



TIBCO PartnerExpress™

User's Guide

*Version 6.2.1
December 2020*



Contents

Preface	iv
Related Documentation	v
TIBCO PartnerExpress Documentation	v
Other TIBCO Product Documentation	v
Typographical Conventions	vii
TIBCO Product Documentation and Support Services	x
How to Access TIBCO Documentation	x
How to Contact TIBCO Support	x
How to Join TIBCO Community	x
Chapter 1 Introduction	1
Overview	2
Functional Components	3
User Profiles	4
TIBCO PartnerExpress Interactions	5
Send Notify Operation	5
Receive Notify Operation	5
Send Request/Response Operation	6
Receive Request/Response Operation	7
Chapter 2 Setting Up the PartnerExpress Server on the Server Side	9
Overview	10
Activating a TIBCO PartnerExpress Server	11
Creating a Gateway Service	12
Assigning the Gateway Service to a Gateway Instance	15
Configuring the Outbound Transport	16
Customizing Product Name and Logo	18
Chapter 3 Using TIBCO PartnerExpress on the Client Side	21
Overview	22
Authenticating with the PartnerExpress Server	23
Signing on to PartnerExpress Using Secure URL	24
Changing Logon Password	25
Home View	26

- Inbox Summary 26
 - Transaction History 27
- Inbox View 29
 - Side Window 29
 - Main Window 32
- History View 34
 - Side Window 34
 - Main Window 35
- PX REST APIs 36
 - Invoking PX REST APIs by Using Clients 37
 - PX REST Operations 42
 - HTTP Wire Format 48
 - JSON Request Parameters Example 53
 - PX REST Swagger Specification 54
- Legal and Third-Party Notices 89**

Preface

TIBCO PartnerExpress[™] provides a secure web-based access for trading partners, so that the external users associated with these trading partners can log on and perform simple file uploads and downloads.

Topics

- [Related Documentation, page v](#)
- [Typographical Conventions, page vii](#)
- [TIBCO Product Documentation and Support Services, page x](#)

Related Documentation

This section lists documentation resources you may find useful.

TIBCO PartnerExpress Documentation

The following documents form the TIBCO PartnerExpress documentation set:

- TIBCO PartnerExpress *Installation and Configuration* Read this manual for instructions on site preparation and installation.
- TIBCO PartnerExpress *User's Guide* Read this manual for instructions on how to use TIBCO PartnerExpress to upload and download transaction files.
- TIBCO PartnerExpress *Release Notes* Read the release notes for a list of new and changed features. This document also contains lists of known issues and closed issues for this release.

Other TIBCO Product Documentation

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

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

products, such as TIBCO Designer, Java Runtime Environment (JRE), TIBCO Hawk[®], and TIBCO Rendezvous[®].

- TIBCO Rendezvous: this product contains two main components, including the Rendezvous programming language interface (API) in several languages, and the Rendezvous daemon. Programs running on many different kinds of computers on a network can communicate seamlessly by using this product.
- TIBCO Enterprise Message Service[™]: this product provides a message service that enables integration of applications within an enterprise based on the Java Message Service (JMS) specifications.
- TIBCO BusinessConnect[™] Palette: this product is integrated with TIBCO ActiveMatrix BusinessWorks to make a good communication with TIBCO BusinessConnect server.

Typographical Conventions

The following typographical conventions are used in this manual.

Table 1 General Typographical Conventions

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

Table 1 General Typographical Conventions (Cont'd)




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

Table 2 Syntax Typographical Conventions

Convention	Use
[]	<p>An optional item in a command or code syntax.</p> <p>For example:</p> <pre>MyCommand [optional_parameter] required_parameter</pre>
	<p>A logical OR that separates multiple items of which only one may be chosen.</p> <p>For example, you can select only one of the following parameters:</p> <pre>MyCommand para1 param2 param3</pre>

Table 2 Syntax Typographical Conventions (Cont'd)

Convention	Use
{ }	<p>A logical group of items in a command. Other syntax notations may appear within each logical group.</p> <p>For example, the following command requires two parameters, which can be either the pair param1 and param2, or the pair param3 and param4.</p> <pre>MyCommand {param1 param2} {param3 param4}</pre> <p>In the next example, the command requires two parameters. The first parameter can be either param1 or param2 and the second can be either param3 or param4:</p> <pre>MyCommand {param1 param2} {param3 param4}</pre> <p>In the next example, the command can accept either two or three parameters. The first parameter must be param1. You can optionally include param2 as the second parameter. And the last parameter is either param3 or param4.</p> <pre>MyCommand param1 [param2] {param3 param4}</pre>

TIBCO Product Documentation and Support Services

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

How to Access TIBCO Documentation

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

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

Documentation for TIBCO PartnerExpress is available on the [TIBCO PartnerExpress](#) Product Documentation page.

How to Contact TIBCO Support

You can contact TIBCO Support in the following ways:

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

How to Join TIBCO Community

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

Chapter 1 **Introduction**

This chapter introduces TIBCO PartnerExpress™.

Topics

- [Overview, page 2](#)
- [Functional Components, page 3](#)
- [User Profiles, page 4](#)
- [TIBCO PartnerExpress Interactions, page 5](#)

Overview

TIBCO PartnerExpress provides a secure web-based access for trading partners, so that external users associated with these trading partners can log on and perform simple file uploads and downloads.

By using TIBCO PartnerExpress clients, external users can connect to the host site running the TIBCO PartnerExpress Server.

TIBCO PartnerExpress software has two components:

- TIBCO PartnerExpress Server

It is located on the TIBCO BusinessConnect Gateway Server, which can also hold other servers, such as HTTP servers and FTP servers.

- TIBCO PartnerExpress Web Client

It provides a browser interface for external users of trading partners.

No software installation is required on the web client.

TIBCO PartnerExpress does not require a specific protocol.



TIBCO PartnerExpress Server is referred to as PartnerExpress server, and TIBCO PartnerExpress Web Client is referred to as PartnerExpress client later in this document.

This product has the following features:

- Standards Support

PartnerExpress clients communicate with the PartnerExpress server by exchanging documents over HTTPS. Support for this industry standard means that the software can be implemented easily by using existing technology infrastructure.

- Scalability

After all components are installed and configured on the PartnerExpress server side, PartnerExpress clients can be added easily. Because only a web browser is required to conduct transactions, the PartnerExpress client setup is minimal.

Functional Components

TIBCO PartnerExpress uses two TIBCO product components, TIBCO BusinessConnect and TIBCO Rendezvous.

TIBCO BusinessConnect

TIBCO BusinessConnect is a business-to-business server that enables secure exchanges of business documents and automates cross-company processes. Its architecture leverages TIBCO expertise with Enterprise Application Integration (EAI) to integrate systems and processes across company boundaries.

TIBCO BusinessConnect also provides partner management features with which you can define and maintain complex partner relationships in electronic commerce efficiently.

TIBCO Rendezvous

With TIBCO Rendezvous, you can exchange data among applications across a network. It supports network data transport, network data representation, and many hardware and software platforms. Therefore, applications running on many different kinds of computers on a network can communicate seamlessly.

TIBCO Rendezvous applications communicate by exchanging messages sent on subjects. The messages can contain any kind of data except memory pointers and usually contain notifications of business events and associated data.

TIBCO Rendezvous features self-describing data structure that contains information about its data type, size, and name. Rendezvous applications on heterogeneous platforms can communicate easily by exchanging self-describing data.

User Profiles

After TIBCO PartnerExpress is installed, the BusinessConnect installed on the PartnerExpress server site can communicate with internal users and the external users that are associated with one of the trading partners securely.

BusinessConnect Administrator

The BusinessConnect administrator on the host site configures the PartnerExpress server components, specifies external users, and associates these external users with a trading partner.

The BusinessConnect administrator must have advanced knowledge of TIBCO BusinessConnect software, other TIBCO softwares, and their network environment.

External Users

After the BusinessConnect administrator sets up a trading partner and associates external users with that trading partner, the external users can log on by using a web browser, and then upload or download transactions to or from the PartnerExpress server.

The external users on the PartnerExpress client site use a web browser, and they are only required to have a basic knowledge of the business transactions they are performing.

TIBCO PartnerExpress Interactions

In TIBCO PartnerExpress, transactions initiated by the PartnerExpress server to PartnerExpress clients are not sent directly to the trading partners' systems, but to a repository where the external users can find the files available for them to download upon logon. While transactions initiated by PartnerExpress clients to the PartnerExpress server are forwarded directly to the business protocol handler for processing.

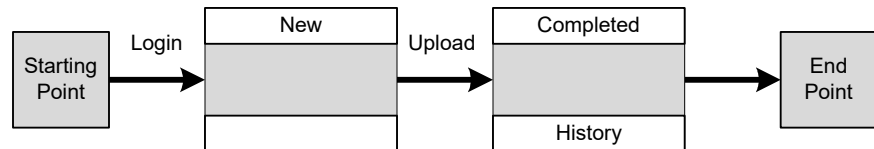
On the PartnerExpress client site, there are four types of operation:

- Send Notify operation
- Receive Notify operation
- Send Request/Response operation
- Receive Request/Response operation

Send Notify Operation

As shown in [Figure 1](#), when a PartnerExpress client logs on, the associated operations with the PartnerExpress server are listed, including the Notify type of operation and the Request/Response type of operation. The PartnerExpress client can then select the Notify operation type and send a Notify request to the PartnerExpress server.

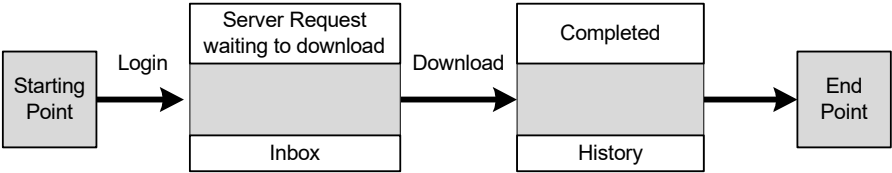
Figure 1 Send Notify Operation in TIBCO PartnerExpress



Receive Notify Operation

When a PartnerExpress client receives Notify requests sent by the PartnerExpress server, all the requests are staged in the Inbox location. After the external user logs on from the client side, a list of transaction files available for downloading is displayed. This list is same for all the external users that are associated with the same trading partner, regardless of the specific request each of these external users initiates. In [Figure 2](#), the Notify request is initiated by the PartnerExpress server.

