BC_MESSAGES: Details

Name	Value
`Data Model`	Physical
`Documentation`	This table provides the audit trail for each transaction summary row. The different state depends on the type of TIBCO BusinessConnect transaction, protocol, and plug-in variations that are used. It also varies based on the stage of transition that the transaction enters, such as authentication, decryption, and so on. • TIBCO BusinessConnect generates audit trails of the startup activities, as well as transitions of each Gateway Services in the Gateway Service Instance of the TIBCO Administrator log. It also generates audit trails of the transaction exchange between partners and TIBCO BusinessConnect in the appropriate protocol level of the audit trail. • Gateway Service session logins for user level activities for the FTP Server and PartnerExpress are logged and audit trails are generated with transitions. • Business protocol level audits are captured and can be viewed in the appropriate protocol log viewer section; for example, TIBCO BusinessConnect Services Plug-in has its own log viewer screen that audits the different transitions the transactions go through when a partner sends or receives a message. These audit trail activities keep adding up and the table can grow

Name	Value
	significantly. This table along with its BIN table (BC_MESSAGES_BIN, BC_RESEND_BIN) should also be archived. Usage of the AUX columns is completely governed by individual business protocols or Gateway instances, while the log viewer for individual plug-ins generates a screen layout based on the metadata of these plug-ins. **Note**: This table will grow based on the exchanged transactions, as well as the number of times the Gateway instances are restarted. It should be added as a part of the archiving process. A typical sample archive query for an Oracle database can be as follows: **Select * from BC_MESSAGES where BC_TRANS_ID IN (SELECT BC_TRANS_ID from BC_TRANSACTIONS where TO_TIMESTAMP(TO_CHAR(TS,'YYYY-MM-DD HH24:MI:SS.FF'),'YYYY-MM-DD HH24:MI:SS.FF') <= TO_TIMESTAMP('<to date>', 'YYYY-MM-DD HH24:MI:SS.FF') AND TO_TIMESTAMP(TO_CHAR(TS,'YYYY-MM-DD HH24:MI:SS.FF'),'YYYY-MM-DD HH24:MI:SS.FF') >= TO_TIMESTAMP('<from date>','YYYY-MM-DD HH24:MI:SS.FF') AND STATUS IN ('COMPLETED', 'ERROR'))** A typical sample archive query for a Microsoft SQL database can be as follows: **Select * from BC_MESSAGES where BC_TRANS_ID IN (SELECT BC_TRANS_ID FROM BC_TRANSACTIONS where convert(datetime,convert (varchar, TS, 121), 121) >= convert(datetime, '<from_date>', 121) AND convert(datetime,convert(varchar, TS, 121), 121) <= convert(datetime, '<to_date>', 121) AND STATUS IN ('COMPLETED', 'ERROR'))** (Queries vary based on the database type.)
User IDLast Numeric Value	0
Records	N/A
DDLClauses	`CREATE TABLE BC_MESSAGES (`

Name	Value
	BC_TRANS_ID VARCHAR2(512) NOT NULL, BININDEX NUMBER(18) NOT NULL UNIQUE, USER_MESSAGE_ID VARCHAR2(512) NULL, STATUS VARCHAR2(64) NULL, STATE VARCHAR2(64) NULL, DESCRIPTION VARCHAR2(4000) NULL, TS TIMESTAMP DEFAULT SYSTIMESTAMP, IS_REPAIRABLE VARCHAR2(5) NULL, RESEND CHAR(1) NULL, PP_TRANSPORT_TYPE VARCHAR2(10) NULL, PAYLOAD_SIZE NUMBER(10) null, AUX1 VARCHAR2(512) NULL, AUX2 VARCHAR2(512) NULL, AUX3 VARCHAR2(512) NULL, AUX4 VARCHAR2(512) NULL,

Name	Value
	AUX5 VARCHAR2(512) NULL, AUX6 VARCHAR2(512) NULL, AUX7 VARCHAR2(512) NULL, AUX8 VARCHAR2(512) NULL, AUX9 VARCHAR2(512) NULL, CONSTRAINT MESSAGES FOREIGN KEY (BC_TRANS_ID) references BC_ TRANSACTIONS(BC_TRANS_ID) ON DELETE CASCADE);

Columns summary for the table BC_MESSAGES is shown in BC_MESSAGES: Columns Summary.

BC_MESSAGES: Columns Summary

Name	Data Type	Constraints	Nullable	Documentation
BC_TRANS_ID	varchar2 (512)	PK/FK (BC_ TRANSACTIONS.BC_ TRANS_ID)	No	Foreign key to the column BC_ TRANSACTIONS.BC_ TRANS_ID and is a auto-generated UID.
BININDEX	number(18)	PK/FK (BC_RESEND_ BIN.BININDEX, BC_ MESSAGES_ BIN.BININDEX, BC_ ALERT.BININDEX); Unique	No	Primary unique index column generated by TIBCO BusinessConnect.
USER_MESSAGE_ ID	varchar2 (512)		Yes	This column stores messages or

Name	Data Type	Constraints	Nullable	Documentation
				transaction IDs sent from the private process to TIBCO BusinessConnect.
STATUS	varchar2 (64)		Yes	Audit trail status for this row at the time the TIBCO BusinessConnect is processing; for example, it could be encrypting or decrypting a payload, or it could be doing a validation at the time.
STATE	varchar2 (64)		Yes	Audit trail state for this row at the time TIBCO BusinessConnect is processing; for example, it could be encrypting or decrypting a payload, or it could be doing a validation at the time.
DESCRIPTION	varchar2 (4096)		Yes	Audit trail description for this row at the time the TIBCO BusinessConnect is processing; for example, it could be encrypting or decrypting a payload, or it could be doing a validation at the time.

Name	Data Type	Constraints	Nullable	Documentation
IS_REPAIRABLE	varchar2(5)		Yes	Not used
RESEND	char(1)		Yes	Indicates whether this row is resendable or not.
PP_ TRANSPORT_ TYPE	varchar2 (10)		Yes	This indicates the private process transport type to be used for the resend transactions.
PAYLOAD_SIZE	number(10)		Yes	If a payload is used to store in `BC_MESSAGES_ BIN`, then this column would indicate the size of the payload.
AUX1	varchar2 (512)		Yes	
AUX2	varchar2 (512)		Yes	
AUX3	varchar2 (512)		Yes	
AUX4	varchar2 (512)		Yes	
AUX5	varchar2 (512)		Yes	
AUX6	varchar2 (512)		Yes	
AUX7	varchar2		Yes	

Name	Data Type	Constraints	Nullable	Documentation
	(512)			
AUX8	varchar2 (512)		Yes	
AUX9	varchar2 (512)		Yes	

Indices information for the table `BC_MESSAGES` is shown in BC_MESSAGES Indices: BC_MESSAGES_IDX and BC_MESSAGES Indices: BC_MESSAGES_USERMESSAGEID.

BC_MESSAGES Indices: BC_MESSAGES_IDX

BC_MESSAGES_IDX	
User IDLast Numeric Value	0
Unique	false
Clustered	Unspecified

BC_MESSAGES Indices: BC_MESSAGES_USERMESSAGEID

BC_MESSAGES_USERMESSAGEID	
User IDLast Numeric Value	0
Unique	false
Clustered	Unspecified

Constraints information for the table `BC_MESSAGES` is shown in BC_MESSAGES: Constraints.

BC_MESSAGES: Constraints

MESSAGES	
User IDLast Numeric Value	0
Check Constraint	`MESSAGES FOREIGN KEY (BC_TRANS_ID) references BC_TRANSACTIONS(BC_TRANS_ID) ON DELETE CASCADE`

Relationship information for the table `BC_MESSAGES` is shown in *BC_MESSAGES Relationships: MESSAGES*, *BC_MESSAGES Relationships: RESEND_BIN*, *BC_MESSAGES Relationships: BC_MESSAGES_BIN*, *and BC_MESSAGES Relationships: BC_ALERT*.

BC_MESSAGES Relationships: MESSAGES

MESSAGES: Relationship	
From	BC_TRANSACTIONS
User IDLast Numeric Value	0
Identifying	true
Subtype	false
On Delete	Cascade
To Multiplicity	1..*
From Multiplicity	1
Sync To Association	Yes
Data Model	Physical

BC_MESSAGES Relationships: RESEND_BIN

RESEND_BIN: Relationship	
From	BC_RESEND_BIN
User IDLast Numeric Value	0
Identifying	true
Subtype	false
On Delete	Cascade
To Multiplicity	1
From Multiplicity	1
Sync To Association	Yes
Data Model	Physical

BC_MESSAGES Relationships: BC_MESSAGES_BIN

BC_MESSAGES: Relationship	
From	BC_MESSAGES_BIN
User IDLast Numeric Value	0
Identifying	true
Subtype	false
On Delete	Cascade
To Multiplicity	0..1

BC_MESSAGES: Relationship	
From Multiplicity	1
Sync To Association	Yes
Data Model	Physical

BC_MESSAGES Relationships: BC_ALERT

BC_ALERT: Relationship	
From	BC_ALERT
User IDLast Numeric Value	0
Identifying	true
Subtype	false
On Delete	Cascade
To Multiplicity	0..1
From Multiplicity	1
Sync To Association	Yes
Data Model	Physical

BC_MESSAGES_BIN

Details for the table `BC_MESSAGES_BIN` are explained in BC_MESSAGES_BIN: Details.

BC_MESSAGES_BIN: Details

Name	Value
Data Model	Physical
Documentation	This table has a foreign key to the BC_MESSAGES table and contains certain transition details that the TIBCO BusinessConnect logs, such as failure descriptions or payloads written by TIBCO BusinessConnect protocols when the option Include Message in Log is enabled. Rows in this table needs to be archived since this table can grow big. **Note**: This table would grow based on the transactions exchanged for business protocols. It should be added as a part of the archiving process. A typical sample archive query for an Oracle database can be as follows: **Select * from BC_MESSAGES_BIN where BININDEX IN (SELECT BININDEX from BC_MESSAGES where BC_TRANS_ID IN (SELECT BC_TRANS_ID FROM BC_TRANSACTIONS where TO_TIMESTAMP(TO_CHAR(TS,'YYYY-MM-DD HH24:MI:SS.FF'),'YYYY-MM-DD HH24:MI:SS.FF') <= TO_TIMESTAMP('<to date>', 'YYYY-MM-DD HH24:MI:SS.FF') AND TO_TIMESTAMP(TO_CHAR(TS,'YYYY-MM-DD HH24:MI:SS.FF'),'YYYY-MM-DD HH24:MI:SS.FF') >= TO_TIMESTAMP('<from_date>','YYYY-MM-DD HH24:MI:SS.FF') AND STATUS IN ('COMPLETED', 'ERROR')))** A typical sample archive query for a Microsoft SQL database can be as follows: **Select * from BC_MESSAGES_BIN where BININDEX IN** **(SELECT A.BININDEX FROM BC_MESSAGES A, BC_TRANSACTIONS B where A.BC_TRANS_ID = B.BC_TRANS_ID AND convert(datetime,convert (varchar, B.TS, 121), 121) >= convert(datetime, '<from_date>', 121) AND convert(datetime,convert(varchar, B.TS, 121), 121) <= convert(datetime, '<to_date>', 121) AND B.STATUS IN ('COMPLETED', 'ERROR'))** (Queries vary based on the database type.)
User IDLast Numeric Value	0

Name	Value

DDLClauses	`CREATE TABLE BC_MESSAGES_BIN (`
	`BININDEX NUMBER(18) PRIMARY KEY,`
	`COMPRESSED VARCHAR2(2) NULL,`
	`BINVAL BLOB NULL,`
	`CONSTRAINT MESSAGES_BIN FOREIGN KEY (BININDEX) references`
	`BC_MESSAGES(BININDEX) ON DELETE CASCADE);`

Columns summary for the table `BC_MESSAGES_BIN` is shown in BC_MESSAGES_BIN: Columns Summary.

BC_MESSAGES_BIN: Columns Summary

Name	Data Type	Constraints	Nullable	Documentation
`BININDEX`	number(18)	PK	No	
`COMPRESSED`	varchar2(2)		Yes	
`BINVAL`	blob		Yes	

Constraints information for the table `BC_MESSAGES_BIN` is shown in BC_MESSAGES_BIN Constraints: MESSAGES_BIN.

BC_MESSAGES_BIN Constraints: MESSAGES_BIN

MESSAGES_BIN	
User IDLast Numeric Value	0
Check Constraint	`MESSAGES_BIN FOREIGN KEY (BININDEX) references BC_MESSAGES(BININDEX) ON DELETE CASCADE`

Relationship information for the table `BC_MESSAGES_BIN` is shown in *BC_MESSAGES_BIN: Relationships*.

BC_MESSAGES_BIN: Relationships

BC_MESSAGES: Relationship	
To	BC_MESSAGES
User IDLast Numeric Value	0
Identifying	true
Subtype	false
On Delete	Cascade
To Multiplicity	0..1
From Multiplicity	1
Sync To Association	Yes
Data Model	Physical

BC_ALERT

Details for the table `BC_ALERT` are explained in BC_ALERT: Details.

BC_ALERT: Details

Name	Value
`Data Model`	Physical
`Documentation`	This table has a foreign key to the BC_MESSAGES table and contains Alert details that the TIBCO BusinessConnect logs like expired certificates etc. Rows in this table needs to be archived since this table can grow.

Name	Value
	A typical sample archive query for an Oracle database can be as follows: **Select * from BC_ALERT where BININDEX IN (SELECT BININDEX from BC_MESSAGES where BC_TRANS_ID IN (SELECT BC_TRANS_ID FROM BC_TRANSACTIONS where TO_TIMESTAMP (TO_CHAR(TS,'YYYY-MM-DD HH24:MI:SS.FF'),'YYYY-MM-DD HH24:MI:SS.FF') <= TO_TIMESTAMP('<to date>', 'YYYY-MM-DD HH24:MI:SS.FF') AND TO_TIMESTAMP(TO_CHAR(TS,'YYYY-MM-DD HH24:MI:SS.FF'),'YYYY-MM-DD HH24:MI:SS.FF') >= TO_TIMESTAMP('<from date>','YYYY-MM-DD HH24:MI:SS.FF')AND STATUS IN ('COMPLETED', 'ERROR')))** A typical sample archive query for a Microsoft SQL database can be as follows: **Select * from BC_ALERT where BININDEX IN (SELECT A.BININDEX FROM BC_MESSAGES A, BC_TRANSACTIONS B where A.BC_TRANS_ID = B.BC_TRANS_ID AND convert(datetime,convert(varchar, B.TS, 121), 121) >= convert(datetime, '<from_date>', 121) AND convert(datetime,convert(varchar, B.TS, 121), 121) <= convert(datetime, '<to_date>', 121) AND B.STATUS IN ('COMPLETED', 'ERROR'))** (Queries vary based on the database type.)
User IDLast Numeric Value	0

DDLClauses	`CREATE TABLE BC_ALERT (`
	`BININDEX NUMBER(18) PRIMARY KEY,`
	`USER_NAME VARCHAR2(20) NULL,`
	`ALERT_STATUS VARCHAR2(15) DEFAULT 'NEW',`
	`ADVISORY VARCHAR2(1024) NULL,`
	`SEVERITY VARCHAR2(20) DEFAULT 'Information',`
	`CREATED TIMESTAMP DEFAULT SYSTIMESTAMP,`
	`CONSTRAINT BC_ALERT FOREIGN KEY (BININDEX) references BC_MESSAGES(BININDEX) ON DELETE CASCADE);`

Columns summary for the table `BC_ALERT` is shown in BC_ALERT: Columns Summary.