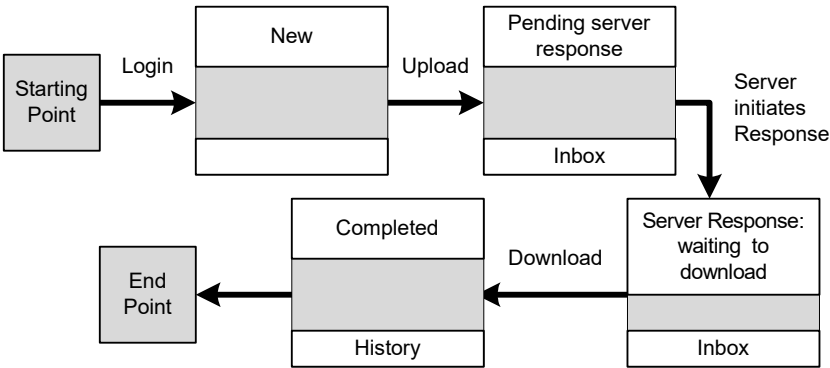
Figure 2 Receive Notify Operation in TIBCO PartnerExpress



Send Request/Response Operation

In the Request/Response operation, a PartnerExpress client logs on and uploads a trading document request. This request is then transferred to the PartnerExpress server. If an outbound store-and-forward process is configured, the PartnerExpress server responds and the response document is stored in the Inbox location. Next time the PartnerExpress client logs on, the response to the previously uploaded request is available for downloading.

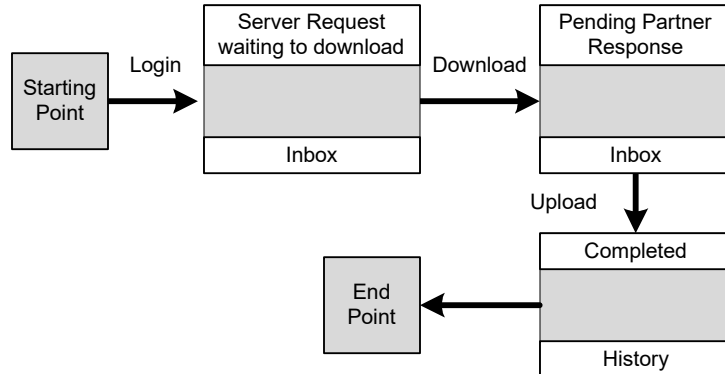
Figure 3 Send Request/Response Operation in TIBCO PartnerExpress



Receive Request/Response Operation

When the PartnerExpress server receives a Request/Response operation, it initiates a request. When a PartnerExpress client logs on, the request is available for downloading. After the PartnerExpress client downloads the request, the client uploads a response and initiates it back to the PartnerExpress server.

Figure 4 Receive Request/Response Operation in TIBCO PartnerExpress



Chapter 2

Setting Up the PartnerExpress Server on the Server Side

This chapter explains how to use TIBCO Administrator to configure the PartnerExpress server.

Topics

- [Overview, page 10](#)
- [Activating a TIBCO PartnerExpress Server, page 11](#)
- [Creating a Gateway Service, page 12](#)
- [Assigning the Gateway Service to a Gateway Instance, page 15](#)
- [Configuring the Outbound Transport, page 16](#)
- [Customizing Product Name and Logo, page 18](#)

Overview

Before setting up the PartnerExpress server, ensure that the trading partners are configured. For more information, see *TIBCO BusinessConnect™ Services Plug-in User's Guide*, Setting Up Trading Hosts and Partners.

To install and configure the PartnerExpress server by using TIBCO Administrator, complete the following tasks:

1. [Activating a TIBCO PartnerExpress Server](#)
2. [Creating an External User](#)

Create an external user that can sign on to the PartnerExpress server. The connection is based on BusinessConnect internal database or LDAP.

For more information, see *TIBCO BusinessConnect Trading Partner Administration*, Adding External User and Authenticating External Users.

3. [Creating a Gateway Service](#)
4. [Assigning the Gateway Service to a Gateway Instance](#)
5. [Configuring the Outbound Transport](#)

Activating a TIBCO PartnerExpress Server

After installing TIBCO PartnerExpress on the server side, you can activate it using the same method as all other TIBCO BusinessConnect plug-ins.

To activate the PartnerExpress server, complete the following steps:

1. Log on to TIBCO Administrator.
2. Click **BusinessConnect** in the left panel, and then click **Manage** in the right panel.
3. In the **Manage Installation** window, click the **Protocol Plug-In Activation** tab.
4. Select the check box next to the GS-PX protocol plug-in, and then click **Activate**. You are logged off from the system automatically.
5. Log on to TIBCO Administrator again and ensure that the status of TIBCO PartnerExpress is **Activated**, which indicates that the PartnerExpress server is activated successfully.

Creating a Gateway Service

After external users are created and assigned to a partner, the BusinessConnect administrator can define which Gateway Services they can use.

For more information about how to create external users, see *TIBCO BusinessConnect Trading Partner Administration*, Adding External Users and Authenticating External Users.

To create a TIBCO PartnerExpress Gateway Service, complete the following steps:

1. In TIBCO Administrator, expand **BusinessConnect > Gateway > Gateway Services**.
2. In the Gateway Services window, click **New**.
3. In the New Gateway Services window, enter a name for the Gateway Service, select **PX** from the **Type** list, and then click **OK**. Click **Save**.
4. In the New Gateway Service: *service name* window, configure the Gateway Service, and then click **Save**.

For descriptions of the fields in the New Gateway Service: *service name* window, see the following table:

Table 1 Edit Gateway Service for TIBCO PartnerExpress

Field	Description
General Tab	
Name	Specify the name of the Gateway Service.
Active	If you want to activate the Gateway Service right away, select this check box. Otherwise, you can select it later.
Type	Select PX as the protocol type. The protocol type cannot be changed later.
Credentials Tab	
Export	If you want to export a specific private key, select it, and then click Export .

Table 1 Edit Gateway Service for TIBCO PartnerExpress (Cont'd)

Field	Description
New Private Key	<p>If you want to add a new private key for this Gateway Service, click New Private Key.</p> <p>In the New Private Key panel, specify the following information:</p> <ul style="list-style-type: none">— Alias The alias of the new private key.— Current Credential If no file is displayed, click change, and then specify the private key that you want to use for this Gateway Service.— Password The password of the private key.
Delete	<p>If you want to delete a specific private key, select it, and then click Delete.</p>

Table 1 Edit Gateway Service for TIBCO PartnerExpress (Cont'd)

Field	Description
Transport Tab Ports	
Secure Port	The port number for the secure port. The default port is 6809.
Private Key Credential for Secure Ports	The private key for the secure port. This private key is configured in the Credentials tab.
Advanced Session Settings	
User Session Timeout (min)	The timeout in minutes for the user session. The default value is 30.
Advanced Gateway to Interior Settings	
Data Streaming Threshold (KB)	The threshold in KB for data streaming between the Gateway Server and the Interior Server. The default value is 10000.
Request Timeout (secs)	The timeout in seconds for the communications between the Gateway Server and Interior Server. The default value is 30.

Assigning the Gateway Service to a Gateway Instance

After configuring the Gateway Service, you have to assign it to a running Gateway Instance so that you can use the Gateway Service. For more information, see *TIBCO BusinessConnect Gateway Server Administration*, Managing Gateway Instances.

To assign the Gateway Service to a Gateway Instance, complete the following steps:

1. In TIBCO Administrator, expand **BusinessConnect > Gateway > Gateway Instances**.
2. In the Gateway Instances window, click the radio button next to an instance, and then click **Manage**.
3. In the Manage Services window, click **PX**, and then click **Assign**.
4. In the Configure Service window, select a service from the **Gateway Server Group** list. Click **OK**. Click **Done**.
5. In the Gateway Instances window, click **Start** to start the service.

After the Gateway Service is started, its state is changed to `RUNNING-service` and its status is `OK`.

Configuring the Outbound Transport

Before configuring the PartnerExpress server, you have to set up the trading participants. For more information, see *TIBCO BusinessConnect™ Services Plug-in User's Guide*, Chapter 6, "Setting Up Trading Hosts and Partners" or *TIBCO BusinessConnect™ EDI Protocol powered by Instream User's Guide*, Chapter 3, in the section "Setting Up Trading Partners."

After setting up the trading participants, you must configure the outbound transport associated with the trading partner by completing the following steps:

1. In TIBCO Administrator, expand **BusinessConnect > Participants > partner > Protocols**.
2. If there are no available enabled protocols, click **Enable**, and then select one or more protocol name check boxes. Click **OK**.
3. In the Edit Enabled Protocol window, click the **Transports** tab, and then click **Add**.
4. In the New Transport window, enter a name for the new transport, select **Inbox** from the **Type** list, and then click **OK**.
5. In the New Inbox Transport window, configure the new transport, and then click **Save**.

For descriptions of the fields in the New Inbox Transport window, see the following table:

Table 2 New Inbox Transport

Field	Description
Transport Name	Specify the name of the new transport.
URL	When a message related to this partner is stored in the Inbox of BusinessConnect database, BusinessConnect sends a notification email to the following address: mailto: John@Acme.com
Subject	Specify a subject for the email message.

After you finish configuring the new Inbox transport, it is displayed in **Participants > partner > Protocols > protocol_name > Transports** as a transport of the Inbox type that is available for the partner.

Defining Timeouts for the Inbox Transport

BusinessConnect honors Asynchronous Request timeouts when uploading requests to the Inbox of BusinessConnect storage to be retrieved by TIBCO PartnerExpress.

For the uploaded Notify and Response operations, BusinessConnect uses the timeout values specified in the following properties:

- `bc.inbox.notify.payload.Timeout`
- `bc.inbox.response.payload.Timeout`

By default, the two properties are set to 1440 minutes, which means that after the Notify or Response transaction payload resides in Inbox for 1440 minutes, it is deleted. But you can also specify the timeout values in TIBCO Administrator by clicking **BusinessConnect > System Settings > Activated Protocol Plug-ins and Properties > BC > Inbox Transport Settings**.

Customizing Product Name and Logo

You can replace the current TIBCO PartnerExpress logo and product name with your own ones. Then your product logo and name can be displayed in your TIBCO PartnerExpress web client.

To use your customized product name and logo:

- 1. Log on to TIBCO Administrator.
- 2. Go to **BusinessConnect > System Settings > Activated Protocols Plug-ins and Properties > GS-PX**. The Edit Plug-in Properties: GS-PX page is displayed.

Figure 5 Using Customized Product Name and Logo

Edit Plug-in Properties :GS-PX

AddDeleteSaveCancel

Customized Product Name

Specify a product name, Default value is empty.

Customized Product Logo

[Image2.png](#) [change](#) [remove](#)

Specify a company logo image in PNG format, and the recommended logo size is 402px*35px.

Customized System Use Statement

[This warning banner provides privacy and security notice](#)

Specify a system defined information.

SSO Settings

Oauth Client ID

SSO Provider Client ID for PX

Oauth Client Secret

SSO Provider Client Secret for PX

- 3. In the **Customized Product Name** field, enter your product name, and then click **Save**.
- 4. In the **Customized Product Logo** region:
 - If you want to display your logo in the default width and height specified by TIBCO PartnerExpress, that is 402 px × 35 px, click **change** to specify your logo image in PNG format, and then click **Save**.
 - If you want to display your logo in customized width and height, click **Add** to add two integer type of properties, they are "BC.PX.customized.productLogo.width" and "BC.PX.customized.productLogo.heigh", and then specify the two values. Click **change** to select your logo image. Click **Save**.

TIBCO PartnerExpress User's Guide

Figure 6 Customizing Logo Size

Edit Plug-in Properties :GS-PX

Add Delete Save Cancel

Customized Product Name
Specify a product name, Default value is empty.

Customized Product Logo [null](#) [change](#) [remove](#)
Specify a company logo image in PNG format, and the recommended logo size is 402px*35px.

BC.PX.customized.productLogo.width

BC.PX.customized.productLogo.height

5. In the **Customized System Use Statement** field, enter the legal warning statement which you want to display on login page, and then click **Save**.
6. To customize **SSO Settings**, refer to the document *TIBCO BusinessConnect Trading Partner Administration*, Chapter System Settings, Section Activated Protocol Plug-ins and Properties.

Chapter 3

Using TIBCO PartnerExpress on the Client Side

This chapter provides instructions for users who are connecting with TIBCO PartnerExpress by using a browser on the client side.

Topics

- [Overview, page 22](#)
- [Authenticating with the PartnerExpress Server, page 23](#)
- [Home View, page 26](#)
- [Inbox View, page 29](#)
- [History View, page 34](#)
- [PX REST APIs, page 36](#)

Overview

Exchanging files securely over the Internet is easier than ever by using TIBCO PartnerExpress client. Simply point your browser to the URL provided by your host running the PartnerExpress server, log on by using the user name and password supplied by the PartnerExpress server's administrator, and then use the simple Inbox interface to upload and download files to or from the PartnerExpress server.

You do not have to install any software on the PartnerExpress client side to run the web browser, and no software is downloaded onto your system. All communications with the PartnerExpress server is secure and user authentication is provided.

The PartnerExpress client's Inbox view lists all the files that are ready for you to download from the PartnerExpress server, as well as the pending files (including the files that both the PartnerExpress client and the PartnerExpress server have to respond to). In the PartnerExpress client's History view, you can see all your file transfer activities and their detailed information.

The supported web browsers for the PartnerExpress client are:

- Mozilla Firefox 77.0.x
- Microsoft Internet Explorer 11.x
- Apple Safari 13.1.1
- Google Chrome 83.0.x

Authenticating with the PartnerExpress Server

To log on to the PartnerExpress server from the client side, you must obtain the PartnerExpress server URL, the name of the trading partner which your client is associated with, the user name, and the password from the PartnerExpress server administrator.

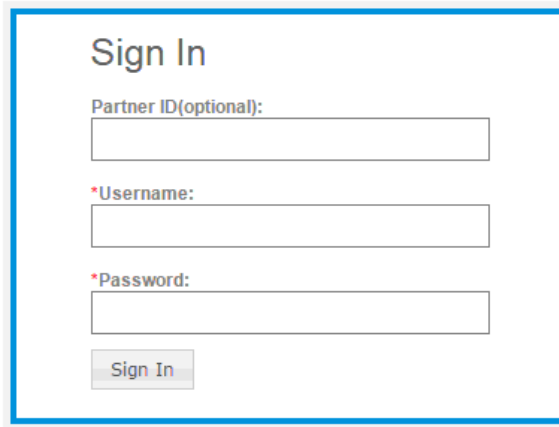
To access the PartnerExpress server from the client side, complete the following steps:

1. Open your browser and type the following URL as the location:

`https://host_name:port/px/login.html`

The authentication screen is displayed.

Figure 7 External User Authentication

The image shows a web form titled "Sign In" enclosed in a blue border. The form contains three input fields: "Partner ID(optional):", "*Username:", and "*Password:". Each field has a corresponding text input box. Below the input fields is a "Sign In" button. The asterisk (*) next to the Username and Password labels indicates they are required fields.

2. Type the following information in the authentication fields:
 - **Partner ID** (Optional) The name of the trading partner which the PartnerExpress server administrator associates your client with.
 - **Username** (Required) The name of your client that is set up by the PartnerExpress server administrator.
 - **Password** (Required) The password for your client that is set up by the PartnerExpress server administrator.
3. Click **Sign In**. The default [Home View](#) is displayed.

Signing on to PartnerExpress Using Secure URL

You can generate a secure login URL via the PX REST interface by which the user can bypass the login form and directly access the application home page.

Generate the secure login URL from the following PX endpoint:

```
https://<host_name>:6809/px/pxserver/rest/1.0/logonURL?user=<userid>
&appKey=<app_key>
```

where appKey is an alphanumeric string that is sufficiently long and random.

You must configure the SHA-256 hash value of the app key in the gengine.tra file. You can reset the key by changing the hash and restarting the gateway engine.

The response JSON payload will be in the format: { "url": "<url value>" }. You can use this generated URL to login.

Example:

```
java.property.gs.px.signon.app.key=CC556D4C3A760BB118F4F033951EC5D
A996877CF4C5980BE65F9B8004554CDA1
```

```
https://machinename:6809/px/pxserver/rest/1.0/logonURL?user=joe@foo.
com&appKey=jwe98hs30g8pm01he
```

```
{"url":
```

```
"https://machinename:6809/px/loginLink.html?auth=r00ABXQAWXmNyZWf0aW
9uVGltZT0xNTkxMTE4MTIyMTAz"} }
```

The generated URL is time bound with configurable expiry and allows you to login for only one time.



By default the generated link expires in 10 seconds, but you can configure the expiry time in the gengine.tra file by adding the below property:

```
java.property.gs.px.signon.link.expiry=<secs value>
```

For example, java.property.gs.px.signon.link.expiry=20

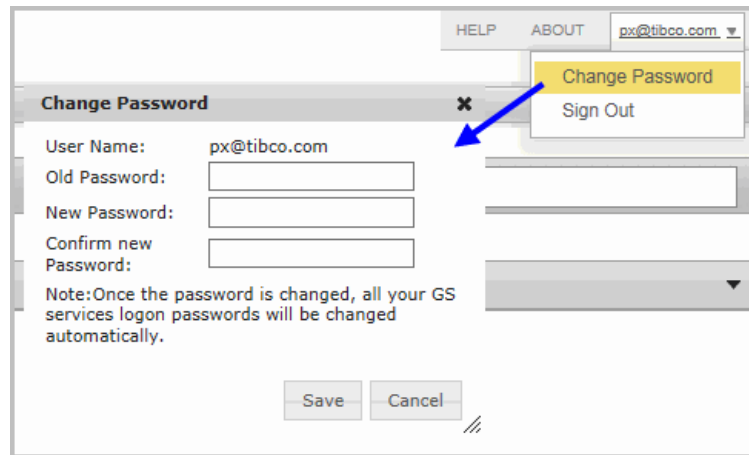
Changing Logon Password

After you log on to the PartnerExpress client, you can change your logon password.

To change your logon password:

1. Log on to your PartnerExpress client.
2. In the upper-right corner, click your logon user name, and then click **Change Password**.
3. In the **Change Password** window, enter your old password, your new password, and the new password again. Click **Save**.

Figure 8 Changing Password

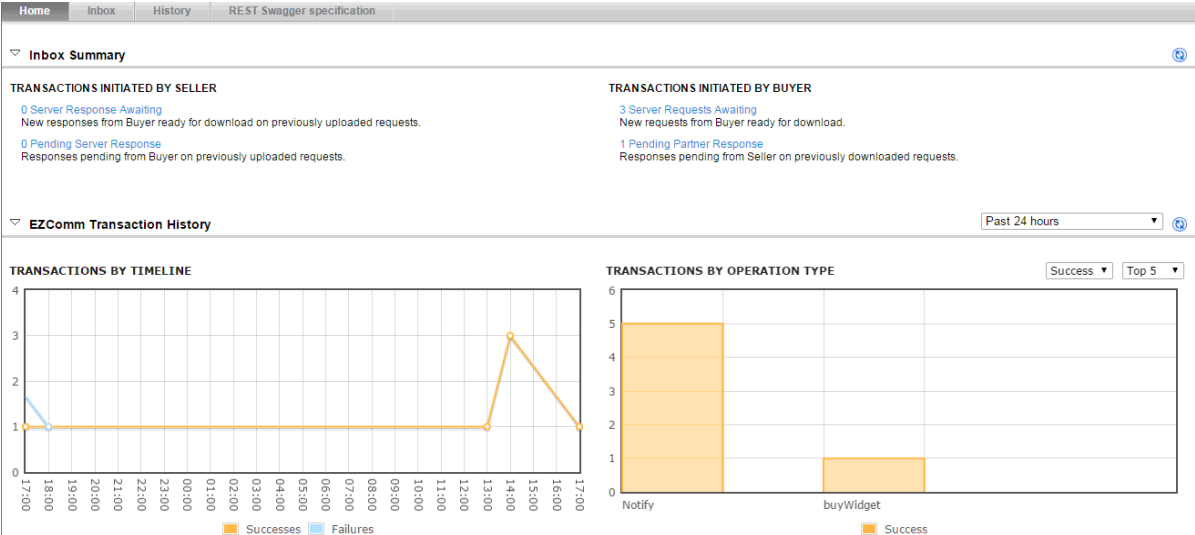


After the password is changed, all the logon passwords of your GS services, such as TCM, SFTP, and FTPS, are changed automatically.