BC_ALERT: Columns Summary

Name	Data Type	Constraints	Nullable	Documentation
BININDEX	number(18)	PK	No	
USER_NAME	varchar2(20)		Yes	
ALERT_ STATUS	varchar2(15)		No	
ADVISORY	varchar2(1024)		Yes	
SEVERITY	varchar2(1024)		No	
CREATED	timestamp		No	

Constraints information for the table BC_ALERT is shown in BC_ALERT Constraints: BC_ ALERT.

BC_ALERT Constraints: BC_ALERT

MESSAGES_BIN	
User IDLast Numeric Value	0
Check Constraint	CONSTRAINT BC_ALERT FOREIGN KEY (BININDEX) references BC_MESSAGES(BININDEX) ON DELETE CASCADE

Relationship information for the table BC_ALERT is shown in *BC_ALERT: Relationships*.

BC_ALERT: Relationships

BC_ALERT: Relationship	
To	BC_MESSAGES
User IDLast Numeric Value	0
Identifying	true

BC_ALERT: Relationship	
Subtype	false
On Delete	Cascade
To Multiplicity	0..1
From Multiplicity	1
Sync To Association	Yes
Data Model	Physical

BC_RESEND_BIN

Details for the table `BC_RESEND_BIN` are explained in BC_RESEND_BIN: Details.

BC_RESEND_BIN: Details

Name	Value
`Data Model`	Physical
`Documentation`	This table stores the information on all the resendable information to be used by TIBCO BusinessConnect. The stored message is not in a readable format and is proprietary. **Note**: This table will grow based on the exchanged transactions. It should be added as a part of the archiving process. A typical sample archive query for an Oracle database can be as follows: **`Select * from BC_RESEND_BIN where BININDEX IN (SELECT BININDEX FROM BC_MESSAGES A, BC_TRANSACTIONS B where A.BC_TRANS_ID = B.BC_TRANS_ID AND B.PROTOCOL_NAME = '<protocol>' AND TO_TIMESTAMP(TO_CHAR (B.TS,'YYYY-MM-DDHH24:MI:SS.FF'),'YYYY-MM-DD HH24:MI:SS.FF') <= TO_TIMESTAMP('<to date>', 'YYYY-MM-DD HH24:MI:SS.FF') AND TO_TIMESTAMP(TO_CHAR(B.TS,'YYYY-MM-DD HH24:MI:SS.FF'),'YYYY-MM-`**

Name	Value
	DD HH24:MI:SS.FF') >= TO_TIMESTAMP('<from date>','YYYY-MM-DD HH24:MI:SS.FF') AND B.STATUS IN ('COMPLETED', 'ERROR')) A typical sample archive query for a Microsoft SQL database can be as follows: **Select * from BC_RESEND_BIN where BININDEX IN** **(SELECT BININDEX FROM BC_MESSAGES A, BC_TRANSACTIONS B where A.BC_TRANS_ID = B.BC_TRANS_ID AND convert(datetime,convert (varchar, B.TS, 121), 121) >= convert(datetime, '<from_date>', 121) AND convert(datetime,convert(varchar, B.TS, 121), 121) <= convert(datetime, '<to_date>', 121) AND B.STATUS IN ('COMPLETED', 'ERROR'))** (Queries vary based on the database type.)
User IDLast Numeric Value	0

DDLClauses	`CREATE TABLE BC_RESEND_BIN (`
	`  BININDEX  NUMBER(18)  PRIMARY KEY,`
	`  COMPRESSED  VARCHAR2(2)  NULL,`
	`  BINVAL  BLOB  NULL,`
	`CONSTRAINT RESEND_BIN FOREIGN KEY (BININDEX) REFERENCES bC_MESSAGES(BININDEX) ON DELETE CASCADE);`

Columns summary for the table `BC_RESEND_BIN` is shown in BC_RESEND_BIN: Columns Summary.

BC_RESEND_BIN: Columns Summary

Name	Data Type	Constraints	Nullable	Documentation
BININDEX	number(18)	PK	No	
COMPRESSED	varchar2(2)		Yes	
BINVAL	blob		Yes	

Constraints information for the table `BC_RESEND_BIN` is shown in BC_RESEND_BIN Constraints.

BC_RESEND_BIN Constraints

RESEND_BIN	
User IDLast Numeric Value	0
Check Constraint	`CONSTRAINT RESEND_BIN FOREIGN KEY (BININDEX) REFERENCES bC_MESSAGES (BININDEX) ON DELETE CASCADE)`

Relationship information for the table `BC_RESEND_BIN` is shown in BC_RESEND_BIN: Relationships.

BC_RESEND_BIN: Relationships

RESEND_BIN: Relationship	
To	BC_MESSAGES
User IDLast Numeric Value	0
Identifying	true
Subtype	false
On Delete	Cascade
To Multiplicity	1
From Multiplicity	1
Sync To Association	Yes
Data Model	Physical

BC_UACLOG

Details for the table BC_UACLOG are explained in BC_UACLOG: Details.

BC_UACLOG: Details

Name	Value
Data Model	Physical
Documentation	The table BC_UACLOG logs any edits that happen to the TIBCO BusinessConnect configuration done by TIBCO Administrator. The changes made by the user logged into the TIBCO Administrator UI are captured in this table. An audit trail for any changes to participants, business agreements, and operation editor is generated in this table. The tables BC_UACLOG and BC_UACLOG_DETAIL can be purged and archived based on the column OPERATION_TIME to be a part of the selection query. **Note**: This table will grow based on the modifications to participants, business agreements, and operations. It should be added as a part of the archiving process.
User IDLast Numeric Value	0

DDLClauses	`CREATE TABLE BC_UACLOG (`
	`ID VARCHAR(255) PRIMARY KEY,`
	`TRANSACTION_ID VARCHAR(128) NULL,`
	`OID VARCHAR(255) NULL,`
	`OBJ_NAME VARCHAR(255) NULL,`
	`OBJ_TYPE VARCHAR(64) NULL,`
	`USERNAME VARCHAR(128) NULL,`
	`USER_CATEGORY VARCHAR(128) NULL,`
	`USER_PARTNER_OID VARCHAR(255) NULL,`
	`OPERATION_ACTION VARCHAR(255) NULL,`
	`OPERATION_TARGET VARCHAR(500) NULL,`
	`OPERATION_TIME TIMESTAMP NULL,`
	`SESSION_ID VARCHAR(128) NULL,`
	`OWNER_OID VARCHAR(255) NULL,`
	`CONTEXT_LEVEL_OID VARCHAR(255) NULL,`
	`CONTEXT_LEVEL_NAME VARCHAR(255) NULL,`
	`CONTEXT_LEVEL_TYPE VARCHAR(255) NULL,`

Name	Value
	`IS_INTERNAL NUMBER(11) NULL` `);`

Columns summary for the table `BC_UACLOG` is shown in BC_UACLOG: Columns Summary.

BC_UACLOG: Columns Summary

Name	Data Type	Constraints	Nullable	Documentation
`ID`	varchar2(255)	PK	No	
`TRANSACTION_ID`	varchar2(128)		Yes	
`OID`	varchar2(255)		Yes	
OBJ_NAME	varchar2(255)		Yes	
OBJ_TYPE	varchar2(64)		Yes	
USERNAME	varchar2(128)		Yes	
USER_CATEGORY	varchar2(128)		Yes	
USER_PARTNER_OID	varchar2(255)		Yes	
OPERATION_ACTION	varchar2(255)		Yes	
OPERATION_TARGET	varchar2(500)		Yes	
OPERATION_TIME	timestamp		Yes	
SESSION_ID	varchar2(128)		Yes	

Name	Data Type	Constraints	Nullable	Documentation
OWNER_OID	varchar2(255)		Yes	
CONTEXT_LEVEL_OID	varchar2(255)		Yes	
CONTEXT_LEVEL_NAME	varchar2(255)		Yes	
CONTEXT_LEVEL_TYPE	varchar2(255)		Yes	
IS_INTERNAL	number(11)		Yes	

Relationship information for the table `BC_UACLOG` is shown in BC_UACLOG: Relationships.

BC_UACLOG: Relationships

UACLOG_DTL: Relationship	
To	BC_UACLOG_DETAIL
User IDLast Numeric Value	0
Identifying	true
Subtype	false
On Delete	Cascade
To Multiplicity	1..*
From Multiplicity	1
Sync To Association	Yes
Data Model	Physical

BC_LOGACL_TEMP

Details for the table BC_LOGACL_TEMP are explained in BC_LOGACL_TEMP: Details.

BC_LOGACL_TEMP: Details

Name	Value
Data Model	Physical
User IDLast Numeric Value	0
DDLClauses	`CREATE TABLE BC_LOGACL_TEMP ( SID VARCHAR(255) NOT NULL, PARTNER VARCHAR(255) NOT NULL, EXPIRE TIMESTAMP NOT NULL );`

Columns summary for the table BC_LOGACL_TEMP is shown in BC_LOGACL_TEMP: Columns Summary.

BC_LOGACL_TEMP: Columns Summary

Name	Data Type	Constraints	Nullable	Documentation
SID	varchar2(255)		No	
PARTNER	varchar2(255)		No	
EXPIRE	timestamp		No	

Indices information for the table BC_LOGACL_TEMP is shown in BC_LOGACL_TEMP Indices.

BC_LOGACL_TEMP Indices

BC_LOGACL_TEMP_TP	
User IDLast Numeric Value	0
Unique	false
Clustered	Unspecified

BC_DUP

Details for the table BC_DUP are explained in BC_DUP: Details.

BC_DUP: Details

Name	Value
Data Model	Physical
Documentation	This table stores the hash values of transactions used to indicate whether an incoming or outgoing transaction is a duplicate. The information stored is completely controlled by the individual business protocols. **Note**: This table will grow as transactions are exchanged. It should be added as a part of the archiving process.
User IDLast Numeric Value	0

DDLClauses	`CREATE TABLE BC_DUP (`
	`  PROTOCOL_NAME  VARCHAR2(32)  not null,`
	`  INSTALLATION_NAME  VARCHAR2(32)  not null,`
	`  PROTOCOL_VERSION  VARCHAR2(128)  not null,`
	`  FINGER_PRINT  VARCHAR2(32)  not null,`
	`  AUX1  VARCHAR2(255)  NULL,`
	`  AUX2  VARCHAR2(255)  NULL,`
	`  AUX3  VARCHAR2(255)  NULL,`

Name	Value

	<pre> AUX4 VARCHAR2(255) NULL,
	    AUX5 VARCHAR2(255) NULL,
	    TS TIMESTAMP DEFAULT
	SYSTIMESTAMP,
	    BINVAL BLOB NULL,
	CONSTRAINT BC_DUP_UNIQUE UNIQUE(PROTOCOL_NAME,
	INSTALLATION_NAME, FINGER_PRINT));</pre>

Columns summary for the table `BC_DUP` is shown in BC_DUP: Columns Summary.

BC_DUP: Columns Summary

Name	Data Type	Constraints	Nullable	Documentation
`PROTOCOL_NAME`	varchar2(32)		No	
`INSTALLATION_NAME`	varchar2(32)		No	
PROTOCOL_VERSION	varchar2(128)		No	
FINGER_PRINT	varchar2(32)		No	
AUX1	varchar2(255)		Yes	
AUX2	varchar2(255)		Yes	
AUX3	varchar2(255)		Yes	
AUX4	varchar2(255)		Yes	
AUX5	varchar2(255)		Yes	
TS	timestamp		No	
BINVAL	blob		Yes	

Constraints information for the table `BC_DUP` is shown in BC_DUP Constraints.

BC_DUP Constraints

BC_DUP_UNIQUE	
User IDLast Numeric Value	0

BC_UACLOG_DETAIL

Details for the table `BC_UACLOG_DETAIL` are explained in BC_UACLOG_DETAIL: Details.

BC_UACLOG_DETAIL: Details

Name	Value
`Data Model`	Physical
Documentation	**Note**: This table will grow based on the modifications to participants, business Agreements, and operations. It should be added as a part of the archiving process.
`User IDLast Numeric Value`	0
DDLClauses	`CREATE TABLE BC_UACLOG_DETAIL (ID VARCHAR(128) PRIMARY KEY, LOG_ID VARCHAR(255) NULL, PROPERTY VARCHAR(255) NULL, OLD_VALUE VARCHAR(1000) NULL, NEW_VALUE VARCHAR(1000) NULL, OPERATION_TIME TIMESTAMP NULL, PROTOCOL VARCHAR(255) NULL, PROPERTY_LABELKEY VARCHAR(255) NULL, OLD_VALUE_NAME VARCHAR(255) NULL, NEW_VALUE_NAME VARCHAR(255) NULL, CONSTRAINT UACLOG_DTL FOREIGN KEY (LOG_ID) REFERENCES BC_UACLOG (ID) ON DELETE CASCADE);`

Columns summary for the table `BC_UACLOG_DETAIL` is shown in BC_UACLOG_DETAIL: Columns Summary.

BC_UACLOG_DETAIL: Columns Summary

Name	Data Type	Constraints	Nullable	Documentation
ID	varchar2(128)	PK	No	
LOG_ID	varchar2(255)	PK/FK (BC_UACLOG.ID)	No	
PROPERTY	varchar2(255)		Yes	
OLD_VALUE	varchar2(1000)		Yes	
NEW_VALUE	varchar2(1000)		Yes	
OPERATION_TIME	timestamp		Yes	
PROTOCOL	varchar2(255)		Yes	
PROPERTY_LABELKEY	varchar2(255)		Yes	
OLD_VALUE_NAME	varchar2(255)		Yes	
NEW_VALUE_NAME	varchar2(255)		Yes	

Indices information for the table BC_UACLOG_DETAIL is shown in BC_UACLOG_DETAIL Indices.

BC_UACLOG_DETAIL Indices

BC_UACDETAIL_I	
User IDLast Numeric Value	0
Unique	false
Clustered	Unspecified

Constraints information for the table `BC_UACLOG_DETAIL` is shown in BC_UACLOG_DETAIL Constraints.

BC_UACLOG_DETAIL Constraints

UACLOG_DTL	
User IDLast Numeric Value	0
Check Constraint	`CONSTRAINT UACLOG_DTL FOREIGN KEY (LOG_ID) REFERENCES BC_UACLOG (ID) ON DELETE CASCADE`

Relationship information for the table `BC_UACLOG_DETAIL` is shown in BC_UACLOG: Relationships.

BC_UACLOG: Relationships

UACLOG_DTL: Relationship	
From	BC_UACLOG
User IDLast Numeric Value	0
Identifying	true
Subtype	false
On Delete	Cascade
To Multiplicity	1..*
From Multiplicity	1
Sync To Association	Yes
Data Model	Physical

BC_POLLER_INFO

Details for the table `BC_POLLER_INFO` are explained in BC_POLLER_INFO Details.

BC_POLLER_INFO Details

Name	Value
Data Model	Physical
Documentation	This table is used to store transient information populated by the TIBCO Administrator UI such as the Resend Actions or Queue Action. It does not grow and need not be a part of the archiving process.
User IDLast Numeric Value	0
Records	N/A

DDLClauses	`CREATE TABLE BC_POLLER_INFO (`
	`POLLER_TYPE VARCHAR2(15),`
	`BININDEX NUMBER(18),`
	`BC_TRANS_ID VARCHAR2(512),`
	`PROTOCOL_NAME VARCHAR2(32),`
	`AUX1 VARCHAR2(128),`
	`AUX2 VARCHAR2(128)`
	`);`

Columns summary for the table BC_POLLER_INFO is shown in BC_POLLER_INFO: Columns Summary.