Home View

After you are authenticated with the PartnerExpress server successfully, the Home view is displayed by default.

Figure 9 Home View



The Home view is made up of two panels: Inbox Summary and Transaction History.

Inbox Summary

The Inbox Summary panel contains the following categories of transaction summaries:

For transactions initiated by the partner:

- **Server Request Awaiting** Shows the number of PartnerExpress server requests ready for the partner to download.
- **Server Response Awaiting** Shows the number of PartnerExpress server responses to the previously uploaded requests by the partner.

For transactions initiated by the PartnerExpress server:

- **Pending Server Response** Shows the number of responses pending from the PartnerExpress server on previously uploaded requests by the partner.

- **Pending Partner Response** Shows the number of responses pending upload from the partner on previously downloaded requests from the PartnerExpress server.

Transaction History

In the Transaction History panel, you can view transaction history by timeline or operation type. This panel can display one or more protocols, and each protocol is displayed on one tab.

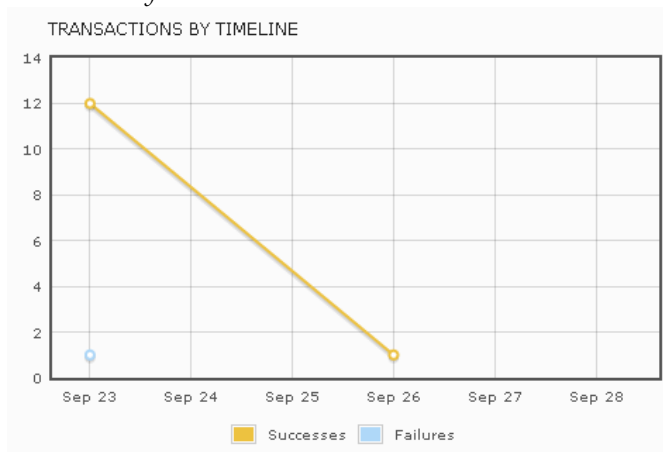


In this release, TIBCO BusinessConnect™ Services Plug-in and TIBCO BusinessConnect™ EDI Protocol powered by Instream are supported.

- **Transactions by Timeline** You can select to display the transactions that happened in the past 24 hours, past week, or past month.

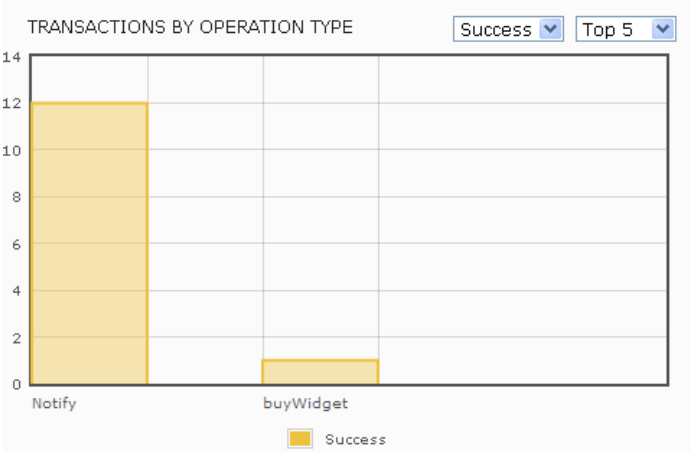
The following chart shows the date and number of the transactions. It also shows how many successful and failed transactions there are in the selected period.

Figure 10 Transactions by Timeline



- **Transactions by Operation Type** You can also select to only display successful transactions, failed transactions, the top 5 or top 10 most frequently used transaction types.

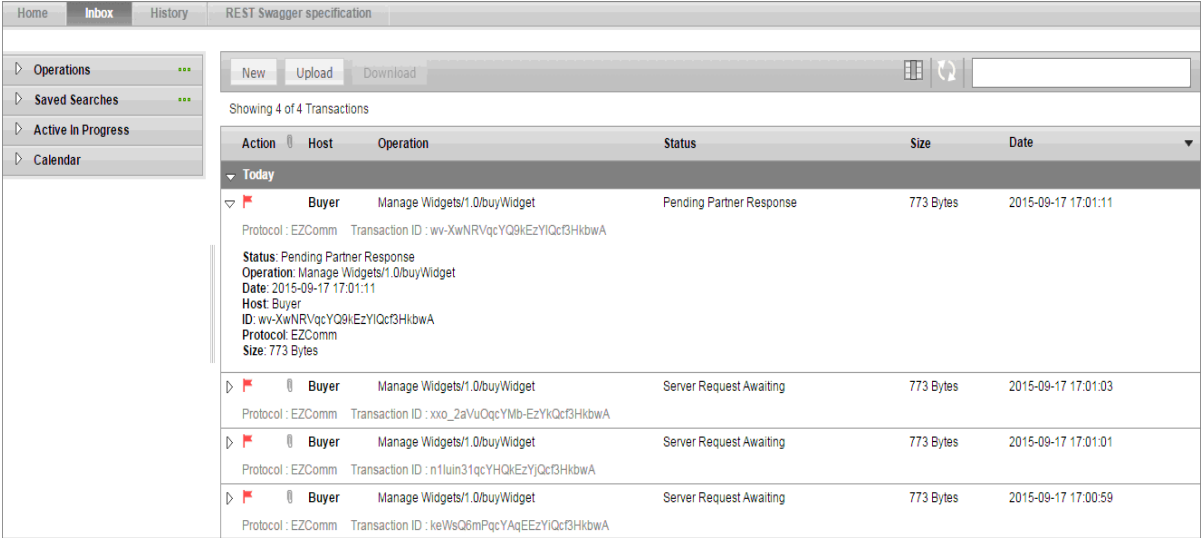
Figure 11 Transactions by Operation Type



Inbox View

Inbox is the main interface for uploading transaction files, downloading transaction files, and searching transactions. It consists of a side window on the left, and the main window on the right where transactions are listed.

Figure 12 Inbox View



Side Window

The side window contains the following palettes:

- Operations
- Saved Searches
- Active in Progress
- Calendar

All the options in these palettes can be used as query filters in Inbox.

Operations

You can expand the Operations palette to select which transactions and operations you want to display in the main window.




Saved Searches

In the Saved Searches palette, you can create a new search, save the search for reuse, view and edit the details of prebuilt and your customized saved searches.

Creating and saving a new search

You can create a new search and save it for reuse.

To create and save a new search:

- 1. In the Saved Searches palette, click the  icon, and then click **Create a new saved search**.
- 2. In the Advanced Search window, type the name for the search, and select the protocol name.
- 3. Click the  icon or the  icon to add or remove conditions for the new search.

For more information about the search criteria, see the following table:

Table 3 Advanced Search Criteria

Attribute	Operator
ID	<ul style="list-style-type: none">• Equals
Host	<ul style="list-style-type: none">• Like• Not equals
Operation	<ul style="list-style-type: none">• Not like
Status	<ul style="list-style-type: none">• Equals• Not equals
Size	<ul style="list-style-type: none">• Equals• Not equals
Date	<ul style="list-style-type: none">• Less than• Greater than






There cannot be more than one Equals operator for the same attribute. For example, ID equals 1 AND ID equals 2 is not valid.

4. After you finish setting up the conditions, click one of the following buttons:
 - **Search** Starts the search and displays the search results. The search is not saved.
 - **Save search and view results** Starts the search, displays the search results, and saves the search.

Viewing and editing saved searches

You can view and edit the details of prebuilt searches and your customized searches.

To view or edit saved searches:

1. In the Saved Searches palette, click the  icon, and then click **Show saved search with details**. The Saved Searches window is displayed.
 In the Saved Searches window, there are four types of prebuilt searches, which cannot be deleted or modified, they are Pending Partner Response, Pending Server Response, Server Response Awaiting, and Server Request Awaiting.
2. Click a search, and then its detailed conditions are displayed.
3. If you want to edit the conditions for a search, click the  icon or the  icon to add or remove conditions.
4. If you want to delete a search, select it, and then click **Delete selected**.

Active In Progress

This palette lists all the uploads and downloads that are in progress for the current logon session. When you click **ACTIVE UPLOADS** or **ACTIVE DOWNLOADS**, the Uploads and Downloads window is displayed. In this window, the active downloads or uploads, including both the transfers that are completed and the ones that are in progress currently, are listed.

Calendar

In this palette, you can select a date to view the operations that are performed in the selected date.

Main Window

In the main window, you can create new transactions, upload or download transaction files, view different sorts of information about transactions, refresh a list of transactions, and search transactions.

Creating a New Transaction

To create a new transaction:

1. Click **New**, and then the Create New Transaction window is displayed.
2. Select an operation type, **Notify** or **Request/Response**.
3. Click **Browse** to specify the file to upload.
4. Click **Upload** to upload the file, or click **Cancel** to exit.

Uploading a Response File

The **Upload** button is only available when the status of the selected transaction is `Pending partner response`.

Click this button to specify and upload a file in response to a previously downloaded server request.


Downloading a Transaction File

The **Download** button is only available when the status of the selected transaction is either `Server Request Awaiting` or `Server Response Awaiting`.

Click this button to download a server request, or to download a response by the PartnerExpress server in response to the previous partner's request.

A transaction is downloaded and moved to the History view only when a user downloads a Notify type server request initiated by the PartnerExpress server (Figure 2) or a Request/Response type server response initiated by the partner (Figure 3).


Selecting Transaction Information to Display

You can select to view different columns of information about transactions by clicking the **Pick Columns** icon .

The following columns of information can be displayed:

- Action
- Host
- Operation
- Protocol
- Size
- Date
- Status
- Transaction ID

Refreshing the Transaction List

You can click the  icon to refresh the list of transactions displayed in the main window.