BC_POLLER_INFO: Columns Summary

Name	Data Type	Constraints	Nullable	Documentation
POLLER_TYPE	varchar2(15)		No	
BININDEX	number(18)		No	
BC_TRANS_ID	varchar2(512)		No	
PROTOCOL_NAME	varchar2(32)		No	
AUX1	varchar2(128)		No	
AUX2	varchar2(128)		No	

Indices information for the table `BC_POLLER_INFO` is shown in BC_POLLER_INFO Indices and BC_POLLER_INFO Indices.

BC_POLLER_INFO Indices

BC_POLLER_INDX	
User IDLast Numeric Value	0
Unique	`false`
Clustered	Unspecified

BC_POLLER_INFO Indices

BC_POLLER_INDX_II	
User IDLast Numeric Value	0
Unique	`false`
Clustered	Unspecified

Non-Repudiation Schema Details

The table description for the non-repudiation schema is depicted in Non-Repudiation Schema Details.

Non-Repudiation Schema Details

Name	Documentation
`BC_NR_BIN_SIGNATURE`	This table holds the following information depending on whether the message is received from the Partner or if it is sent to the Partner from TIBCO BusinessConnect. If the message is sent from TIBCO BusinessConnect to the Partner or

Name	Documentation
	received from Partner, the BINVAL column stores the actual digest signature of the signed message. This table should be part of the Archiver strategy since the table can grow depending on the number of signed messages
BC_NR_BIN_EDCRED	This table holds the following information depending on whether the message is received from the Partner or if it is sent to the Partner from TIBCO BusinessConnect. If the message is sent from TIBCO BusinessConnect to the Partner, the BINVAL column stores the actual public certificate of the message used to encrypt the message. If it is the message received from the Partner, then BINVAL would store the Host's private key used to decrypt the encrypted payload. This table should be part of the Archiver strategy since the table can grow depending on the number of encrypted messages
BC_NR_BIN_SIGCRED	This table holds the following information depending on whether the message is received from the Partner or if it is sent to the Partner from TIBCO BusinessConnect. If the message is sent from TIBCO BusinessConnect to the Partner, the BINVAL column stores the actual private key of the message used to sign the message. If it is the message received from the Partner, then BINVAL would store the Partner's public certificate used to verify the signed payload. This table should be part of the Archiver strategy since the table can grow depending on the number of signed messages
BC_NR_TRANSACTIONS	BC_NR_TRANSACTIONS table stores the summary information of signed messages and the signed receipts that the Partner had sent or received during a transaction exchange. Only TIBCO BusinessConnect Protocols that support Non-Repudiation support would store information in this table. This table grows if the Business Protocol has Non-Repudiation enabled in

Name	Documentation
	their configurations and if TIBCO BusinessConnect sends and/or receives signed messages and signed receipts. This table needs to be part of the Archiving strategy. AUX column values are determined by the individual Business Protocol. For example, a search for Select * from BC_NR_TRANSACTIONS where STATUS like '%COMPLETED%' OR '%ERROR%' and TS > <from Date> can be used to query.
BC_NR_MESSAGES	This table has a many to 1 relationship with BC_NR_TRANSACTIONS table and stores entries whenever signed messages or receipts (or a encrypted-signed message and a signed receipt) is being exchanged with the Partner. This table grows along with the BC_NR_TRANSACTIONS table if the Business Protocol has Non-Repudiation enabled and the Partner exchanges at least signed messages requesting signed receipts. This table should be part of the Archiver strategy for the users. AUX column values are populated differently based on Business Protocols.
BC_NR_BIN_VALICERT	This table holds the following information depending on whether the message is received from the Partner or if it is sent to the Partner from TIBCO BusinessConnect. If the message is received from the Partner, then BINVAL would store the Partner's public certificate used to verify the signed payload. This table can grown depending on the number of signed messages used for verification and should be part of the Archiver strategy.
BC_NR_BIN	This table holds the actual signed message in the BINVAL column or the signed receipt sent or received from the Partner. This table has a foreign key to BC_NR_MESSAGES table to the BININDEX column and can grow based on the Non-repudiation messages sent and received from the Partner. This table needs to be part of the Archiver strategy: For e.g, To search for a message sent to a Partner via a User Transaction ID, you can create the following query:

Name	Documentation
	select * from BC_NR_BIN B where B.BININDEX in (select BININDEX from BC_NR_MESSAGES A where A.BC_TRANS_ID in (SELECT BC_TRANS_ID from BC_NR_TRANSACTIONS where TPNAME = <value> and USER_TRANS_ID = <value>));

BC_NR_BIN_SIGNATURE

Details for the table BC_NR_BIN_SIGNATURE are explained in BC_NR_BIN_SIGNATURE: Details.

BC_NR_BIN_SIGNATURE: Details

Name	Value
Data Model	Physical
Documentation	This table stores information that can be either received from the partner, or sent to the partner from TIBCO BusinessConnect. Whether messages are sent from TIBCO BusinessConnect to the partner or received from the partner, the BINVAL column stores the actual digest signature of the signed message. This table should be part of the archiving strategy since the table can grow large depending on the number of signed messages.
User IDLast Numeric Value	0

DDLClauses	`CREATE TABLE BC_NR_BIN_SIGNATURE (`
	`  BININDEX  NUMBER(18)  NOT NULL PRIMARY KEY,`
	`  COMPRESSED  VARCHAR2(2)  NULL,`
	`  BINVAL  BLOB  NULL,`
	`CONSTRAINT BC_NR_BIN_SIG_MID FOREIGN KEY (BININDEX) REFERENCES BC_NR_MESSAGES (BININDEX) ON DELETE CASCADE);`

Columns summary for the table BC_NR_BIN_SIGNATURE is shown in BC_NR_BIN_SIGNATURE: Columns Summary.

BC_NR_BIN_SIGNATURE: Columns Summary

Name	Data Type	Constraints	Nullable	Documentation
BININDEX	number(18)	PK/FK (BC_NR_MESSAGES.BININDEX)	No	
COMPRESSED	varchar2(2)		Yes	
BINVAL	blob		Yes	

Relationship information for the table BC_NR_BIN_SIGNATURE is shown in *BC_NR_BIN_SIGNATURE: Relationships*.

BC_NR_BIN_SIGNATURE: Relationships

BC_NR_BIN_SIG_MID: Relationship	
From	BC_NR_MESSAGES
User IDLast Numeric Value	0
Identifying	true
Subtype	false
On Delete	Cascade
On Update	Cascade
To Multiplicity	1
From Multiplicity	1
Sync To Association	Yes
Data Model	Physical

BC_NR_BIN_EDCRED

Details for the table BC_NR_BIN_EDCRED are explained in BC_NR_BIN_EDCRED: Details.

BC_NR_BIN_EDCRED: Details

Name	Value
Data Model	Physical
Documentation	This table stores information that can be either received from the partner, or sent to the partner from TIBCO BusinessConnect. • If the message was sent from TIBCO BusinessConnect to the partner, the BINVAL column stores the actual public certificate of the message used to encrypt the message. • If the message was received from the partner, then the column BINVAL stores the host's private key used to decrypt the encrypted payload. This table should be part of the archiving strategy since the table can grow large depending on the number of encrypted messages
User IDLast Numeric Value	0

DDLClauses	`CREATE TABLE BC_NR_BIN_EDCRED (`
	` BININDEX NUMBER(18) NOT NULL PRIMARY KEY,`
	` COMPRESSED VARCHAR2(2) NULL,`
	` BINVAL BLOB NULL,`
	`CONSTRAINT BC_NR_BIN_EDCRED_MID FOREIGN KEY (BININDEX)`
	`REFERENCES BC_NR_MESSAGES (BININDEX) ON DELETE CASCADE);`

Columns summary for the table BC_NR_BIN_EDCRED is shown in BC_NR_BIN_EDCRED: Columns Summary.

BC_NR_BIN_EDCRED: Columns Summary

Name	Data Type	Constraints	Nullable	Documentation
BININDEX	number(18)	PK/FK (BC_NR_MESSAGES.BININDEX)	No	

Name	Data Type	Constraints	Nullable	Documentation
COMPRESSED	varchar2(2)		Yes	
BINVAL	blob		Yes	

Relationship information for the table BC_NR_BIN_EDCRED is shown in *BC_NR_BIN_EDCRED: Relationships*.

BC_NR_BIN_EDCRED: Relationships

BC_NR_BIN_EDCRED_MID: Relationship	
From	BC_NR_MESSAGES
User IDLast Numeric Value	0
Identifying	true
Subtype	false
On Delete	Cascade
On Update	Cascade
To Multiplicity	1
From Multiplicity	1
Sync To Association	Yes
Data Model	Physical

BC_NR_BIN_SIGCRED

Details for the table BC_NR_BIN_SIGCRED are explained in BC_NR_BIN_SIGCRED: Details.

BC_NR_BIN_SIGCRED: Details

Name	Value
Data Model	Physical
Documentation	This table stores information that can be either received from the partner, or sent to the partner from TIBCO BusinessConnect. • If the message is sent from TIBCO BusinessConnect to the partner, the BINVAL column stores the actual private key of the message used to sign the message. • If the message was received from the Partner, then BINVAL would store the partner's public certificate used to verify the signed payload. This table should be part of the archiving strategy since the table can grow large depending on the number of signed messages
User IDLast Numeric Value	0

DDLClauses	`CREATE TABLE BC_NR_BIN_SIGCRED (`
	` BININDEX NUMBER(18) NOT NULL PRIMARY KEY,`
	` COMPRESSED VARCHAR2(2) NULL,`
	` BINVAL BLOB NULL,`
	`CONSTRAINT BC_NR_BIN_SIGCRED_MID FOREIGN KEY (BININDEX)`
	`REFERENCES BC_NR_MESSAGES (BININDEX) ON DELETE CASCADE);`

Columns summary for the table BC_NR_BIN_SIGCRED is shown in BC_NR_BIN_SIGCRED: Columns Summary.

BC_NR_BIN_SIGCRED: Columns Summary

Name	Data Type	Constraints	Nullable	Documentation
BININDEX	number (18)	PK/FK (BC_NR_MESSAGES.BININDEX)	No	A foreign key that refers to the column BC_NR_MESSAGES.BININDEX.
COMPRESSED	varchar2(2)		Yes	This column indicates

Name	Data Type	Constraints	Nullable	Documentation
				if the BINVAL column is compressed or not.
BINVAL	blob		Yes	Actual contents of either the private key or the public certificate.

Relationship information for the table BC_NR_BIN_SIGCRED is shown in *BC_NR_BIN_SIGCRED: Relationships*.

BC_NR_BIN_SIGCRED: Relationships

	BC_NR_BIN_EDCRED_MID: Relationship
From	BC_NR_MESSAGES
User IDLast Numeric Value	0
Identifying	true
Subtype	false
On Delete	Cascade
On Update	Cascade
To Multiplicity	1
From Multiplicity	1
Sync To Association	Yes
Data Model	Physical

BC_NR_TRANSACTIONS

Details for the table BC_NR_TRANSACTIONS are explained in BC_NR_TRANSACTIONS: Details.

BC_NR_TRANSACTIONS: Details

Name	Value
Data Model	Physical
Documentation	The table BC_NR_TRANSACTIONS stores the summary information of signed messages and the signed receipts that the partner had sent or received during a transaction exchange. Only the TIBCO BusinessConnect protocols that support non-repudiation store information in this table. This table grows if the business protocol has non-repudiation enabled in their configurations and if TIBCO BusinessConnect sends and/or receives signed messages and signed receipts. This table needs to be a part of the archiving strategy. AUX column values are determined by the individual business protocol; for example, a search for **Select * from BC_NR_TRANSACTIONS** where STATUS such as %COMPLETED% OR %ERROR% and TS > <from Date> can be used to query.
User IDLast Numeric Value	0
DDLClauses	CREATE TABLE BC_NR_TRANSACTIONS (BC_TRANS_ID VARCHAR2(512) NOT NULL PRIMARY KEY, USER_TRANS_ID VARCHAR2(512) NULL, PARENT_ID VARCHAR2(512) NULL, OPERATION_ID VARCHAR2(512) NULL, TPNAME VARCHAR2(128) NULL, TPDOMAIN VARCHAR2(32) NULL, TPID VARCHAR2(32) NULL, HOSTNAME VARCHAR2(128) NULL, HOSTDOMAIN VARCHAR2(32) NULL, HOSTID VARCHAR2(32) NULL, PROTOCOL_VERSION VARCHAR2(128) NULL, PROTOCOL_NAME VARCHAR2(32) NULL, INSTALLATION_NAME VARCHAR2(32) NULL, STATUS VARCHAR2(64) NULL, HOST_INITIATES VARCHAR2(5) NULL, USAGE_MODE VARCHAR2(16) NULL,

Name	Value

	<pre>GS_INSTANCE_INFO VARCHAR2(128) NULL,
	AUX1 VARCHAR2(512) NULL,
	AUX2 VARCHAR2(512) NULL,
	AUX3 VARCHAR2(512) NULL,
	AUX4 VARCHAR2(512) NULL,
	AUX5 VARCHAR2(512) NULL,
	AUX6 VARCHAR2(512) NULL,
	AUX7 VARCHAR2(512) NULL,
	AUX8 VARCHAR2(512) NULL,
	AUX9 VARCHAR2(512) NULL,
	STARTDATE TIMESTAMP DEFAULT SYSTIMESTAMP,
	TS TIMESTAMP DEFAULT SYSTIMESTAMP
	);</pre>

Columns summary for the table `BC_NR_TRANSACTIONS` is shown in BC_NR_TRANSACTIONS: Columns Summary.

BC_NR_TRANSACTIONS: Columns Summary

Name	Data Type	Constraints	Nullable	Documentation
BC_TRANS_ID	varchar2(512)	PK	No	Primary key column generated by TIBCO BusinessConnect.
USER_TRANS_ID	varchar2(512)		Yes	Column value specified from the private process to TIBCO BusinessConnect.
PARENT_ID	varchar2(512)		Yes	Not used
OPERATION_ID	varchar2(512)		Yes	Operation ID of the business protocol for a particular transaction.
TPNAME	varchar2(128)		Yes	Trading partner name to send to or to receive the non-repudiation transactions from.

Name	Data Type	Constraints	Nullable	Documentation
TPDOMAIN	varchar2(32)		Yes	Trading partner domain value (if the business protocol supports domain or ID metadata).
TPID	varchar2(32)		Yes	Trading partner ID value (if the business protocol supports domain or ID metadata).
HOSTNAME	varchar2(128)		Yes	Host participant name of the TIBCO BusinessConnect installation that sends or receives the non-repudiation transactions.
HOSTDOMAIN	varchar2(32)		Yes	Host domain value (if the business protocol supports domain or ID metadata).
HOSTID	varchar2(32)		Yes	Host ID value (if the business protocol supports domain or ID metadata).
PROTOCOL_ VERSION	varchar2(128)		Yes	Protocol version that is currently installed and enabled in the TIBCO BusinessConnect configuration.
PROTOCOL_NAME	varchar2(32)		Yes	Name of the protocol requiring the non-repudiation transfer and storage.