Searching Transactions

To search transactions:

1. Type the item you are searching for in the query field in the upper-right corner. The following query categories are displayed:
 - Search *item_name* in Host
 - Search *item_name* in Status
 - Search *item_name* in Protocol Name
 - Search *item_name* in Operation
 - Search *item_name* in Transaction ID
2. Click the category your search item belongs to. The search results are displayed in the main window.

History View

The History view lists all the completed and failed transactions, and the summary of them.

Figure 13 History View

Home	Inbox	History	REST Swagger specification
Operations	Saved Searches	Calendar	
Showing 20 of 41 Transactions			
Operation	Status	Date	
Today			
Manage Widgets/1.0/buyWidget	COMPLETED	2015-09-17 17:01:10	
Protocol : EZComm Transaction ID : wv-XwNRVqcYQ9kEzYIqc3HkbwA			
BC/1.0/Notify	COMPLETED	2015-09-17 14:08:40	
Protocol : EZComm Transaction ID : 9A2777C6-6091-49E3-96B7-5487D5C8DEAB			
BC/1.0/Notify	COMPLETED	2015-09-17 14:07:59	
Protocol : EZComm Transaction ID : 5DFFFCFCE2-468B-48F3-BCCF-BA8F548C53A4			
BC/1.0/Notify	COMPLETED	2015-09-17 14:03:09	
Protocol : EZComm Transaction ID : CDF72E58-02A9-48BE-A762-57E7086B0CD8			
BC/1.0/Notify	COMPLETED	2015-09-17 13:37:18	
Protocol : EZComm Transaction ID : BA00F9A6-0BE5-43A5-8C85-5ECE7037A1E5			
Yesterday			
Manage Widgets/1.0/buyWidget	ERROR	2015-09-16 18:11:32	
Protocol : EZComm Transaction ID : FA2E7CB0-A7E3-40E6-B09A-B3E760ED8F37			
BC/1.0/Notify	COMPLETED	2015-09-16 17:03:42	
Protocol : EZComm Transaction ID : 384398F8-AC33-44B9-B22C-DD67FDB02DC4			
BC/1.0/Notify	ERROR	2015-09-16 15:04:02	
Protocol : EZComm Transaction ID : ZyhnmlBnx1VUkEzYbQc3HkbwA			

Side Window


The side window contains the following palettes:

- **Operations** You can expand the Operations palette to select which transactions and operations you want to display in the main window.
- **Saved Searches** You can create a new search, save the search for reuse, and view or edit the details of your customized saved searches. For more information, see [Saved Searches](#) in the Inbox view.
- **Calendar** You can select a date to view the operations that are performed in the selected date.

Main Window

In the main window, you can create new transactions, upload or download transaction files, view different sorts of information about transactions, refresh the list of transactions, and search transactions.


Selecting Transaction Information to Display

You can select to view different columns of information about transactions by clicking the **Pick Columns** icon .

The following columns of information can be displayed:

- Operation
- Status
- Date
- Protocol
- Transaction ID

Refreshing the Transactions List

You can click the  icon to refresh the list of transactions displayed in the main window.

Searching Transactions

To search transactions:

1. Type the item you are searching for in the query field in the upper-right corner. The following query categories are displayed:
 - Search *item_name* in Host
 - Search *item_name* in Status
 - Search *item_name* in Protocol Name
 - Search *item_name* in Operation
 - Search *item_name* in Transaction ID
2. Click the category your search item belongs to. The search results are displayed in the main window.

PX REST APIs

All the operations in TIBCO PartnerExpress are exposed as RESTful APIs that can be invoked over HTTP protocol. The schema of these RESTful APIs are described by using Swagger specification in JSON format.

After you log on to TIBCO PartnerExpress, click the **REST Swagger Specification** tab, and then you can find the definitions of all the REST APIs in TIBCO PartnerExpress.

Figure 14 REST Swagger Specification View

```
{
  "swagger": "2.0",
  "basePath": "/px/pxserver/rest/1.0",
  "schemes": ["https"],
  "info": {
    "title": "Tibco BusinessConnect Partner Express (PX) REST API",
    "description": "REST interface to Partner Express (PX)",
    "contact": {
      "name": "Tibco Software Inc",
      "url": "http://www.tibco.com/support",
      "email": "support@tibco.com"
    }
  },
  "version": "6.1.0"
},
{
  "paths": {
    "/logout": {
      "get": {
        "description": "",
        "produces": [
          "application/json"
        ],
        "operationId": "logout",
        "parameters": [
          {
            "name": "session",
            "in": "query",
            "required": true,
            "description": "\"type\": \"string\"",
            "type": "string"
          }
        ],
        "responses": {
          "200": {
            "description": "Success",
            "schema": {
              "$ref": "#/definitions/logoutResponse"
            }
          },
          "400": {
            "description": "Error processing request",
            "schema": {
              "$ref": "#/definitions/Error"
            }
          }
        }
      }
    }
  }
}
```

Swagger

Swagger is an open source project for representing RESTful APIs. Swagger provides specifications for defining RESTful APIs, and also provides a browser based Swagger UI on which you can test and interact with the Swagger-compliant APIs while checking generated documentation.

For more information about Swagger specifications, go to <http://swagger.io/>.

JSON

JSON, JavaScript Object Notation, is a language-independent data format designed for interchanging data objects in human-readable text.

In JSON, data objects are represented by name/value pairs, which usually take the form of Object, Array, Value, String, and Number.

For more information about JSON format, go to <http://json.org/>.

Invoking PX REST APIs by Using Clients

REST being an open standard, REST APIs can be invoked through a variety of tools and frameworks regardless of whether they natively interpret Swagger specification. For example, you can write raw HTTP client code by using the Swagger specification as a guiding tool for the request and response formats, and then use tools like the official Swagger UI and SOAP UI to interpret the Swagger specification and present the customized UI. There are also tools that you can use to generate programming language specific bindings out of the Swagger specification.

This section provides instructions on how to invoke PX REST APIs by using Swagger UI Client and Swagger Java Client.

You can disable the REST interface by setting the below property in the `gsengine.tra` file:

```
java.property.gs.px.rest.disable=true
```

Invoking PX REST APIs by Using Swagger UI Client

Swagger UI Client is a tool that takes Swagger specification files and displays them visually. The RESTful operations in PX can be invoked by using Swagger UI Client which can show the detailed representation of each operation using the PX REST Swagger specification.

Complete the following steps to invoke PX REST APIs in Swagger UI Client:

1. Go to <https://github.com/swagger-api/swagger-ui>, and click **Download ZIP** to download Swagger UI Client.

2. Extract the downloaded file into a local folder, and find the **dist** folder in the extracted folder. Open the **index.html** page with your browser.
3. Replace `http://petstore.swagger.io/v2/swagger.json` with `https://host_name:port/px/pxserver/rest/1.0/swagger`, and then click **Explore**. The PX REST APIs are displayed.
4. Click **List Operations** to select an operation, enter the required parameters, and then click **Try it out** to execute the operation.

Figure 15 PX REST APIs in Swagger UI Client



Invoking PX REST APIs by Using Swagger Java Client

Swagger also supports various programming language clients. The stubs for these programming languages can be generated at the <http://editor.swagger.io/> website by entering the Swagger specification for applications.



Below steps are only for sample reference. Check the official Swagger website for latest documentation.

The following sample demonstrates this approach for Java which is based on the open source Jersey framework.

1. Go to <http://editor.swagger.io/>.

2. Generate the Java stubs package by completing the following steps:
 - a. Click **File > Import URL / Import file**, to import the Swagger specification for PX.
 - b. Click **Generate Client > Java**.
3. Save the `java-client-generated.zip` file to your local file system, extract it, and then navigate to the `java-client\src\main\java\io\swagger\client` directory. Open the `ApiClient.java` file.

4. Change the `getClient()` method as follows so that the underlying Jersey framework can be configured correctly to send multipart requests that are required for file upload.

```
private Client getClient() {
    if(!this.hostMap.containsKey(this.basePath)) {
        DefaultClientConfig config = new DefaultClientConfig();
        config.getClasses().add(com.sun.jersey.multipart.impl.Mul
tiPartWriter.class);
        Client client = Client.create(config);
        if(this.debugging) {
            client.addFilter(new LoggingFilter());
        }
        this.hostMap.put(this.basePath, client);
    }
    return (Client)this.hostMap.get(this.basePath);
}
```

5. Compile the extracted Java client source. The easiest way to do this is to invoke Maven by using the default POM file. For example,
`c:\TEMP\java-client>C:\apache-maven-3.3.3\bin\mvn compile`
6. Download the following third-party dependencies from the official Maven repository at <http://mvnrepository.com/>:
 - Jersey Client 1.18
 - Jersey Multipart 1.18
 - Jackson Core 2.4.2
 - Jackson Datatype Joda 2.1.5
 - Joda Time 2.3
7. Using the compiled Java client stub classes and the third-party JARs in the classpath to compile and execute the following sample Java test program:

```
import io.swagger.client.ApiClient;
```

```

import io.swagger.client.api.DefaultApi;
import io.swagger.client.model.GetInboxTransactionListResponse;
import io.swagger.client.model.InboxTransactionRow;
import io.swagger.client.model.LogonResponse;
import io.swagger.client.model.UploadFileResponse;
import javax.net.ssl.HostnameVerifier;
import javax.net.ssl.HttpsURLConnection;
import javax.net.ssl.SSLSession;
import java.io.File;
import java.io.FileOutputStream;
import java.util.List;

public class SwaggerRESTClient {
    public static void main(String args []) throws Exception {
        System.setProperty("javax.net.ssl.trustStore",
            "BC_HOME\\samples\\keys\\bc_trustcacertstore.jks");
        HttpsURLConnection.setDefaultHostnameVerifier(new
            HostnameVerifier() {
                public boolean verify(String s, SSLSession sslSession) {
                    return true;
                }
            });
        ApiClient client = new ApiClient();
        client.setBasePath("https://host_name:port/px/pxserver/rest/1.0");
        DefaultApi defaultApi = new DefaultApi(client);
        // Logon
        LogonResponse logonResponse = defaultApi.logon("user", "user",
            "partner");
        String sessionId = logonResponse.getSession();
        System.out.println("Session ID = " + sessionId);
        // File upload
    }
}