Name	Data Type	Constraints	Nullable	Documentation
INSTALLATION_NAME	varchar2(32)		Yes	Installation name of the TIBCO BusinessConnect configuration.
STATUS	varchar2(64)		Yes	Last known status value of the non-repudiation transaction.
HOST_INITIATES	varchar2(5)		Yes	This column indicates whether the transaction is received from the partner or sent to the partner.
USAGE_MODE	varchar2(16)		Yes	Not used
GS_INSTANCE_INFO	varchar2(128)		Yes	Specifies which Gateway Instance transferred this inbound message.
AUX1	varchar2(512)		Yes	
AUX2	varchar2(512)		Yes	
AUX3	varchar2(512)		Yes	
AUX4	varchar2(512)		Yes	
AUX5	varchar2(512)		Yes	
AUX6	varchar2(512)		Yes	
AUX7	varchar2(512)		Yes	
AUX8	varchar2(512)		Yes	
AUX9	varchar2(512)		Yes	

Name	Data Type	Constraints	Nullable	Documentation
STARTDATE	timestamp		Yes	Starting time of the non-repudiation transaction.
TS	timestamp		Yes	Current timestamp of the transaction that is being stored in this table .

Relationship information for the table `BC_NR_TRANSACTIONS` is shown in *BC_NR_TRANSACTIONS: Relationships*.

BC_NR_TRANSACTIONS: Relationships

	NR_MESSAGES: Relationship
To	BC_NR_MESSAGES
User IDLast Numeric Value	0
Identifying	false
Subtype	false
On Delete	Cascade
On Update	Cascade
To Multiplicity	0..*
From Multiplicity	1
Sync To Association	Yes
Data Model	Physical

BC_NR_MESSAGES

Details for the table BC_NR_MESSAGES are explained in BC_NR_MESSAGES: Details.

BC_NR_MESSAGES: Details

Name	Value
Data Model	Physical
Documentation	This table has a many-to-1 relationship with the table BC_NR_TRANSACTIONS and stores entries whenever signed messages or receipts (or encrypted-signed message and signed receipts) are exchanged with the partner. This table grows along with the table BC_NR_TRANSACTIONS if the business protocol has non-repudiation enabled and the partner exchanges at least signed messages requesting signed receipts. This table should be part of the archiving strategy for the users. Values of the AUX columns are populated differently based on business protocols.
User IDLast Numeric Value	0

DDLClauses	`CREATE TABLE BC_NR_MESSAGES(`
	` BC_TRANS_ID VARCHAR2(512) NOT NULL,`
	` BININDEX NUMBER(18) UNIQUE,`
	` USER_MESSAGE_ID VARCHAR2(512) NULL,`
	` AUX1 VARCHAR2(512) NULL,`
	` AUX2 VARCHAR2(512) NULL,`
	` AUX3 VARCHAR2(512) NULL,`
	` AUX4 VARCHAR2(512) NULL,`
	` AUX5 VARCHAR2(512) NULL,`
	` TS TIMESTAMP DEFAULT SYSTIMESTAMP,`
	`CONSTRAINT NR_MESSAGES FOREIGN KEY (BC_TRANS_ID) references BC_NR_TRANSACTIONS(BC_TRANS_ID) ON DELETE CASCADE);`

Columns summary for the table BC_NR_MESSAGES is shown in BC_NR_MESSAGES: Columns Summary.

BC_NR_MESSAGES: Columns Summary

Name	Data Type	Constraints	Nullable	Documentation
BC_TRANS_ID	varchar2(512)	FK (BC_NR_TRANSACTIONS.BC_TRANS_ID)	No	Foreign key to BC_NR_TRANSACTIONS.BC_TRANS_ID
BININDEX	number(18)	Unique	Yes	Unique primary key for this table that is generated by TIBCO BusinessConnect
USER_MESSAGE_ID	varchar2(512)		Yes	User transaction ID populated from the private process sent to TIBCO BusinessConnect.
AUX1	varchar2(512)		Yes	
AUX2	varchar2(512)		Yes	
AUX3	varchar2(512)		Yes	
AUX4	varchar2(512)		Yes	
AUX5	varchar2(512)		Yes	
TS	timestamp		Yes	

Indices information for the table BC_NR_MESSAGES is shown in BC_NR_MESSAGES: Indices.

BC_NR_MESSAGES: Indices

BC_NR_MESSAGES_IDX	
User IDLast Numeric Value	0
Unique	false

BC_NR_MESSAGES_IDX	
Index Name Pattern	{table_name}
Clustered	Non-clustered

Relationship information for the table `BC_NR_MESSAGES` is shown in *BC_NR_MESSAGES Relationships: BC_NR_BIN_MID*, *BC_NR_MESSAGES Relationships: BC_NR_BIN_SIG_MID*, *BC_NR_MESSAGES Relationships: BC_NR_BIN_VALICERT_MID*, *BC_NR_MESSAGES Relationships: BC_NR_BIN_SIGCRED_MID* , *BC_NR_MESSAGES Relationships: BC_NR_BIN_EDCRED_MID*, *and BC_NR_MESSAGES Relationships: NR_MESSAGES*

BC_NR_MESSAGES Relationships: BC_NR_BIN_MID

BC_NR_BIN_MID: Relationship	
To	BC_NR_BIN
User IDLast Numeric Value	0
Identifying	true
Subtype	false
On Delete	Cascade
On Update	Cascade
To Multiplicity	1
From Multiplicity	1
Sync To Association	Yes
Data Model	Physical

BC_NR_MESSAGES Relationships: BC_NR_BIN_SIG_MID

BC_NR_BIN_SIG_MID: Relationship	
To	BC_NR_BIN_SIGNATURE
User IDLast Numeric Value	0
Identifying	true
Subtype	false
On Delete	Cascade
On Update	Cascade
To Multiplicity	1
From Multiplicity	1
Sync To Association	Yes
Data Model	Physical

BC_NR_MESSAGES Relationships: BC_NR_BIN_VALICERT_MID

BC_NR_BIN_VALICERT_MID: Relationship	
To	BC_NR_BIN_VALICERT
User IDLast Numeric Value	0
Identifying	true
Subtype	false
On Delete	Cascade

BC_NR_BIN_VALICERT_MID: Relationship	
On Update	Cascade
To Multiplicity	1
From Multiplicity	1
Sync To Association	Yes
Data Model	Physical

.

BC_NR_MESSAGES Relationships: BC_NR_BIN_SIGCRED_MID

BC_NR_BIN_SIGCRED_MID: Relationship	
To	BC_NR_BIN_SIGCRED
User IDLast Numeric Value	0
Identifying	true
Subtype	false
On Delete	Cascade
On Update	Cascade
To Multiplicity	1
From Multiplicity	1
Sync To Association	Yes
Data Model	Physical

BC_NR_MESSAGES Relationships: BC_NR_BIN_EDCRED_MID

BC_NR_BIN_EDCRED_MID: Relationship	
To	BC_NR_BIN_EDCRED
User IDLast Numeric Value	0
Identifying	true
Subtype	false
On Delete	Cascade
On Update	Cascade
To Multiplicity	1
From Multiplicity	1
Sync To Association	Yes
Data Model	Physical

BC_NR_MESSAGES Relationships: NR_MESSAGES

NR_MESSAGES: Relationship	
To	BC_NR_TRANSACTIONS
User IDLast Numeric Value	0
Identifying	false
Subtype	false
On Delete	Cascade

NR_MESSAGES: Relationship	
On Update	Cascade
To Multiplicity	0..*
From Multiplicity	1
Sync To Association	Yes
Data Model	Physical

BC_NR_BIN_VALICERT

Details for the table `BC_NR_BIN_VALICERT` are explained in BC_NR_BIN_VALICERT: Details.

BC_NR_BIN_VALICERT: Details

Name	Value
`Data Model`	Physical
`Documentation`	This table stores information that can be either received from the partner, or sent to the partner from TIBCO BusinessConnect. If the message is received from the partner, then `BINVAL` stores the partner's public certificate used to verify the signed payload. This table can grown depending on the number of signed messages used for verification and should be a part of the archiving strategy.
`User IDLast Numeric Value`	0

`DDLClauses`	`CREATE TABLE BC_NR_BIN_VALICERT (`
	` BININDEX NUMBER(18) NOT NULL PRIMARY KEY,`
	` COMPRESSED VARCHAR2(2) NULL,`
	` BINVAL BLOB NULL,`
	`CONSTRAINT BC_NR_BIN_VALICERT_MID FOREIGN KEY (BININDEX) REFERENCES`
	`BC_NR_MESSAGES (BININDEX) ON DELETE CASCADE);`

Columns summary for the table `BC_NR_BIN_VALICERT` is shown in BC_NR_BIN_VALICERT: Columns Summary.

BC_NR_BIN_VALICERT: Columns Summary

Name	Data Type	Constraints	Nullable	Documentation
`BININDEX`	number(18)	`PK/FK (BC_NR_MESSAGES.BININDEX)`	No	
`COMPRESSED`	varchar2(2)		Yes	
`BINVAL`	blob		Yes	

Relationship information for the table `BC_NR_BIN_VALICERT` is shown in *BC_NR_BIN_VALICERT: Relationships*.

BC_NR_BIN_VALICERT: Relationships

	BC_NR_BIN_EDCRED_MID: Relationship
From	BC_NR_MESSAGES
User IDLast Numeric Value	0
Identifying	true
Subtype	false
On Delete	Cascade
On Update	Cascade
To Multiplicity	1
From Multiplicity	1
Sync To Association	Yes
Data Model	Physical

BC_NR_BIN

Details for the table BC_NR_BIN are explained in BC_NR_BIN: Details.

BC_NR_BIN: Details

Name	Value
Data Model	Physical
Documentation	This table stores in the BINVAL column the actual signed messages, or the signed receipt sent or received from the partner. It has a foreign key to the table BC_NR_MESSAGES to the BININDEX column and can grow large based on the non-repudiation messages sent and received from the partner. This table needs to be part of the archiving strategy; for example, to search for a message sent to a partner via a user transaction ID, you can create the following query: **select * from BC_NR_BIN B where B.BININDEX in (select BININDEX from BC_NR_MESSAGES A where A.BC_TRANS_ID in (SELECT BC_TRANS_ID from BC_NR_TRANSACTIONS where TPNAME = <value> and USER_TRANS_ID = <value>));**
User IDLast Numeric Value	0

DDLClauses	`CREATE TABLE BC_NR_BIN (`
	` BININDEX NUMBER(18) NOT NULL PRIMARY KEY,`
	` BINID VARCHAR2(128) NULL,`
	` COMPRESSED VARCHAR2(2) NULL,`
	` BINVAL BLOB NULL,`
	`CONSTRAINT BC_NR_BIN_MID FOREIGN KEY (BININDEX) REFERENCES BC_NR_MESSAGES(BININDEX) ON DELETE CASCADE);`

Columns summary for the table BC_NR_BIN is shown in BC_NR_BIN: Columns Summary.

BC_NR_BIN: Columns Summary

Name	Data Type	Constraints	Nullable	Documentation
BININDEX	number(18)	PK/FK (BC_NR_MESSAGES.BININDEX)	No	Foreign index to the column BC_NR_MESSAGES BININDEX.
BINID	varchar2 (128)		Yes	
COMPRESSED	varchar2(2)		Yes	This column shows whether the payload is compressed or not.
BINVAL	blob		Yes	This column shows the actual signed message or receipt.

Relationship information for the table BC_NR_BIN is shown in *BC_NR_BIN: Relationships: BC_NR_BIN_MID*.

BC_NR_BIN: Relationships: BC_NR_BIN_MID

BC_NR_BIN_MID: Relationship	
From	BC_NR_MESSAGES
User IDLast Numeric Value	0
Identifying	true
Subtype	false
On Delete	Cascade

BC_NR_BIN_MID: Relationship	
On Update	Cascade
To Multiplicity	1
From Multiplicity	1
Sync To Association	Yes
Data Model	Physical

Runtime Schema Details

Runtime table schemas are mostly transient tables except for the tables BC_SCHEDULED_TASK . Consequently, only the table BC_SCHEDULED_TASK needs to be archived, as well as any protocol tables where it is specifically indicated that they need to be archived.

The table description for the runtime schema is depicted in Runtime Schema Details.

Runtime Schema Details

Name	Documentation
BC_SFWS_ATTACHMENTS_BIN	This table contains the actual transaction payload for the PartnerExpress or FTP Server to download. The payload is stored encrypted, and the data is removed once the partner downloads it, or if the transaction times out. This table need not be part of archiving process, and sizing of the table has to be considered when large amount of partner data is stored in it.
BC_SFWS_MESSAGES	This table stores transient transaction details of a business protocol to be retrieved through PartnerExpress or FTP Server. The actual data is stored in the table BC_SFWS_ATTACHMENTS_BIN with the trading partner information stored in the table BC_SFWS_TPINFO. The entries in the table are time-bound; they would expire if the

Name	Documentation
	transactions are not downloaded by the partner, and would be deleted. If the transactions are downloaded by the partner within the time, they get deleted as well. Therefore, there is no need to prepare this table for archiving. Make sure that this table along with `BC_SFWS_TPINFO` and `BC_SFWS_ATTACHMENTS_BIN` has sufficient storage space when configuring for your partner data, since the storage can grow quickly if the number of partners is big and so is the number of transactions.
`BC_SCHEDULED_TASK`	This table is used for several scenarios and is used in following ways: • Some portion of the data is transient, such as when Gateway Instances are registered, marked for running, and need not be archived. However, this doesn't apply when protocol pollers get registered and become transient (such as in the case of tibEDI plug-in). • This table is also used to store transaction data in conjunction with the business protocol tables (in the case of tibEDI), when the transactions are scheduled or batch at certain time. This data is never deleted and will be growing permanently; it should be archived so that it does not run out of table space. • `Completed` or `Error` status data for the business protocols can be archived. • The `AUX` column values are dependent on the protocol usage.
`BC_SFWS_TPINFO`	This table holds information about the partner and host exchanging transaction documents in the PartnerExpress or FTP Server installation. This table is a transient table and need not be prepared for archiving. However, proper sizing is needed to store large number of row data if the partner as well as transaction volume is huge.
`BC_MDN`	This table stores information that is required for the transport level receipts (Message Disposition Notification) to be received from the partner. It is used when protocols use public transports such as

Name	Documentation
	Email, AS1, and AS2, or transports that require receipts. This is a transient table where the row data gets deleted once the receipt arrives, or when a timeout occurs when the receipt does not arrive within the time specified by the transport. It need not be a part of the archiving strategy.
BC_HIBERNATION	This table stores transient information to be retrieved when receiving a response from the Partner or the private process. This table data is used by almost all business protocols that support Synchronous and Asynchronous Request Response operations. This table is used in conjunction with other table data, such as when using it with the AS2, AS1, and Email receipts for the table BC_MDN, or as a part of marking the transactions as overdue when the messages are stored for PartnerExpress or FTP Server in the tables BC_SFWS_MESSAGES. This table can grow large depending on the number of transactions, but the data gets cleared when there are timeouts, or when the responses or appropriate handlers take care of retrieving and removing the row data. This table need not be part of the archiving process; however, it should be considered when doing database sizing if there is a big number of transactions transferred through TIBCO BusinessConnect. The AUX columns are determined by either the business protocols, or the handlers using it.
BC_	This table is used to store user saved queries from the TIBCO Administrator UI, or queries saved from PartnerExpress. The table value does not grow as much as the transaction audit trail table; it grows only when a big number of users save queries. This table need not be considered for archiving.
BC_HIBERNATION_BIN	This table holds additional context information about the hibernation entries that are stored, such as information how to collate the audit log when a response is received.

Name	Documentation
	This table is transient like its parent and need not be considered for archiving.
BC_LOGQUERYBIN	This table contains the actual query saved as a blob, which has a foreign key to the table BC_ . This table need not be considered for archiving.

BC_SFWS_ATTACHMENTS_BIN

Details for the table BC_SFWS_ATTACHMENTS_BIN are explained in BC_SFWS_ATTACHMENTS_BIN: Details.

Note	If you are using the Oracle 11g Release 1 driver for the Oracle 11g Release 1 database, you will experience SQL errors when using BC_SFWS tables. To avoid these errors, use the Oracle 11g Release 2 driver.

BC_SFWS_ATTACHMENTS_BIN: Details

Name	Value
Data Model	Physical
Documentation	This table contains the actual transaction payload for the PartnerExpress or FTP Server to download. The payload is stored encrypted, and the data is removed once the partner downloads it, or once the transaction times out. This table need not be archived, but sizing of the table has to be considered when a large amount of partner data is stored.
User IDLast Numeric Value	0

Name	Value
DDLClauses	CREATE TABLE BC_SFWS_ATTACHMENTS_BIN (TRANS_ID VARCHAR2(512) NOT NULL , ATTACHMENT_ID VARCHAR2(512) NOT NULL PRIMARY KEY, COMPRESSED VARCHAR2(2) NULL, DATA_ENCODING VARCHAR2(60) NULL, SHARED_KEY VARCHAR2(255) NULL, PWD_PROTECT VARCHAR2(2) NULL, BINVAL BLOB NULL, PAYLOAD_SIZE NUMBER(10) NULL, STREAM_SIZE NUMBER(18) NULL, ATTACHMENT_ORDER NUMBER(10) NULL, CONTENT_TYPE VARCHAR2(512) NULL, CONTENT_ID VARCHAR2(512) NULL, AUX1 VARCHAR2(512) NULL, AUX2 VARCHAR2(512) NULL, AUX3 VARCHAR2(512) NULL, AUX4 VARCHAR2(512) NULL, AUX5 VARCHAR2(512) NULL, AUX6 VARCHAR2(512) NULL, AUX7 VARCHAR2(512) NULL, AUX8 VARCHAR2(512) NULL, AUX9 VARCHAR2(512) NULL, CONSTRAINT ATTACHMENTS FOREIGN KEY (TRANS_ID) REFERENCES BC_SFWS_MESSAGES(TRANS_ID) ON DELETE CASCADE);

Columns summary for the table BC_SFWS_ATTACHMENTS_BIN is shown in BC_SFWS_ATTACHMENTS_BIN: Columns Summary.

BC_SFWS_ATTACHMENTS_BIN: Columns Summary

Name	Data Type	Constraints	Nullable	Documentation
TRANS_ID	varchar2 (512)	FK (BC_SFWS_MESSAGES.TRANS_ID)	No	Constraint to the column BC_SFWS_MESSAGES.TRANS_ID
ATTACHMENT_ID	varchar2 (512)	PK	No	It is generated by

Name	Data Type	Constraints	Nullable	Documentation
				TIBCO BusinessConnect and is the primary key.
COMPRESSED	varchar2(2)		Yes	Indicates whether the payload is compressed or not.
DATA_ENCODING	varchar2(60)		Yes	Encoding type (not used for now)
SHARED_KEY	varchar2 (255)		Yes	Not used
PWD_PROTECT	varchar2(2)		Yes	Indicates whether the payload is encrypted or not.
BINVAL	blob		Yes	Encrypted payload
PAYLOAD_SIZE	number(10)		Yes	Payload size
STREAM_SIZE	number(18)		Yes	Stream size
ATTACHMENT_ ORDER	number(10)		Yes	Not used
CONTENT_TYPE	varchar2 (512)		Yes	For future use
CONTENT_ID	varchar2 (512)		Yes	For future use
AUX1	varchar2 (512)		Yes	
AUX2	varchar2		Yes	

Name	Data Type	Constraints	Nullable	Documentation
	(512)			
AUX3	varchar2 (512)		Yes	
AUX4	varchar2 (512)		Yes	
AUX5	varchar2 (512)		Yes	
AUX6	varchar2 (512)		Yes	
AUX7	varchar2 (512)		Yes	
AUX8	varchar2 (512)		Yes	
AUX9	varchar2 (512)		Yes	

Indices information for the table `BC_SFWS_ATTACHMENTS_BIN` is shown in BC_SFWS_ATTACHMENTS_BIN Indices: SFWS_ATTACH_IDX and BC_SFWS_ATTACHMENTS_BIN Indices: SFWS_ATTACH_CRIT.

BC_SFWS_ATTACHMENTS_BIN Indices: SFWS_ATTACH_IDX

SFWS_ATTACH_IDX	
User IDLast Numeric Value	0
Unique	false
Index Name Pattern	{table_name}
Clustered	Non-clustered

BC_SFWS_ATTACHMENTS_BIN Indices: SFWS_ATTACH_CRIT

SFWS_ATTACH_CRIT	
User IDLast Numeric Value	0
Unique	false
Index Name Pattern	{table_name}
Clustered	Non-clustered

Relationship information for the table `BC_SFWS_ATTACHMENTS_BIN` is shown in *BC_SFWS_ATTACHMENTS_BIN: Relationships*.

BC_SFWS_ATTACHMENTS_BIN: Relationships

ATTACHMENTS: Relationship	
From	BC_SFWS_MESSAGES
User IDLast Numeric Value	0
Identifying	false
Subtype	false
On Delete	Cascade
On Update	Cascade
To Multiplicity	0..*
From Multiplicity	1
Sync To Association	Yes
Data Model	Physical

BC_SFWS_MESSAGES

Details for the table `BC_SFWS_MESSAGES` are explained in BC_SFWS_MESSAGES: Details.

Note If you are using the Oracle 11g Release 1 driver for the Oracle 11g Release 1 database, you will experience SQL errors when using `BC_SFWS` tables. To avoid these errors, use the Oracle 11g Release 2 driver.

BC_SFWS_MESSAGES: Details

Name	Value
Data Model	Physical
Documentation	This table stores transient transaction details of a business protocol to be retrieved through PartnerExpress or FTP Server. The actual data is stored in the table `BC_SFWS_ATTACHMENTS_BIN`, with the trading partner information stored in the table `BC_SFWS_TPINFO`. The entries in the table are time-bound; they will expire if the transactions are not downloaded by the partner and will be deleted. If the transactions are downloaded by the partner within the time, they will be deleted as well. Therefore, there is no need to prepare this table for archiving. Make sure that this table along with the tables `BC_SFWS_TPINFO` and `BC_SFWS_ATTACHMENTS_BIN` has sufficient storage space when preparing for your partner data, since the storage can grow quickly if the number of partners is big and so is the number of transactions.
User IDLast Numeric Value	0

DDLClauses	`CREATE TABLE BC_SFWS_MESSAGES (`
	`TRANS_ID VARCHAR2(512) NOT NULL PRIMARY KEY,`
	`USER_TRANS_ID VARCHAR2(512) NULL,`

Name	Value

```
      OPERATION_ID       VARCHAR2(512)   NULL,
      OPERATION_TYPE     VARCHAR2(512)   NULL,
      PROTOCOL_VERSION   VARCHAR2(128)   NULL,
      PROTOCOL_NAME      VARCHAR2(32)    NULL,
      CORRELATION_ID     VARCHAR2(64)    NULL,
      FLOW_DIRECTION     VARCHAR2(32)    NULL,
      ACTION_TYPE        VARCHAR2(64)    NULL,
      STATUS             VARCHAR2(64)    NULL,
             AUX1               VARCHAR2(512)   NULL,
      AUX2               VARCHAR2(512)   NULL,
      AUX3               VARCHAR2(512)   NULL,
      AUX4               VARCHAR2(512)   NULL,
      AUX5               VARCHAR2(512)   NULL,
      AUX6               VARCHAR2(512)   NULL,
      AUX7               VARCHAR2(512)   NULL,
      AUX8               VARCHAR2(512)   NULL,
      AUX9               VARCHAR2(512)   NULL,
      TS                 TIMESTAMP DEFAULT
SYSTIMESTAMP
);
```

Columns summary for the table `BC_SFWS_MESSAGES` is shown in BC_SFWS_ATTACHMENTS_BIN: Columns Summary.

BC_SFWS_MESSAGES: Columns Summary

Name	Data Type	Constraints	Nullable	Documentation
TRANS_ID	varchar2(512)	PK	No	Uniquely generated ID by TIBCO BusinessConnect.
USER_TRANS_ID	varchar2(512)		Yes	User transaction ID specified from the private process.
OPERATION_ID	varchar2(512)		Yes	Operation ID of the transaction sent from the private process.

Name	Data Type	Constraints	Nullable	Documentation
OPERATION_TYPE	varchar2(512)		Yes	Operation type of the transaction, such as Notify or Async
PROTOCOL_ VERSION	varchar2(128)		Yes	Protocol version that is used.
PROTOCOL_NAME	varchar2(32)			Business protocol used to store data for the partner.
CORRELATION_ID	varchar2(64)			Currently not used
FLOW_DIRECTION	varchar2(32)			Indicates whether direction is inbound or outbound.
ACTION_TYPE	varchar2(64)			Request, Response or Notify type.
STATUS	varchar2(64)			Current status of the transaction.
AUX1	varchar2(512)		Yes	
AUX2	varchar2(512)		Yes	
AUX3	varchar2(512)		Yes	
AUX4	varchar2(512)		Yes	
AUX5	varchar2(512)		Yes	
AUX6	varchar2(512)		Yes	
AUX7	varchar2(512)		Yes	

Name	Data Type	Constraints	Nullable	Documentation
AUX8	varchar2(512)		Yes	
AUX9	varchar2(512)		Yes	
TS	timestamp		Yes	

Indices information for the table `BC_SFWS_MESSAGES` is shown in BC_SFWS_MESSAGES Indices: SFWS_MSGS_UTXID, BC_SFWS_MESSAGES Indices: SFWS_MSG_TOP, and BC_SFWS_MESSAGES Indices: SFWS_MSG_AUX1.

BC_SFWS_MESSAGES Indices: SFWS_MSGS_UTXID

SFWS_MSGS_UTXID	
User IDLast Numeric Value	0
Unique	false
Index Name Pattern	{table_name}
Clustered	Non-clustered

BC_SFWS_MESSAGES Indices: SFWS_MSG_TOP

SFWS_MSG_TOP	
User IDLast Numeric Value	0
Unique	false
Index Name Pattern	{table_name}
Clustered	Non-clustered

BC_SFWS_MESSAGES Indices: SFWS_MSG_AUX1

SFWS_MSG_AUX1	
User IDLast Numeric Value	0
Unique	false
Index Name Pattern	{table_name}
Clustered	Non-clustered

Relationship information for the table `BC_SFWS_MESSAGES` is shown in *BC_SFWS_MESSAGES Relationships: BC_SFWS_TPS* and *BC_SFWS_MESSAGES Relationships: ATTACHMENTS*.

BC_SFWS_MESSAGES Relationships: BC_SFWS_TPS

BC_SFWS_TPS: Relationship	
To	BC_SFWS_TPINFO
User IDLast Numeric Value	0
Identifying	false
Subtype	false
On Delete	Cascade
On Update	Cascade
To Multiplicity	0..*
From Multiplicity	1
Sync To Association	Yes
Data Model	Physical

BC_SFWS_MESSAGES Relationships: ATTACHMENTS

ATTACHMENTS: Relationship	
To	BC_SFWS_ATTACHMENTS_BIN
User IDLast Numeric Value	0
Identifying	false
Subtype	false
On Delete	Cascade
On Update	Cascade
To Multiplicity	0..*
From Multiplicity	1
Sync To Association	Yes
Data Model	Physical

BC_SCHEDULED_TASK

Details for the table `BC_SCHEDULED_TASK` are explained in BC_SCHEDULED_TASK: Details.

BC_SCHEDULED_TASK: Details

Name	Value
`Data Model`	Physical
`Documentation`	This table is used for several scenarios and is behaves as follows: • Some portion of the data is transient, such as when Gateway

Name	Value
	Instances are registered, marked for running, and need not be archived. This does not apply when protocol pollers are registered and becomes transient in the case of the tibEDI plug-in. • This table is also used to store transaction data in conjunction with the business protocol tables (in the case of tibEDI) , when the transactions are scheduled or batch at certain time. This data is never deleted and will be growing permanently. Therefore, it should be archived in order not to run out of table space. • `Completed` or `Error` status data for the business protocols can be archived. • AUX column values are dependent on the protocol usage.
`User IDLast Numeric Value`	0

`DDLClauses`	`CREATE TABLE BC_SCHEDULED_TASK (`
	`TASKID VARCHAR2(255) PRIMARY KEY,`
	`PROTOCOLNAME VARCHAR2(32),`
	`PROTOCOLVERSION VARCHAR2(32) ,`
	`INSTALLATIONNAME VARCHAR2(32),`
	`TASKNAME VARCHAR2(512) ,`
	`TASKTYPE VARCHAR2(32) ,`
	`TASKINFO VARCHAR2(512) ,`
	`HOSTNAME VARCHAR2(128) ,`
	`TPNAME VARCHAR2(128) ,`
	`SCHEDULERINFO VARCHAR2(128) ,`
	`STATUS VARCHAR2(64) ,`
	`NEXTSTARTTIME TIMESTAMP DEFAULT SYSTIMESTAMP,`
	`MAXNUMTX NUMBER(18) ,`
	`TXCOUNT NUMBER(18) ,`
	`TS TIMESTAMP DEFAULT SYSTIMESTAMP,`
	`HANDLER VARCHAR2(512) ,`
	`LOGCONTEXTID VARCHAR2(512) ,`

Name	Value

	<pre> AUX1 VARCHAR2(512) ,
	    AUX2 VARCHAR2(512) ,
	    AUX3 VARCHAR2(512) ,
	    AUX4 VARCHAR2(512) ,
	    AUX5 VARCHAR2(512) ,
	    AUX6 VARCHAR2(512) ,
	    AUX7 VARCHAR2(512) ,
	    AUX8 VARCHAR2(512) ,
	    AUX9 VARCHAR2(512) ,
	    AUX10 VARCHAR2(512) ,
	    AUX11 VARCHAR2(512) ,
	    AUX12 VARCHAR2(512) ,
	    AUX13 VARCHAR2(512) ,
	    AUX14 VARCHAR2(512) ,
	    AUX15 VARCHAR2(512)
	);</pre>

Columns summary for the table BC_SCHEDULED_TASK is shown in BC_SCHEDULED_TASK: Columns Summary.

BC_SCHEDULED_TASK: Columns Summary

Name	Data Type	Constraints	Nullable	Documentation
TRANS_ID	varchar2(255)	PK	No	Uniquely generated Id by TIBCO BusinessConnect.
PROTOCOLNAME	varchar2(32)		Yes	Business protocol information
PROTOCOLVERSION	varchar2(32)		Yes	Protocol version
INSTALLATIONNAME	varchar2(32)		Yes	TIBCO BusinessConnect installation name
TASKNAME	varchar2(64)		Yes	Internal use

Name	Data Type	Constraints	Nullable	Documentation
TASKTYPE	varchar2(32)		Yes	Internal use
TASKINFO	varchar2(512)		Yes	Internal use
HOSTNAME	varchar2(128)		Yes	Host name of the participant
TPNAME	varchar2(128)		Yes	Partner name
SCHEDULERINFO	varchar2(128)		Yes	Scheduling information
STATUS	varchar2(64)		Yes	Status of the scheduled transactions.
NEXTSTARTTIME	timestamp		Yes	Start time of the scheduled transactions.
MAXNUMTX	number(18)		Yes	Maximal number of transactions in a batch or queue.
TXCOUNT	number(18)		Yes	Current number of transactions in a batch (in case of tibEDI)
TS	timestamp		Yes	
HANDLER	varchar2(512)		Yes	
LOGCONTEXTID	varchar2(512)		Yes	
AUX1	varchar2(512)		Yes	

Name	Data Type	Constraints	Nullable	Documentation
AUX2	varchar2(512)		Yes	
AUX3	varchar2(512)		Yes	
AUX4	varchar2(512)		Yes	
AUX5	varchar2(512)		Yes	
AUX6	varchar2(512)		Yes	
AUX7	varchar2(512)		Yes	
AUX8	varchar2(512)		Yes	
AUX9	varchar2(512)		Yes	
AUX10	varchar2(512)		Yes	
AUX11	varchar2(512)		Yes	
AUX12	varchar2(512)		Yes	
AUX13	varchar2(512)		Yes	
AUX14	varchar2(512)		Yes	
AUX15	varchar2(512)		Yes	

Indices information for the table `BC_SCHEDULED_TASK` is shown in BC_SCHEDULED_TASK Indices: BC_SCHEDULED_TASK_INDX_B and BC_SCHEDULED_TASK Indices: BC_SCHEDULED_TASK_INDX_C.