```

```

        File file = new
File("BC_HOME\\protocols\\tibedi\\samples\\sampleDocs\\X12_4030_8
50.dat");

        UploadFileResponse fileResponse = defaultApi.uploadFile(file, "X12",
"EDI/Inbound/Interchange", sessionId, null, null);

        System.out.println("File upload status = " + fileResponse.getResult());

        // Get pending downloads

        GetInboxTransactionListResponse ibTxnListResponse =
defaultApi.getInboxTransactionList("0", "100", "[", "]", sessionId);

        List<InboxTransactionRow> inboxTransactionRows =
ibTxnListResponse.getRows();

        if(inboxTransactionRows != null) {

            for(InboxTransactionRow row : inboxTransactionRows)

                // Issue file download

                if("Server Request Awaiting".equals(row.getSTATUS()) || "Server
Response Awaiting".equals(row.getSTATUS())) {

                    System.out.println("Downloading file for transaction id = " +
row.getBCTRANSID());

                    String fileStr = defaultApi.downloadFile(row.getUserTRANSID(),
row.getBCTRANSID(), row.getPROTOCOLNAME(),
row.getOPERATIONID(), sessionId);

                    byte[] fileBytes = fileStr.getBytes();

                    FileOutputStream fileOutputStream = new
FileOutputStream("c:/temp/filedownload" + row.getBCTRANSID() + ".dat");

                    fileOutputStream.write(fileBytes);

                    fileOutputStream.close();

                }

            }

            // Logout

            defaultApi.logout(sessionId);

            System.out.println("Logout, exiting...");

        }

    }

```

PX REST Operations

This section provides an overview of all the RESTful operations in TIBCO PartnerExpress. These RESTful operations are categorized according to the HTTP methods they use, including GET, POST, PUT, and DELETE.

Table 4 Overview of the RESTful Operations

Operation	Description	Resource	Request Parameter	Response
GET Method				
Log on	Logs on to the PX server.	/logon	partner : the partner name that you are associated with. This parameter is optional, and you can also leave it blank. user : the name of the client that is set up by the PartnerExpress server administrator. password : the password for the client that is set up by the PartnerExpress server administrator.	Returns the session ID that can be used to authenticate all the subsequent requests.
Logon URL	Provides the URL through which you can login to the PX UI	/logonurl	user : the name of the client that is set up by the PartnerExpress server administrator. appKey : the alphanumeric string that is provided by the PartnerExpress server administrator.	Returns the logonURL by which the user can bypass the login form and directly access the application home page.
Log off	Logs off from the PX server.	/logout	session ID : this value is returned by the GET/logon operation.	Returns the status result of this operation, that is success or error .
List queries	Gets the list of all the user-defined queries.	/querylist	type : the type of the query, which is history or inbox . session ID : this value is returned by the GET/logon operation.	Returns the list of saved queries and their attributes, including query name, type, and search conditions.

Operation	Description	Resource	Request Parameter	Response
Get history transactions list	Gets the list of historical transactions up to now.	/historytransactionslist	<p>startRow: this value specifies that transactions from which row on are returned by the response. For example, if the startRow is 2, the response returns transactions from row 3 on.</p> <p>maxRow: this value specifies the maximum rows of transactions that can be returned in the response.</p> <p>sort: this value specifies what sort of information about the transactions is returned by the response.</p> <p>conditions: the query conditions, such as [{"name": "PROTOCOL_NAME", "operator": "EQUALS", "value": "string"}]. If timestamp (TS) is used in the conditions, it has to be in the "yyyy-mm-dd" format, such as "2015-09-10".</p> <p>session ID: this value is returned by the GET/logon operation.</p> <p>Note: When timestamp (TS) is used in the conditions, for operator "EQUALS", all the transactions that occur within that day are returned; while for operator "NOT_EQUALS", all the transactions that did not occur within that day are returned.</p>	Returns the list of all the historical transactions and their detailed information, including the protocol name, trading partner name, user transaction ID, BC transaction ID, status, timestamp, and operation ID.
Get history transaction summary	Gets the summary of one historical transaction.	/historytransactionssummary	<p>transaction ID: this is the bc transaction ID.</p> <p>session ID: this value is returned by the GET/logon operation.</p>	Returns detailed information about the historical transaction with the specified transaction ID, such as the protocol name, operation ID, and transaction ID.

Operation	Description	Resource	Request Parameter	Response
Get Inbox transactions list	Gets the list of all the pending transactions in the current session.	/inboxtransactionlist	<p>startRow: this value specifies that transactions from which row on are returned by the response. For example, if the startRow is 2, the response returns transactions from row 3 on.</p> <p>maxRow: this value specifies the maximum rows of transactions that can be returned in the response.</p> <p>sort: this value specifies what sort of information about the transactions is returned by the response.</p> <p>conditions: the query conditions, such as [{ "name" : "PROTOCOL_NAME", "operator": "EQUALS", "value": "string" }]. If timestamp (TS) is used in the conditions, it has to be in the "yyyy-mm-dd" format, such as "2015-09-10".</p> <p>session ID: this value is returned by the GET/logon operation.</p> <p>Note: When timestamp (TS) is used in the conditions, for operator "EQUALS", all the transactions that occur within that day are returned; whereas for operator "NOT_EQUALS", all the transactions that did not occur within that day are returned.</p>	Returns the list of all the Inbox transactions and their detailed information, including the protocol name, processing status, user transaction ID, BC transaction ID, action, TS, size, status, operation ID, user name, and attachment status.
Get Inbox transaction summary	Gets the summary of one Inbox transaction.	/inboxtransactionsummary	<p>transaction ID: this is the bc transaction ID.</p> <p>session ID: this value is returned by the GET/logon operation.</p>	Returns detailed information about the Inbox transaction with the specified transaction ID, such as the protocol name, operation ID, and transaction ID.
Download upload info	Gets the summary of all the transactions in the current session.	/downloaduploadinfo	session ID: this value is returned by the GET/logon operation.	Returns a summary of downloads and uploads information.

Operation	Description	Resource	Request Parameter	Response
Download file	Downloads files from the PX server Inbox.	/file	user_transaction_id , bc_transaction_id , protocol , operation : values of these parameters are returned by the GET/inboxtransactionlist operation. session ID : this value is returned by the GET/logon operation.	Returns the file content. The content type of the HTTP response is <code>application/octet-stream</code> and content disposition is <code>attachment; filename=xxx</code> .
Get operations	Gets all types of operations belonging to the partner's business agreement protocol bindings.	/operations	all : if this value is <code>true</code> , for each protocol, all operations for business transactions are fetched; if this value is <code>false</code> , for each protocol, all operations for file upload transactions are fetched. For EDI protocols, only the "EDI/Inbound/Interchange" or "EDI/Inbound" operations for file upload transactions are fetched when this value is <code>false</code> . session ID : this value is returned by the GET/logon operation.	Returns the list of protocols bound under the partner's business agreement and the operations under those protocols.

POST Method

Operation	Description	Resource	Request Parameter	Response
Create query	Creates a new query.	/query	<p>name: the name of the query.</p> <p>conditions: the query conditions, such as [{"name": "OPERATION_ID", "value": "850", "operator": "EQUALS"}]</p> <p>type: the type of the query, that is inbox or history.</p> <p>session ID: this value is returned by the GET/logon operation.</p> <p>Note: The Swagger schema type "QueryField" lists the valid fields for query conditions. For query type inbox, use the fields PROTOCOL_NAME, USER_TRANS_ID, GSUSER, STREAM_SIZE, OPERATION_ID, STATUS, and TS as the conditions. For query type history, use the fields PROTOCOL_NAME, USER_TRANS_ID, OPERATION_ID, STATUS, and TS as the conditions.</p>	Returns the object ID (oid) of the created query. This oid value is required when you delete or update the query.

Operation	Description	Resource	Request Parameter	Response
Upload file	Uploads a transaction file to the PX server.	/file	<p>The file content is embedded as multipart form data. The content type of the HTTP request is <code>multipart/form-data</code>, and the content type of the part containing the file is <code>application/octet-stream</code>. The remaining request parameters are sent as regular query string parameters like the parameters of other operations.</p> <p>file-upload: if you want to upload a file, specify a file for uploading; if not, you can leave it blank.</p> <p>user_transaction_id: leave it blank, or get this value from the response returned by the GET/inboxtransactionlist operation.</p> <p>bc_transaction_id: leave it blank, or get this value from the response returned by the GET/inboxtransactionlist operation.</p> <p>protocol: the protocol name, such as EZComm.</p> <p>operation: the operation ID, such as BC/1.0/Notify.</p> <p>session ID: this value is returned by the GET/logon operation.</p>	Returns the status result of this upload operation, that is success or error.
PUT Method				
Update query	Updates a query.	/query	<p>oid: the ID of the query object, which is returned by the POST/query operation.</p> <p>conditions: enter your new search conditions.</p> <p>session ID: this value is returned by the GET/logon operation.</p> <p>Note: For information about the valid fields for the conditions, refer to the create query operation.</p>	Returns the status result of this operation, that is success or error.

Operation	Description	Resource	Request Parameter	Response
Change password	Changes the logon password.	/password	oldpassword: your old logon password. newpassword: your new logon password. session ID: this value is returned by the GET/logon operation.	Returns the status result of this operation, that is success or error.
DELETE Method				
Delete query	Deletes one or more queries.	/query	oid: the array of query object IDs to be deleted. session ID: this value is returned by the GET/logon operation.	Returns the status result of this operation, that is success or error.

HTTP Wire Format

This section provides some examples of how the REST requests and responses are transported over HTTP.

All query parameter values have to be encoded appropriately for HTTP URL.

In general, all request parameters are passed in as HTTP query string parameter values. The only exception is file upload, where in addition to query string parameters, the file content is presented in the request body as a multipart form file control.

Similarly, all responses are JSON-formatted texts in the HTTP response body except for file download, where the file content is embedded in the response body.

Logon

The following table provides an example of the HTTP request and response messages for the logon operation.