BC_SCHEDULED_TASK Indices: BC_SCHEDULED_TASK_INDX_B

BC_SCHEDULED_TASK_INDX_B	
User IDLast Numeric Value	0

BC_SCHEDULED_TASK_INDX_B	
Unique	false
Index Name Pattern	{table_name}
Clustered	Non-clustered

BC_SCHEDULED_TASK Indices: BC_SCHEDULED_TASK_INDX_C

BC_SCHEDULED_TASK_INDX_C	
User IDLast Numeric Value	0
Unique	false
Index Name Pattern	{table_name}
Clustered	Non-clustered

BC_SFWS_TPINFO

Details for the table `BC_SFWS_TPINFO` are explained in BC_SFWS_TPINFO: Details.

Note If you are using the Oracle 11g Release 1 driver for the Oracle 11g Release 1 database, you will experience SQL errors when using `BC_SFWS` tables. To avoid these errors, use the Oracle 11g Release 2 driver.

BC_SFWS_TPINFO: Details

Name	Value
Data Model	Physical
Documentation	This table holds information about the partner and host that are exchanging transaction documents in the installation of the PartnerExpress or FTP Server. It is a transient table and need not be archived. However, proper sizing must be done for storing large number of row data if the partner data and transaction volume are large.
User IDLast Numeric Value	0

DDLClauses	`CREATE TABLE BC_SFWS_TPINFO (`
	`TRANS_ID VARCHAR2(512) NOT NULL,`
	`TPINFO_ID VARCHAR2(512) NOT NULL PRIMARY KEY,`
	`TPNAME VARCHAR2(255) NULL,`
	`TPDOMAIN VARCHAR2(255) NULL,`
	`TPID VARCHAR2(255) NULL,`
	`HOSTNAME VARCHAR2(255) NULL,`
	`HOSTDOMAIN VARCHAR2(255) NULL,`
	`HOSTID VARCHAR2(255) NULL,`
	`STATUS VARCHAR2(64) NULL,`
	`AUX1 VARCHAR2(512) NULL,`
	`AUX2 VARCHAR2(512) NULL,`
	`AUX3 VARCHAR2(512) NULL,`
	`AUX4 VARCHAR2(512) NULL,`
	`AUX5 VARCHAR2(512) NULL,`
	`RTN_RECEIPT VARCHAR2(255) NULL,`
	`TS TIMESTAMP DEFAULT SYSTIMESTAMP,`
	`EXPIRATION TIMESTAMP NULL,`
	`MAXMSGCOUNT NUMBER(18) NOT NULL,`
	`CURMSGCOUNT NUMBER(18) NOT NULL,`
	`EXTRA_INFO BLOB NULL,`
	`AUDITLOG_CONTEXTID VARCHAR2(100) NULL,`
	`MARKED_DEL NUMBER(2) NULL,`
	`CONSTRAINT BC_SFWS_TPS FOREIGN KEY (TRANS_ID) REFERENCES BC_SFWS_MESSAGES(TRANS_ID) ON DELETE CASCADE`
	`);`

Columns summary for the table BC_SFWS_TPINFO is shown in BC_SFWS_TPINFO: Columns Summary.

BC_SFWS_TPINFO: Columns Summary

Name	Data Type	Constraints	Nullable	Documentation
TRANS_ID	varchar2 (512)	FK (BC_SFWS_ MESSAGES.TRANS_ ID)	No	Foreign key to the column BC_SFWS_ MESSAGES.TRANS_ ID.
TPINFO_ID	varchar2 (512)		No	Primary key generated by TIBCO BusinessConnect.
TPNAME	varchar2 (255)		Yes	Partner name
TPDOMAIN	varchar2 (255)		Yes	Partner domain info, if supported by the business protocol.
TPID	varchar2 (255)		Yes	Partner ID, if supported by the business protocol.
HOSTNAME	varchar2 (255)		Yes	Participant host name
HOSTDOMAIN	varchar2 (255)		Yes	Host domain info, if supported by the business protocol.
HOSTID	varchar2 (255)		Yes	Host ID, if supported by the business protocol.
STATUS	varchar2(64)		Yes	Status of the current entry.

Name	Data Type	Constraints	Nullable	Documentation
AUX1	varchar2 (512)		Yes	
AUX2	varchar2 (512)		Yes	
AUX3	varchar2 (512)		Yes	
AUX4	varchar2 (512)		Yes	
AUX5	varchar2 (512)		Yes	
RTN_RECEIPT	varchar2 (255)		Yes	For future use
TS	timestamp		Yes	
EXPIRATION	timestamp		Yes	Expiration time
MAXMSGCOUNT	number(18)		No	For future use
CURMSGCOUNT	number(18)		No	For future use
EXTRA_INFO	blob		Yes	Additional Partner information stored as blob .
AUDITLOG_ CONTEXTID	varchar2 (100)		Yes	
MARKED_DEL	number(2)		Yes	

Indices information for the table BC_SFWS_TPINFO is shown in BC_SFWS_TPINFO Indices: SFWS_TPINFO_UTXID and BC_SFWS_TPINFO Indices: SFWS_TPINFO_TOP.

BC_SFWS_TPINFO Indices: SFWS_TPINFO_UTXID

SFWS_TPINFO_UTXID	
User IDLast Numeric Value	0
Unique	false
Index Name Pattern	{table_name}
Clustered	Non-clustered

BC_SFWS_TPINFO Indices: SFWS_TPINFO_TOP

SFWS_TPINFO_TOP	
User IDLast Numeric Value	0
Unique	false
Index Name Pattern	{table_name}
Clustered	Non-clustered

Relationship information for the table `BC_SFWS_TPINFO` is shown in *BC_SFWS_ATTACHMENTS_BIN: Relationships*.

BC_SFWS_ATTACHMENTS_BIN: Relationships

BC_SFWS_TPS: Relationship	
From	BC_SFWS_MESSAGES
User IDLast Numeric Value	0
Identifying	false
Subtype	false

BC_SFWS_TPS: Relationship	
On Delete	Cascade
On Update	Cascade
To Multiplicity	0..*
From Multiplicity	1
Sync To Association	Yes
Data Model	Physical

BC_MDN

Details for the table `BC_MDN` are explained in BC_MDN: Details.

BC_MDN: Details

Name	Value
`Data Model`	Physical
`Documentation`	This table stores information that is required for transport level receipts (Message Disposition Notification) to be received from the partner. It is used when protocols use public transports such as Email, AS1, and AS2, or transports that require receipts. This is a transient table, where the row data gets deleted once the receipt arrives, or if a timeout occurs when the receipt does not arrive within the time specified by the transport. This table need not be archived.
`User IDLast Numeric Value`	0
Primary Key Constraint Name	MDNREQUEST_LIST_PK

Name	Value

DDLClauses	`create table BC_MDN(`
	`MESSAGE_ID VARCHAR2(200) NOT NULL,`
	`PROTOCOL VARCHAR2(30) NOT NULL,`
	`OPERATION_ID VARCHAR2(60) NOT NULL,`
	`TRANSACTION_ID VARCHAR2(60) NOT NULL,`
	`HOST_NAME VARCHAR2(60) NOT NULL,`
	`TRADING_PARTNER VARCHAR2(60) NOT NULL,`
	`MESSAGE_MIC VARCHAR2(120) NOT NULL,`
	`RECORD_TIME TIMESTAMP dEFAULT SYSTIMESTAMP NOT NULL,`
	`MDN_OVERDUE_STATUS VARCHAR2(1) NOT NULL,`
	`CONSTRAINT MDNREQUEST_LIST_PK PRIMARY KEY(MESSAGE_ID)`
	`);`

Columns summary for the table `BC_MDN` is shown in BC_MDN: Columns Summary.

BC_MDN: Columns Summary

Name	Data Type	Constraints	Nullable	Documentation
MESSAGE_ID	varchar2(200)	PK	No	
PROTOCOL	varchar2(30)		No	
OPERATION_ID	varchar2(60)		No	
TRANSACTION_ID	varchar2(60)		No	
HOST_NAME	varchar2(60)		No	
MESSAGE_MIC	varchar2(60)		No	
RECORD_TIME	timestamp		No	
MDN_OVERDUE_ STATUS	varchar2(1)		No	

BC_HIBERNATION

Details for the table `BC_HIBERNATION` are explained in BC_HIBERNATION: Details.

BC_HIBERNATION: Details

Name	Value
Data Model	Physical
Documentation	This table stores transient information that has to be retrieved when receiving a response from a partner, or when receiving a response from a private process. This table data is used by almost all business protocols that support Synchronous and Asynchronous Request Response operations. This table is used in conjunction with other table data, such when used with AS2, AS1, and Email receipts for BC_MDN tables, as well as a part of marking the transactions as overdue when the messages are stored for PartnerExpress or FTP Server in `BC_SFWS_MESSAGES` tables. This table can grow depending on the number of transactions, but the data gets cleared when there are timeouts, or when the responses or appropriate handlers take care of retrieving and removing the row data. It need not be archived, however it should be considered when doing database sizing if there is a big number of transactions that are passing through TIBCO BusinessConnect. AUX columns are determined either by the business protocols, or by the handlers that are using them.
User IDLast Numeric Value	0

DDLClauses	`create table BC_HIBERNATION(`
	`HIBERKEY VARCHAR2(255) NOT NULL PRIMARY KEY,`
	`BININDEX NUMBER(18) NOT NULL UNIQUE,`
	`PROTOCOL VARCHAR2(32) NULL,`
	`EXPIRATION NUMBER(18) NOT NULL,`
	`STATUS NUMBER(3) NOT NULL,`
	`ALERTS NUMBER(5),`

Name	Value

```
        ORIGKEY         VARCHAR2(4000)  NULL,
        GSSTATE         VARCHAR2(255)   NULL,
        GSTYPE          VARCHAR2(128)   NULL,
        GSUSER          VARCHAR2(255)   NULL,
        TPNAME          VARCHAR2(255)   NULL,
        HOSTNAME        VARCHAR2(255)   NULL,
        OPERATION_ID    VARCHAR2(512)   NULL,
        STREAM_SIZE     NUMBER(18)      NULL,
        SF_TRANS_ID     VARCHAR2(255)   NULL,
        AUX1            VARCHAR2(255)   NULL,
        AUX2            VARCHAR2(255)   NULL,
        AUX3            VARCHAR2(255)   NULL,
        AUX4            VARCHAR2(255)   NULL,
        AUX5            VARCHAR2(255)   NULL,
        AUDITLOG_CONTEXT_ID VARCHAR2(255) NULL,
        USER_TRANS_ID   VARCHAR2(255)   NULL,
        TS              TIMESTAMP        DEFAULT SYSTIMESTAMP
);
```

Columns summary for the table `BC_HIBERNATION` is shown in BC_HIBERNATION: Columns Summary.

BC_HIBERNATION: Columns Summary

Name	Data Type	Constraints	Nullable	Documentation
`HIBERKEY`	varchar2(255)	`PK`	No	Internal key used by TIBCO BusinessConnect.
`BININDEX`	number(18)	Unique	No	Uniquely generated value which is a foreign key to `BC_HIBERNATION_BIN` table.
`PROTOCOL`	varchar2(32)		Yes	Business protocol or a Gateway protocol name.
`EXPIRATION`	number(18)		No	Indicates when this row will expire.

Name	Data Type	Constraints	Nullable	Documentation
STATUS	number(3)		No	Internal status value
ALERTS	number(5)		Yes	Currently not used
ORIGKEY	varchar2 (4000)		Yes	Original hibernation key, if length is more than 255 characters.
GSSTATE	varchar2(255)		Yes	Used only for PartnerExpress and FTP Server.
GSTYPE	varchar2(128)		Yes	Indicates whether the data is available for PartnerExpress or FTP Server.
GSUSER	varchar2(255)		Yes	Name of the user or host.
TPNAME	varchar2(255)		Yes	Partner name
HOSTNAME	varchar2(255)		Yes	Host name
OPERATION_ID	varchar2(512)		Yes	Operation ID of the transaction
STREAM_SIZE	number(18)		Yes	Payload size
SF_TRANS_ID	varchar2(255)		Yes	Reference to the BC_SFWS_MESSAGES column, if used by PartnerExpress or FTP Server.
AUX1	varchar2(255)		Yes	
AUX2	varchar2(255)		Yes	

Name	Data Type	Constraints	Nullable	Documentation
AUX3	varchar2(255)		Yes	
AUX4	varchar2(255)		Yes	
AUX5	varchar2(255)		Yes	
AUDITLOG_ CONTEXT_ID	varchar2(255)		Yes	
USER_TRANS_ID	varchar2(255)		Yes	
TS	timestamp		Yes	

Indices information for the table `BC_HIBERNATION` is shown in BC_HIBERNATION Indices: BC_HIBER_IDX1 and BC_HIBERNATION Indices: BC_HIBER_IDX2.

BC_HIBERNATION Indices: BC_HIBER_IDX1

BC_HIBER_IDX1	
User IDLast Numeric Value	0
Unique	false
Index Name Pattern	{table_name}
Clustered	Non-clustered

BC_HIBERNATION Indices: BC_HIBER_IDX2

BC_HIBER_IDX2	
User IDLast Numeric Value	0
Unique	false
Index Name Pattern	{table_name}
Clustered	Non-clustered

Relationship information for the table `BC_HIBERNATION` is shown in *BC_HIBERNATION: Relationships*.

BC_HIBERNATION: Relationships

BC_HIBREF: Relationship	
To	BC_HIBERNATION_BIN
User IDLast Numeric Value	0
Identifying	false
Subtype	false
On Delete	Cascade
On Update	Cascade
To Multiplicity	0..*
From Multiplicity	1
Sync To Association	Yes
Data Model	Physical

BC_

Details for the table `BC_` are explained in BC_: Details.

BC_ : Details

Name	Value
`Data Model`	Physical

Name	Value
Documentation	This table is used to store user saved queries from the TIBCO Administrator UI, or queries saved from PartnerExpress. The table value does not grow as much as transaction audit trail tables: it grows only when a big number of users saves queries. This table need not be archived.
User IDLast Numeric Value	0
DDLClauses	`CREATE TABLE BC_ ( QUERYOID VARCHAR2(128) , QUERYTP VARCHAR2(128) , QUERYDESC VARCHAR2(1024) , QUERYPROTOCOL VARCHAR2(128) , USERNAME VARCHAR2(128) , QUERYNAME VARCHAR2(128) , QUERYTYPE NUMBER(2) , BININDEX NUMBER(15) NOT NULL UNIQUE, CONSTRAINT BC_QUERY UNIQUE (USERNAME, QUERYNAME, QUERYTYPE) ) ;`

Columns summary for the table BC_ is shown in BC_: Columns Summary.