HTTP Message	Example
Request	<pre>GET /px/pxserver/rest/1.0/logon?partner=partner&user=user&password=user HTTP/1.1 Host: host_name:6809 Connection: keep-alive Accept: application/json Origin: null User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/43.0.2357.134 Safari/537.36 Accept-Encoding: gzip, deflate, sdch Accept-Language: en-US,en;q=0.8</pre>
Response	<pre>HTTP/1.1 200 OK Server: Apache-Coyote/1.1 Set-Cookie: JSESSIONID=2FEC880BAE1942041D28A5FCFEECCB2F; Path=/px/; Secure; HttpOnly Access-Control-Allow-Origin: * Access-Control-Allow-Methods: GET, PUT, POST, DELETE, OPTIONS Content-Type: application/json;charset=ISO-8859-1 Content-Length: 124 Date: Tue, 28 Jul 2015 20:27:22 GMT {"session":"4FF43A7A-2999-4DD7-A806-00B7489903C9","user":{"readOnly":false,"host":"host","name":"user","partner":"partner"}}</pre>

File Upload

The following table provides an example of the HTTP request and response messages for the file upload operation.

HTTP Message	Example
Request	<pre>POST /px/pxserver/rest/1.0/file?protocol=X12&operation=EDI%2FInbound%2 FInterchange&session=4FF43A7A-2999-4DD7-A806-00B7489903C9 HTTP/1.1 Host: host_name:6809 Connection: keep-alive Content-Length: 16122 Accept: application/json Origin: null User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/43.0.2357.134 Safari/537.36 Content-Type: multipart/form-data; boundary=----WebKitFormBoundaryB00DmHvaOwmy1N1r Accept-Encoding: gzip, deflate Accept-Language: en-US,en;q=0.8 -----WebKitFormBoundaryB00DmHvaOwmy1N1r Content-Disposition: form-data; name="file-upload"; filename="X12_5010_850.dat" Content-Type: application/octet-stream <bytes> -----WebKitFormBoundaryB00DmHvaOwmy1N1r--</pre>
Response	<pre>HTTP/1.1 200 OK Server: Apache-Coyote/1.1 Set-Cookie: JSESSIONID=FDB3E1DCC07E78EB0CCBB6710D195FF2; Path=/px/; Secure; HttpOnly Access-Control-Allow-Origin: * Content-Type: application/json;charset=ISO-8859-1 Content-Length: 20 Date: Tue, 28 Jul 2015 20:32:16 GMT {"result":"success"}</pre>

File Download

To download a file, you have to perform the GET/inboxtransactionlist operation first to get the transaction ID, and then perform the file download operation.

The following table provides an example of the HTTP request and response messages for the GET/inboxtransactionlist operation:

HTTP Message	Example
Request	<pre>GET /px/pxserver/rest/1.0/inboxtransactionlist?startRow=0&maxRow=100& sort=%5B%5D&conditions=%5B%5D&session=4FF43A7A-2999-4DD7-A806-00B 7489903C9 HTTP/1.1 Host: host_name:6809 Connection: keep-alive Accept: application/json Origin: null User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/43.0.2357.134 Safari/537.36 Accept-Encoding: gzip, deflate, sdch Accept-Language: en-US,en;q=0.8</pre>

HTTP Message	Example
Response	<pre>HTTP/1.1 200 OK Server: Apache-Coyote/1.1 Set-Cookie: JSESSIONID=888133499ACC3E8B439C650AD4571DE5; Path=/px/; Secure; HttpOnly Access-Control-Allow-Origin: * Access-Control-Allow-Methods: GET, PUT, POST, DELETE, OPTIONS Content-Type: application/json;charset=ISO-8859-1 Content-Length: 354 Date: Tue, 28 Jul 2015 20:48:22 GMT { "conditions": [], "totalRows": 1, "rows": [{ "PROTOCOL_NAME": "EZComm", "PROCESSING": false, "USER_TRANS_ID": "hLbWRRGOpal0FURF1wD21BKknas", "BC_TRANS_ID": "2C8E4C84-3D5A-4B95-8FC5-CF5BF66C8881", "ACTION": "downloadServerRequest", "TS": "2015-07-28 13:48:15", "SIZE": 4, "STATUS": "Server Request Awaiting", "OPERATION_ID": "BC/1.0/Notify", "USER": "host", "ATTACHMENT": true }] }</pre>

The following table provides an example of the HTTP request and response messages for the file download operation:

HTTP Message	Example
Request	<pre>GET /px/pxserver/rest/1.0/file?user_transaction_id=hLbWRRGOpal0FURF1w D21BKknas&bc_transaction_id=2C8E4C84-3D5A-4B95-8FC5-CF5BF66C8881& protocol=EZComm&operation=BC/1.0/Notify&session=4FF43A7A-2999-4DD 7-A806-00B7489903C9 HTTP/1.1 Host: host_name:6809 Connection: keep-alive Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp, */*;q=0.8 User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/43.0.2357.134 Safari/537.36 Accept-Encoding: gzip, deflate, sdch Accept-Language: en-US,en;q=0.8 Cookie: JSESSIONID=80E3808005F2361F504D673BC6543AB0; testName=testVal; px_sessionId=C79946D3-168C-47A7-A7F2-05DCA9DE6CB8; px_username=user; px_partner=partner; px_host=host</pre>
Response	<pre>HTTP/1.1 200 OK Server: Apache-Coyote/1.1 Content-Disposition: attachment;filename="hLbWRRGOpal0FURF1wD21BKknas.dat" Content-Type: application/octet-stream Content-Length: 4 Date: Tue, 28 Jul 2015 20:55:31 GMT <file bytes></pre>

JSON Request Parameters Example

Some request parameters and most responses are in JSON format. This is used to represent complex request parameters as simple request parameters when they are passed as query string parameters. For complex request parameters that are JSON-encoded, additional encoding has to be made to make them suitable for being passed in the HTTP query string URL. The JSON schema of the complex request parameters are specified in [PX REST Swagger Specification](#).

The following is an example of a complex JSON request schema and its specific parameters:

```
"type": "array", "items": { "$ref": "#/definitions/Condition" }
[
  {
    "name": "PROTOCOL_NAME",
    "value": "X12",
    "type": "STRING",
    "operator": "EQUALS"
  },
  {
    "name": "OPERATION_ID",
    "value": "EDI/Inbound/Interchange",
    "type": "STRING",
    "operator": "EQUALS"
  }
]
```

Because Swagger only supports string type of request parameters, the complex JSON schema is described in the **Description** field.

PX REST Swagger Specification

This section provides the Swagger specification of the PX REST APIs.

```
{
  "swagger": "2.0",
  "basePath": "/px/pxserver/rest/1.0",
  "schemes": [
    "https"
  ],
  "info": {
    "title": "Tibco BusinessConnect Partner Express (PX) REST
API",
    "description": "REST interface to Partner Express (PX)",
    "contact": {
      "name": "Tibco Software Inc",
      "url": "http://www.tibco.com/support",
      "email": "support@tibco.com"
    },
    "version": "6.2.1"
  },
}
```

```

"paths":{
  "/logon":{
    "get":{
      "description":"",
      "produces":[
        "application/json"
      ],
      "operationId":"logon",
      "parameters":[
        {
          "name":"partner",
          "in":"query",
          "required":false,
          "description":"\"type\": \"string\"",
          "type":"string"
        },
        {
          "name":"user",
          "in":"query",
          "required":true,
          "description":"\"type\": \"string\"",
          "type":"string"
        },
        {
          "name":"password",
          "in":"query",
          "required":true,
          "description":"\"type\": \"string\"",
          "type":"string"
        }
      ],
      "responses":{
        "200":{
          "description":"Success",
          "schema":{
            "$ref":"#/definitions/logonResponse"
          }
        },
        "400":{
          "description":"Error processing request",

```

```

        "schema":{
            "$ref":"#/definitions/Error"
        }
    }
}
},
"/query":{
    "post":{
        "description":"",
        "produces":[
            "application/json"
        ],
        "operationId":"createQuery",
        "parameters":[
            {
                "name":"name",
                "in":"query",
                "required":true,
                "description":"\"type\": \"string\"",
                "type":"string"
            },
            {
                "name":"conditions",
                "in":"query",
                "required":true,

                "description":"\"type\": \"array\", \"items\": {\"$ref\": \"#/definitions/QueryCondition\"}",
                "type":"string"
            },
            {
                "name":"type",
                "in":"query",
                "required":true,

                "description":"\"type\": \"string\", \"enum\": [\"history\", \"inbox\"]",
                "type":"string"
            },
            {

```

```

        "name": "session",
        "in": "query",
        "required": true,
        "description": "\"type\": \"string\"",
        "type": "string"
    }
],
"responses": {
    "200": {
        "description": "Success",
        "schema": {
            "$ref": "#/definitions/createQueryResponse"
        }
    },
    "400": {
        "description": "Error processing request",
        "schema": {
            "$ref": "#/definitions/Error"
        }
    }
}
},
"put": {
    "description": "",
    "produces": [
        "application/json"
    ],
    "operationId": "updateQuery",
    "parameters": [
        {
            "name": "oid",
            "in": "query",
            "required": true,
            "description": "\"type\": \"string\"",
            "type": "string"
        },
        {
            "name": "conditions",
            "in": "query",
            "required": true,

```

```

    "description": "\"type\": \"array\", \"items\": {\"$ref\": \"#/definitions/QueryCondition\"}\",
    \"type\": \"string\"
  },
  {
    \"name\": \"session\",
    \"in\": \"query\",
    \"required\": true,
    \"description\": \"\"type\": \"string\"\",
    \"type\": \"string\"
  }
],
\"responses\": {
  \"200\": {
    \"description\": \"Success\",
    \"schema\": {
      \"$ref\": \"#/definitions/updateQueryResponse\"
    }
  },
  \"400\": {
    \"description\": \"Error processing request\",
    \"schema\": {
      \"$ref\": \"#/definitions/Error\"
    }
  }
}
},
\"delete\": {
  \"description\": \"\",
  \"produces\": [
    \"application/json\"
  ],
  \"operationId\": \"deleteQuery\",
  \"parameters\": [
    {
      \"name\": \"oid\",
      \"in\": \"query\",
      \"required\": true,