BC_
: Columns Summary

Name	Data Type	Constraints	Nullable	Documentation
QUERYOID	varchar2 (128)		Yes	
QUERYTP	varchar2 (128)		Yes	
QUERYDESC	varchar2		Yes	

Name	Data Type	Constraints	Nullable	Documentation
	(128)			
QUERYPROTOCOL	varchar2 (128)		Yes	
USERNAME	varchar2 (128)	Unique	Yes	
QUERYNAME	varchar2 (128)	Unique	Yes	
QUERYTYPE	number(2)	Unique	Yes	
BININDEX	number (15)	Unique	No	

Indices information for the table BC_ is shown in BC_ Indices: BC_QUERY.

BC_
Indices: BC_QUERY

BC_QUERY	
User IDLast Numeric Value	0
Unique	true
Index Name Pattern	{table_name}
Clustered	Non-clustered

Relationship information for the table BC_ is shown in *BC_: Relationships*.

**BC_
: Relationships**

BC_LOGQUERY: Relationship	
To	BC_LOGQUERYBIN
User IDLast Numeric Value	0
Identifying	false
Subtype	false
On Delete	Cascade
On Update	Cascade
To Multiplicity	0..*
From Multiplicity	0..1
Sync To Association	Yes
Data Model	Physical

BC_HIBERNATION_BIN

Details for the table `BC_HIBERNATION_BIN` are explained in BC_HIBERNATION_BIN: Details.

BC_HIBERNATION_BIN: Details

Name	Value
`Data Model`	Physical
`Documentation`	This table holds additional context information about the hibernation entries that are stored, such as information on how to collate the audit log when a response is received. This table is transient like its parent and need not be archived.

Name	Value
User IDLast Numeric Value	0
DDLClauses	`CREATE TABLE BC_HIBERNATION_BIN ( BININDEX NUMBER (18) NOT NULL, COMPRESSED VARCHAR2(2) NULL, BINVAL BLOB, CONSTRAINT BC_HIBREF FOREIGN KEY (BININDEX) REFERENCES BC_HIBERNATION(BININDEX) ON DELETE CASCADE);`

Columns summary for the table `BC_HIBERNATION_BIN` is shown in BC_HIBERNATION_BIN: Columns Summary.

BC_HIBERNATION_BIN: Columns Summary

Name	Data Type	Constraints	Nullable	Documentation
BININDEX	number (18)	FK (BC_HIBERNATION.BININDEX)	No	
COMPRESSED	varchar2 (2)		Yes	
BINVAL	blob		Yes	

Indices information for the table `BC_HIBERNATION_BIN` is shown in BC_HIBERNATION_BIN Indices: BC_HIBER_BIN_INDX.

BC_HIBERNATION_BIN Indices: BC_HIBER_BIN_INDX

BC_HIBER_BIN_INDX	
User IDLast Numeric Value	0
Unique	false
Index Name Pattern	{table_name}

BC_HIBER_BIN_INDX	
Clustered	Non-clustered

Relationship information for the table `BC_HIBERNATION_BIN` is shown in *BC_HIBERNATION_BIN: Relationships*.

BC_HIBERNATION_BIN: Relationships

BC_HIBREF: Relationship	
To	BC_HIBERNATION
User IDLast Numeric Value	0
Identifying	false
Subtype	false
On Delete	Cascade
On Update	Cascade
To Multiplicity	0..*
From Multiplicity	1
Sync To Association	Yes
Data Model	Physical

BC_LOGQUERYBIN

Details for the table `BC_LOGQUERYBIN` are explained in BC_LOGQUERYBIN: Details.

BC_LOGQUERYBIN: Details

Name	Value
Data Model	Physical
Documentation	This table contains the actual query saved as a blob, which has a foreign key to BC_ table. This table need not be archived.
User IDLast Numeric Value	0

DDLClauses	`CREATE TABLE BC_LOGQUERYBIN (`
	`BININDEX NUMBER(15),`
	`COMPRESSED VARCHAR2(2),`
	`BINVAL BLOB,`
	`CONSTRAINT BC_LOGQUERY FOREIGN KEY (BININDEX) REFERENCES BC_`
	`(BININDEX) ON DELETE CASCADE`
	`);`

Columns summary for the table BC_LOGQUERYBIN is shown in BC_LOGQUERYBIN: Columns Summary.

BC_LOGQUERYBIN: Columns Summary

Name	Data Type	Constraints	Nullable	Documentation
BININDEX	number	FK (BC_ .BININDEX)	Yes	Foreign key to the BC_ .BININDEX column.
COMPRESSED	varchar2(2)		Yes	Whether the BINVAL is compressed or not.
BINVAL	blob		Yes	Actual saved query with attributes and values stored in this BLOB.

Indices information for the table `BC_LOGQUERYBIN` is shown in BC_LOGQUERYBIN Indices: BC_LOGQUERYBIN.

BC_LOGQUERYBIN Indices: BC_LOGQUERYBIN

BC_LOGQUERYBIN	
User IDLast Numeric Value	0
Unique	false
Index Name Pattern	{table_name}
Clustered	Non-clustered

Relationship information for the table `BC_LOGQUERYBIN` is shown in *BC_LOGQUERYBIN: Relationships*.

BC_LOGQUERYBIN: Relationships

BC_LOGQUERY: Relationship	
To	BC_
User IDLast Numeric Value	0
Identifying	false
Subtype	false
On Delete	Cascade
On Update	Cascade
To Multiplicity	0..*
From Multiplicity	0..1
Sync To Association	Yes
Data Model	Physical

Configuration Store Reporting Schema Details

The tables in TIBCO BusinessConnect have configuration data in blob format. For creating External reports, new tables are derived from existing TIBCO BusinessConnect database schemas in order to store the binary (BLOB) information in a structured format. You can write SQL queries using these tables for building reports. These tables consist of key-value pair which holds data about partner settings, protocols, transports, business agreements, scheduled transmission and operation bindings components. In general, the key is internal identifier of a component's property of BusinessConnect.

The table description for the External Reporting schema is depicted in the following table:

Configuration Store Reporting Schema Details

Name	Documentation

BC_CS_PROTOCOL_PROPERTIES	This table stores the host or the partner settings of each protocol enabled for a trading partner. The data is sourced from BC_PROTOCOL table. The PROTOCOLID of this table corresponds to BC_PROTOCOL.OBJOID.
	
	For example, the following query returns the names of trading partners having X12 protocol enabled with Outbound XML to EDI Data Encoding property set to UTF-8:
	
	`SELECT TP.OBJNID`
	`FROM BC_PARTICIPANT TP`
	`INNER JOIN BC_PROTOCOL P`
	`ON P.OWNER_OID = TP.OBJOID`
	`INNER JOIN BC_CS_PROTOCOL_PROPERTIES PP`
	`ON PP.PROTOCOLID = P.OBJOID`
	`WHERE PP.PROTOCOLNAME = 'X12'`
	`AND PP.BCKEY = '_tpEnableOBEDIEncoding'`
	`AND PP.BCVALUE = 'UTF-8';`

BC_CS_TRANSPORT_PROPERTIES	This table stores the transport settings of each protocol enabled for a

Name	Documentation
	trading partner. The data is sourced from BC_CHANNELINFO table. The TRANSPORTID of this table corresponds to BC_ CHANNELINFO.OBJOID. For example, the following query returns the names of trading partners that use HTTP Basic Authentication: **SELECT TP.OBJNID** **FROM BC_PARTICIPANT TP** **INNER JOIN BC_PROTOCOL P** **ON P.OWNER_OID = TP.OBJOID** **INNER JOIN BC_CHANNELINFO TR** **ON TR.OWNER_OID = P.OBJOID** **INNER JOIN BC_CS_TRANSPORT_PROPERTIES TRP** **ON TRP.TRANSPORTID = TR.OBJOID** **WHERE TRP.TRANSPORTNAME = 'http'** **AND TRP.BCKEY = 'useHTTPBasicAuthentication'** **AND TRP.BCVALUE = 'true';**
BC_CS_BA_PROTOCOL_ PROPERTIES	This table stores the Agreement Protocol Binding settings of each protocol configured for a Business Agreement between the trading partners. The data is sourced from BC_PBV table. The PROTOCOLBINDINGID of this table corresponds to BC_PBV.OBJOID. For example, the following query returns the names of trading partners having X12 protocol enabled with Regenerate Control Number For Batch Resend set to true **SELECT TP.OBJNID** **FROM BC_PARTICIPANT TP** **INNER JOIN BC_BIZAGREEMENT BA** **ON BA.PROFILEB_OID = TP.OBJOID** **INNER JOIN BC_PB PB**

Name	Documentation
	``` ON PB.OWNER_OID = BA.OBJOID INNER JOIN BC_PBV PBV ON PBV.OWNER_OID = PB.OBJOID INNER JOIN BC_CS_BA_PROTOCOL_PROPERTIES PP ON PP.PROTOCOLBINDINGID = PBV.OBJOID WHERE PB.OBJNID = 'X12' AND PP.BCKEY = 'Batching._regenerateCtrlNum' AND PP.BCVALUE = 'true' ```
BC_CS_BA_STMS	This table stores the Scheduled Transmission settings of applicable Agreement Protocol Binding configured for a Business Agreement between the trading partners. The data is sourced from BC_PBV table. The PROTOCOLBINDINGID of this table corresponds to BC_ PBV.OBJOID. For example, the following query returns the Agreement name and Transmission Mode of Scheduled Transmission defined at Agreement Protocol Binding: ``` SELECT BZ.DISPLAY_NAME baName, STMS.BCVALUE transmissionMode FROM BC_BIZAGREEMENT BZ INNER JOIN BC_PB PB ON PB.OWNER_OID = BZ.OBJOID INNER JOIN BC_PBV PBV ON PBV.OWNER_OID = PB.OBJOID INNER JOIN BC_CS_BA_STMS STMS ON STMS.PROTOCOLBINDINGID = PBV.OBJOID WHERE STMS.BCKEY LIKE 'TransmissionWindow' ```
BC_CS_BA_OB_ PROPERTIES	This table stores the Operation Bindings settings of each Agreement Protocol Binding configured for a Business Agreement between the

Name	Documentation

	trading partners. The data is sourced from BC_OPB table. The PROTOCOLBINDINGID of this table corresponds to BC_OPB.OBJOID.
	
	For example, the following query returns the names of ebMS3 operations with Require MSH Receipt set to 'always' for Request Action:
	
	`SELECT OP.OBJNID`
	`FROM BC_INTFCOMPONENT OP`
	`INNER JOIN BC_OPB OB`
	`ON OB.TX_OID = OP.OBJOID`
	`INNER JOIN BC_CS_BA_OB_PROPERTIES OBP`
	`ON OBP.OPERATIONBINDINGID = OB.OBJOID`
	`WHERE OBP.BCKEY = 'request.Receipt.mshReceiptRequired'`
	`AND OBP.BCVALUE = 'always'`

Note

TIBCO BusinessConnect supports following database views:

- BC_CS_PARTICIPANT
- BC_CS_PROTOCOL
- BC_CS_LOCATION
- BC_CS_TRANSPORT
- BC_CS_BIZAGREEMENT
- BC_CS_PROTOCOL_BINDING
- BC_CS_PROTOCOL_BINDING_VIEW
- BC_CS_OPERATION_BINDING

BC_CS_PROTOCOL_PROPERTIES

Details for the table `BC_CS_PROTOCOL_PROPERTIES` are explained in BC_CS_PROTOCOL_PROPERTIES: Details

BC_CS_PROTOCOL_PROPERTIES: Details

Name	Valuel
Data Model	Physical
Documentation	This table stores the host or the partner settings of each protocol enabled for a trading partner. The data is sourced from BC_PROTOCOL table. The PROTOCOLID of this table corresponds to BC_PROTOCOL.OBJOID.
User IDLast Numeric Value	0

DDLClauses	`CREATE TABLE BC_CS_PROTOCOL_PROPERTIES (`
	`  PROTOCOLNAME VARCHAR(255) NOT NULL,`
	`  BCKEY VARCHAR(255) NOT NULL,`
	`  BCVALUE VARCHAR(255),`
	`  PROTOCOLID VARCHAR(255) NOT NULL,`
	`  PRIMARY KEY (PROTOCOLNAME, BCKEY, PROTOCOLID),`
	`  INDEX BC_CS_PROTOCOL_PROPERTIES_IX (PROTOCOLID ASC)`
	`  );`

Relationship information for the table `BC_CS_PROTOCOL_PROPERTIES` is shown in BC_CS_PROTOCOL_PROPERTIES: Relationships: BC_PROTOCOL

BC_CS_PROTOCOL_PROPERTIES: Relationships: BC_PROTOCOL

MESSAGES: Relationship	
From	BC_PROTOCOL
User IDLast Numeric Value	0
Identifying	true
Subtype	false
On Delete	No
To Multiplicity	1..*
From Multiplicity	1
Sync To Association	Yes
Data Model	Physical
BLOB type columns	HPROPS, PPROPS

BC_CS_TRANSPORT_PROPERTIES

Details for the table BC_CS_TRANSPORT_PROPERTIES are explained in BC_CS_TRANSPORT_PROPERTIES: Details.

BC_CS_TRANSPORT_PROPERTIES: Details

Name	Valuel
Data Model	Physical
Documentation	This table stores the transport settings of each protocol enabled for a trading partner. The data is sourced from BC_CHANNELINFO table. The

Name	Valuel
	TRANSPORTID of this table corresponds to BC_CHANNELINFO.OBJOID.
User IDLast Numeric Value	0

DDLClauses	`CREATE TABLE BC_CS_TRANSPORT_PROPERTIES (`
	`  TRANSPORTNAME VARCHAR(255) NOT NULL,`
	`  BCKEY VARCHAR(255) NOT NULL,`
	`  BCVALUE VARCHAR(255),`
	`  TRANSPORTID VARCHAR(255) NOT NULL,`
	`PRIMARY KEY (TRANSPORTNAME, BCKEY, TRANSPORTID),`
	`INDEX BC_CS_TRANSPORT_PROPERTIES_IX (TRANSPORTID ASC)`
	`  );`

Relationship information for the table `BC_CS_TRANSPORT_PROPERTIES` is shown in BC_CS_TRANSPORT_PROPERTIES: Relationships: .

BC_CS_TRANSPORT_PROPERTIES: Relationships:

MESSAGES: Relationship	
From	BC_CHANNELINFO
User IDLast Numeric Value	0
Identifying	true
Subtype	false
On Delete	No
To Multiplicity	1..*

MESSAGES: Relationship	
From Multiplicity	1
Sync To Association	Yes
Data Model	Physical
BLOB type columns	CONTENT

bc_channelinfo

BC_CS_BA_PROTOCOL_PROPERTIES

Details for the table `BC_CS_BA_PROTOCOL_PROPERTIES` are explained in BC_CS_BA_PROTOCOL_PROPERTIES : Details.

BC_CS_BA_PROTOCOL_PROPERTIES : Details

Name	Valuel
`Data Model`	Physical
`Documentation`	This table stores the Agreement Protocol Binding settings of each protocol configured for a Business Agreement between the trading partners. The data is sourced from BC_PBV table. The PROTOCOLBINDINGID of this table corresponds to BC_PBV.OBJOID.
`User IDLast` `Numeric Value`	0

Name	Valuel

DDLClauses	`CREATE TABLE BC_CS_BA_PROTOCOL_PROPERTIES (`
	`  PARTICIPANTID VARCHAR(255) NOT NULL,`
	`  BCKEY VARCHAR(255) NOT NULL,`
	`  BCVALUE VARCHAR(255),`
	`  PROTOCOLBINDINGID VARCHAR(255) NOT NULL,`
	`  PRIMARY KEY (PARTICIPANTID, BCKEY,`
	`PROTOCOLBINDINGID),`
	`  INDEX BC_CS_BA_PROTOCOL_PROPERTIES_IX`
	`(PROTOCOLBINDINGID ASC)`
	`  );`

Relationship information for the table `BC_CS_BA_PROTOCOL_PROPERTIES` is shown in BC_CS_BA_PROTOCOL_PROPERTIES : Relationships: .

BC_CS_BA_PROTOCOL_PROPERTIES : Relationships:

MESSAGES: Relationship	
From	BC_PBV
User IDLast Numeric Value	0
Identifying	true
Subtype	false
On Delete	No
To Multiplicity	1..*
From Multiplicity	1
Sync To Association	Yes
Data Model	Physical
BLOB type columns	PPROPS

bc_pbv

BC_CS_BA_STMS

Details for the table `BC_CS_BA_STMS` are explained in Table 138.

BC_CS_BA_STMS : Details

Name	Valuel
Data Model	Physical
Documentation	This table stores the Scheduled Transmission settings of applicable Agreement Protocol Binding configured for a Business Agreement between the trading partners. The data is sourced from BC_PBV table. The PROTOCOLBINDINGID of this table corresponds to BC_PBV.OBJOID.
User IDLast Numeric Value	0
DDLClauses	`CREATE TABLE BC_CS_BA_STMS ( PROTOCOLBINDINGID VARCHAR(255) NOT NULL, BCKEY VARCHAR(255) NOT NULL, BCVALUE VARCHAR(255), PRIMARY KEY (PROTOCOLBINDINGID, BCKEY), INDEX BC_CS_BA_STMS_IX (PROTOCOLBINDINGID ASC) );`

Relationship information for the table `BC_CS_BA_STMS` is shown in Table 139.

BC_CS_BA_STMS : Relationships:

MESSAGES: Relationship	
From	BC_PBV
User IDLast Numeric Value	0
Identifying	true
Subtype	false
On Delete	No
To Multiplicity	1..*
From Multiplicity	1
Sync To Association	Yes
Data Model	Physical
BLOB type columns	STMS

bc_pbv

BC_CS_BA_OB_PROPERTIES

Details for the table `BC_CS_BA_OB_PROPERTIES` are explained in BC_CS_BA_OB_PROPERTIES: Details.

BC_CS_BA_OB_PROPERTIES: Details

Name	Valuel
`Data Model`	Physical

Name	Valuel
Documentation	This table stores the Operation Bindings settings of each Agreement Protocol Binding configured for a Business Agreement between the trading partners. The data is sourced from BC_OPB table. The PROTOCOLBINDINGID of this table corresponds to BC_OPB.OBJOID.
User IDLast Numeric Value	0

DDLClauses	`CREATE TABLE BC_CS_BA_OB_PROPERTIES (`
	`  OPERATIONBINDINGNAME VARCHAR(255) NOT NULL,`
	`  BCKEY VARCHAR(255) NOT NULL,`
	`  BCVALUE VARCHAR(255),`
	`  OPERATIONBINDINGID VARCHAR(255) NOT NULL,`
	`  PRIMARY KEY (OPERATIONBINDINGNAME, BCKEY, OPERATIONBINDINGID),`
	`  INDEX BC_CS_BA_OB_PROPERTIES_IX (OPERATIONBINDINGID ASC)`
	`  ) ;`

Relationship information for the table BC_CS_BA_OB_PROPERTIES is shown in BC_CS_BA_OB_PROPERTIES: Relationships: .

BC_CS_BA_OB_PROPERTIES: Relationships:

MESSAGES: Relationship	
From	BC_OPB
User IDLast Numeric Value	0
Identifying	true
Subtype	false

MESSAGES: Relationship	
On Delete	No
To Multiplicity	1..*
From Multiplicity	1
Sync To Association	Yes
Data Model	Physical
BLOB type columns	PROPS

bc_opb

TIBCO BusinessConnect Palette Projects

This section explains how to deploy TIBCO BusinessConnect Palette projects on TIBCO BusinessConnect.

Deploying TIBCO BusinessConnect Palette Projects

If you used TIBCO BusinessConnect Palette to configure private processes for your TIBCO BusinessConnect application, create a project EAR (enterprise archive resource) file in TIBCO Designer and deploy the EAR on a server container in TIBCO ActiveMatrix BusinessWorks.

To deploy a TIBCO BusinessConnect palette project, first create the project EAR file as described in *TIBCO Designer User Guid*e, section Creating an Enterprise Archive.

Next, perform these steps in TIBCO Administrator:

1. In the left panel of TIBCO Administrator, expand **Application Management**.
2. Click **All Applications** in the left panel.
3. Click **New Application** in the right panel.
4. Browse for and select the EAR file.
5. Click **OK**.
6. Click **Save**.

Click **Deploy** to deploy and run the process following the same procedure described in Checking the State of the Interior Server Instance

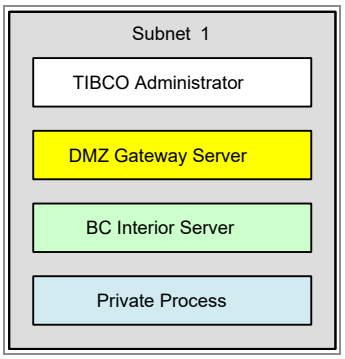
Deployment Tuning

This section explains how to tune some of the advanced properties used with the system.

Deploying the Interior Server on two Subnets

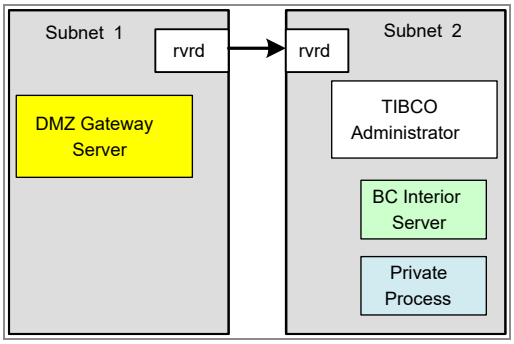
In general, it is recommended that a B2B deployment be located entirely within one single subnet for simplicity. TIBCO BusinessConnect Deployed on One Subnet shows deployment on one subnet.

Figure 10: TIBCO BusinessConnect Deployed on One Subnet



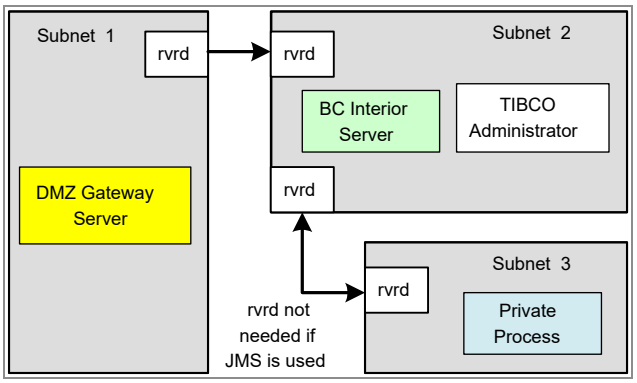
In a B2B deployment where the Gateway Server is residing on an isolated and dedicated machine, for security reasons the Gateway Server and the Interior Server are located on different subnets. In this case, it is required to have a pair of RVRDs (TIBCO Rendezvous routing daemons) to pull messages from the Gateway Server into interior subnet (Gateway Server and Interior Server Deployed on Two Subnets).

Figure 11: Gateway Server and Interior Server Deployed on Two Subnets



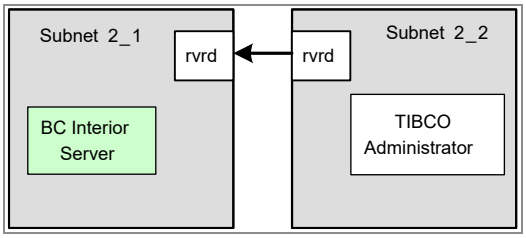
In addition to deploying the Gateway Server on a separate subnet, it is very common to separate private processes on their own subnet for better de-coupling of functional areas (Private Processes Deployed on a Separate Subnet).

Figure 12: Private Processes Deployed on a Separate Subnet



It is also possible to further break up subnet 2 in order to place TIBCO Administrator in its own subnet (TIBCO Administrator Deployed on a Separate Subnet).

Figure 13: TIBCO Administrator Deployed on a Separate Subnet



In order to ensure that messages are sent and received correctly by the TIBCO BusinessConnect engine, these RVRD processes must be configured to route the following messages: *installation_prefix*.>

For example, `AX.BC.ACME.>`

The TIBCO Rendezvous service should follow the value established for the service in **Intercomponent Communication > Interior Settings > Service**.

Modifying Load Balancing Properties

There are two different RVCM queues used to achieve load balancing:

- DMZ to Interior inbound RVCM queue
- Private Process to Interior outbound RVCM queue

In addition to specifying load balancing properties using the TIBCO BusinessConnect GUI, you can edit them directly in the following file:

`BusinessConnect-Interior_Server.tra`

This file is located in the directory such as:

tra_home\domain*domain_name*\application\BusinessConnect

(For Windows) `C:\tibco\tra\domain\`*domain_name*`\application\BusinessConnect`

DMZ to Interior Queue

The properties for the DMZ to Interior inbound Queue are as follows:

DMZ to Interior Inbound Queue

Property	Description
`tibco.clientVar.gatewayProperties/transports/Intercomponent/msh/wweight`	Worker Weight for Interior worker
`tibco.clientVar.gatewayProperties/transports/Intercomponent/msh/wtasks`	Worker Tasks for Interior worker
`tibco.clientVar.gatewayProperties/transports/Intercomponent/msh/wcompletetime`	Worker Completion time for

Property	Description
	Interior worker
`tibco.clientVar.gatewayProperties/transports/Intercomponent/msh/sweight`	Scheduler Weight for Interior scheduler
`tibco.clientVar.gatewayProperties/transports/Intercomponent/msh/sheartbeat`	Scheduler Heartbeat for Interior scheduler
`tibco.clientVar.gatewayProperties/transports/Intercomponent/msh/sactivation`	Scheduler Activation for Interior scheduler

To override the defaults follow these instructions:

1. Edit the `BusinessConnect-Interior_Server.tra` file located in the *tra_home*`\domain\`*domain_name*`\application\BusinessConnect` directory.

   tra_home is the directory where TIBCO Runtime Agent is installed.

   domain_name is the name of the domain in which TIBCO BusinessConnect is installed and deployed.

   For the definitions of RVCMQ, Worker Weight, Scheduler Weight, and other values used in this document, please refer to the TIBCO Rendezvous Distributed Queue documentation.

2. Modify the properties in the `BusinessConnect-Interior_Server.tra` file as required. These properties should be modified in the `BusinessConnect-Interior_Server.tra` file every time TIBCO BusinessConnect is deployed or redeployed.

Private Process to Interior Queue

The properties for the Private Process to Interior Queue are as follows:

Private Process to Interior Queue

Property	Description
tibco.clientVar.gatewayProperties/transports/backoffice/RVCMQ/worker/weight	Worker Weight for Outbound worker
tibco.clientVar.gatewayProperties/transports/backoffice/RVCMQ/worker/tasks	Worker Tasks for Outbound worker
tibco.clientVar.gatewayProperties/transports/backoffice/RVCMQ/worker/completeTime	Worker Completion time for Outbound worker
tibco.clientVar.gatewayProperties/transports/backoffice/RVCMQ/scheduler/weight	Scheduler Weight for Outbound scheduler
tibco.clientVar.gatewayProperties/transports/backoffice/RVCMQ/scheduler/heartbeat	Scheduler Heartbeat for Outbound scheduler
tibco.clientVar.gatewayProperties/transports/backoffice/RVCMQ/scheduler/activation	Scheduler Activation for Outbound scheduler

To override the defaults follow these instructions:

3. Edit the `BusinessConnect-Interior_Server.tra` file located in the *tra_home*`\domain\`*domain_name*`\application\BusinessConnect` directory.

tra_home is the directory where TIBCO Runtime Agent is installed.

domain_name is the name of the domain in which TIBCO BusinessConnect is installed and deployed.

For the definitions of RVCMQ, Worker Weight, Scheduler Weight, and other values used in this document, please refer to the TIBCO Rendezvous Distributed Queue documentation.

4. Modify the properties in the `BusinessConnect-Interior_Server.tra` file as required. These properties should be modified in the `BusinessConnect-Interior_Server.tra` file every time TIBCO BusinessConnect is deployed or redeployed.

Editing bcengine.tra to Tune Load Balancing

The `bcengine.tra` file is located in the following directory:

BC_HOME`/bin/bcengine.tra`

There are some properties that you can edit in the file `bcengine.tra` in order to tune load balancing performance.

These properties are:

- `java.property.bmh.process.outbound.threads=16`

  This property controls the size of the threadpool used to spin off sub-tasks in a protocol, such as outbound document schema validation.

- `java.property.bmh.doc.validation.threads=16`

  This property controls the size of the threadpool used to spin off sub-tasks in a protocol, such as inbound document schema validation.

- `java.property.msh.process.response.threads=16`

  This property controls the size of the threadpool used to spin off sub-tasks in a protocol, such as processing synchronous response from HTTP POST.

- `java.property.msh.process.inbound.threads=16`

  This property controls the size of the threadpool used to spin off sub-tasks in a protocol, such as decrypting a document or verifying a digital signature in the incoming message.

- `java.property.bc.dblock.monitor.wait=30000`

  This property controls the interval in milliseconds in which the `DBLock` monitor will examine the states of locks in the system and publish lock alerts.

TIBCO Documentation and Support Services

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

How to Access TIBCO Documentation

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

The TIBCO Product Documentation website is updated frequently and is more current than any other documentation included with the product.

Product-Specific Documentation

Documentation for TIBCO BusinessConnect™ is available on the TIBCO BusinessConnect™ Product Documentation page.

To directly access documentation for this product, double-click the following file:

TIBCO_HOME`/release_notes/TIB_bc_7.4.0_docinfo.html` where *TIBCO_HOME* is the top-level directory in which TIBCO products are installed. On Windows, the default *TIBCO_HOME* is `C:\tibco`. On UNIX systems, the default *TIBCO_HOME* is `/opt/tibco`.

The following documents for this product can be found in the TIBCO Documentation site:

- *TIBCO BusinessConnect™ Installation and Configuration*
- *TIBCO BusinessConnect™ Concepts*
- *TIBCO BusinessConnect™ Interior Server Administration*
- *TIBCO BusinessConnect™ Gateway Server Administration*
- *TIBCO BusinessConnect™ Training Partner Administration Guide*
- *TIBCO BusinessConnect™ Scripting Deployment User Guide*
- *TIBCO BusinessConnect™ Release Notes*

How to Contact TIBCO Support

Get an overview of TIBCO Support. You can contact TIBCO Support in the following ways:

- For accessing the Support Knowledge Base and getting personalized content about products you are interested in, visit the TIBCO Support website.
- 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 TIBCO Support website. 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 TIBCO Community.

Legal and Third-Party Notices

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

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

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

TIBCO, the TIBCO logo, the TIBCO O logo, ActiveMatrix BusinessWorks, TIBCO Administrator, TIBCO Designer, Hawk, Rendezvous, and TIBCO Runtime Agent are either registered trademarks or trademarks of Cloud Software Group, Inc. in the United States and/or other countries.

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

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

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

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. CLOUD SOFTWARE GROUP, 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 Cloud Software Group, Inc. may be covered by registered patents. Please refer to TIBCO's Virtual Patent Marking document (https://www.tibco.com/patents) for details.

Copyright © 2001-2023. Cloud Software Group, Inc. All Rights Reserved.