```

```

"description": "\
"type\": \"array\", \"items\": { \"$ref\": \"#/definitions/String\" }\",
    "type": "string"
  },
  {
    "name": "session",
    "in": "query",
    "required": true,
    "description": "\
"type\": \"string\"\",
    "type": "string"
  }
],
"responses": {
  "200": {
    "description": "Success",
    "schema": {
      "$ref": "#/definitions/deleteQueryResponse"
    }
  },
  "400": {
    "description": "Error processing request",
    "schema": {
      "$ref": "#/definitions/Error"
    }
  }
}
},
"/logonurl": {
  "get": {
    "description": "",
    "produces": [
      "application/json"
    ],
    "operationId": "generateLogonURL",
    "parameters": [
      {
        "name": "user",
        "in": "query",
        "required": true,

```

```

        "description": "\"type\": \"string\"",
        "type": "string"
    },
    {
        "name": "appKey",
        "in": "query",
        "required": true,
        "description": "\"type\": \"string\"",
        "type": "string"
    }
],
"responses": {
    "200": {
        "description": "Success",
        "schema": {
            "$ref": "#/definitions/generateLogonURLResponse"
        }
    },
    "400": {
        "description": "Error processing request",
        "schema": {
            "$ref": "#/definitions/Error"
        }
    }
}
},
"/historytransactionlist": {
    "get": {
        "description": "",
        "produces": [
            "application/json"
        ],
        "operationId": "getHistoryTransactionList",
        "parameters": [
            {
                "name": "startRow",
                "in": "query",
                "required": true,
                "description": "\"type\": \"integer\"",

```

```

        "type": "string"
      },
      {
        "name": "maxRow",
        "in": "query",
        "required": true,
        "description": "\"type\": \"integer\"",
        "type": "string"
      },
      {
        "name": "sort",
        "in": "query",
        "required": true,

"description": "\"type\": \"array\", \"items\": {\"$ref\": \"#/definitions/HistoryTransactionSort\"}",
        "type": "string"
      },
      {
        "name": "conditions",
        "in": "query",
        "required": true,

"description": "\"type\": \"array\", \"items\": {\"$ref\": \"#/definitions/HistoryTransactionCondition\"}",
        "type": "string"
      },
      {
        "name": "session",
        "in": "query",
        "required": true,
        "description": "\"type\": \"string\"",
        "type": "string"
      }
    ],
    "responses": {
      "200": {
        "description": "Success",
        "schema": {
"$ref": "#/definitions/getHistoryTransactionListResponse"

```



```

        }
    },
    "400":{
        "description":"Error processing request",
        "schema":{
            "$ref":"#/definitions/Error"
        }
    }
}
},
"/inboxtransactionssummary":{
    "get":{
        "description":"",
        "produces":[
            "application/json"
        ],
        "operationId":"getInboxTransactionSummary",
        "parameters":[
            {
                "name":"transaction",
                "in":"query",
                "required":true,
                "description":"\"type\": \"string\"",
                "type":"string"
            },
            {
                "name":"session",
                "in":"query",
                "required":true,
                "description":"\"type\": \"string\"",
                "type":"string"
            }
        ],
        "responses":{
            "200":{
                "description":"Success",
                "schema":{
                    "$ref":"#/definitions/getInboxTransactionSummaryResponse"
                }
            }
        }
    }
}

```

```

    }
  },
  "400":{
    "description":"Error processing request",
    "schema":{
      "$ref":"#/definitions/Error"
    }
  }
}
},
"/logout":{
  "get":{
    "description":"",
    "produces":[
      "application/json"
    ],
    "operationId":"logout",
    "parameters":[
      {
        "name":"session",
        "in":"query",
        "required":true,
        "description":"\"type\": \"string\"",
        "type":"string"
      }
    ],
    "responses":{
      "200":{
        "description":"Success",
        "schema":{
          "$ref":"#/definitions/logoutResponse"
        }
      },
      "400":{
        "description":"Error processing request",
        "schema":{
          "$ref":"#/definitions/Error"
        }
      }
    }
  }
}

```

```

    }
  }
},
"/password":{
  "put":{
    "description":"",
    "produces":[
      "application/json"
    ],
    "operationId":"changePassword",
    "parameters":[
      {
        "name":"oldpassword",
        "in":"query",
        "required":true,
        "description":"\"type\": \"string\"",
        "type":"string"
      },
      {
        "name":"newpassword",
        "in":"query",
        "required":true,
        "description":"\"type\": \"string\"",
        "type":"string"
      },
      {
        "name":"session",
        "in":"query",
        "required":true,
        "description":"\"type\": \"string\"",
        "type":"string"
      }
    ],
    "responses":{
      "200":{
        "description":"Success",
        "schema":{
          "$ref":"#/definitions/changePasswordResponse"
        }
      }
    }
  },

```

```

        "400":{
            "description":"Error processing request",
            "schema":{
                "$ref":"#/definitions/Error"
            }
        }
    },
    "/operations":{
        "get":{
            "description":"",
            "produces":[
                "application/json"
            ],
            "operationId":"getOperations",
            "parameters":[
                {
                    "name":"all",
                    "in":"query",
                    "required":true,
                    "description":"\"type\": \"boolean\"",
                    "type":"string"
                },
                {
                    "name":"session",
                    "in":"query",
                    "required":true,
                    "description":"\"type\": \"string\"",
                    "type":"string"
                }
            ],
            "responses":{
                "200":{
                    "description":"Success",
                    "schema":{
                        "$ref":"#/definitions/getOperationsResponse"
                    }
                }
            }
        }
    },

```

```

        "400":{
            "description":"Error processing request",
            "schema":{
                "$ref":"#/definitions/Error"
            }
        }
    }
},
"/file":{
    "get":{
        "description": "",
        "produces": [
            "application/octet-stream"
        ],
        "operationId": "downloadFile",
        "parameters": [
            {
                "name": "user_transaction_id",
                "in": "query",
                "required": true,
                "description": "\"type\": \"string\"",
                "type": "string"
            },
            {
                "name": "bc_transaction_id",
                "in": "query",
                "required": true,
                "description": "\"type\": \"string\"",
                "type": "string"
            },
            {
                "name": "protocol",
                "in": "query",
                "required": true,
                "description": "\"type\": \"string\"",
                "type": "string"
            },
            {
                "name": "operation",

```

```

        "in": "query",
        "required": true,
        "description": "\"type\": \"string\"",
        "type": "string"
    },
    {
        "name": "session",
        "in": "query",
        "required": true,
        "description": "\"type\": \"string\"",
        "type": "string"
    }
],
"responses": {
    "200": {
        "description": "Success",
        "schema": {
            "type": "string"
        }
    },
    "400": {
        "description": "Error processing request",
        "schema": {
            "$ref": "#/definitions/Error"
        }
    }
}
},
"post": {
    "description": "",
    "produces": [
        "application/json"
    ],
    "consumes": [
        "multipart/form-data"
    ],
    "operationId": "uploadFile",
    "parameters": [
        {
            "name": "file-upload",

```

```

        "in": "formData",
        "required": true,
        "description": "File upload",
        "type": "file"
    },
    {
        "name": "user_transaction_id",
        "in": "query",
        "required": false,
        "description": "\"type\": \"string\"",
        "type": "string"
    },
    {
        "name": "bc_transaction_id",
        "in": "query",
        "required": false,
        "description": "\"type\": \"string\"",
        "type": "string"
    },
    {
        "name": "protocol",
        "in": "query",
        "required": true,
        "description": "\"type\": \"string\"",
        "type": "string"
    },
    {
        "name": "operation",
        "in": "query",
        "required": true,
        "description": "\"type\": \"string\"",
        "type": "string"
    },
    {
        "name": "session",
        "in": "query",
        "required": true,
        "description": "\"type\": \"string\"",
        "type": "string"
    }
}

```

```

],
"responses":{
  "200":{
    "description":"Success",
    "schema":{
      "$ref":"#/definitions/uploadFileResponse"
    }
  },
  "400":{
    "description":"Error processing request",
    "schema":{
      "$ref":"#/definitions/Error"
    }
  }
}
},
"/historytransactionssummary":{
  "get":{
    "description":"",
    "produces":[
      "application/json"
    ],
    "operationId":"getHistoryTransactionSummary",
    "parameters":[
      {
        "name":"transaction",
        "in":"query",
        "required":true,
        "description":"\"type\": \"string\"",
        "type":"string"
      },
      {
        "name":"session",
        "in":"query",
        "required":true,
        "description":"\"type\": \"string\"",
        "type":"string"
      }
    ]
  },

```



```

        "responses":{
            "200":{
                "description":"Success",
                "schema":{
                    "$ref":"#/definitions/getHistoryTransactionSummaryResponse"
                }
            },
            "400":{
                "description":"Error processing request",
                "schema":{
                    "$ref":"#/definitions/Error"
                }
            }
        }
    },
    "/downloaduploadinfo":{
        "get":{
            "description":"",
            "produces":[
                "application/json"
            ],
            "operationId":"getDownloadUploadInfo",
            "parameters":[
                {
                    "name":"session",
                    "in":"query",
                    "required":true,
                    "description":"\"type\\\":\\\"string\\\"\"",
                    "type":"string"
                }
            ],
            "responses":{
                "200":{
                    "description":"Success",
                    "schema":{
                        "$ref":"#/definitions/getDownloadUploadInfoResponse"
                    }
                }
            }
        },

```

```

        "400":{
            "description":"Error processing request",
            "schema":{
                "$ref":"#/definitions/Error"
            }
        }
    },
    "/querylist":{
        "get":{
            "description":"",
            "produces":[
                "application/json"
            ],
            "operationId":"getQueryList",
            "parameters":[
                {
                    "name":"type",
                    "in":"query",
                    "required":true,

"description":"\"type\": \"string\", \"enum\": [\"history\", \"inbox\"
]",
                    "type":"string"
                },
                {
                    "name":"session",
                    "in":"query",
                    "required":true,
                    "description":"\"type\": \"string\"",
                    "type":"string"
                }
            ],
            "responses":{
                "200":{
                    "description":"Success",
                    "schema":{
                        "$ref":"#/definitions/getQueryListResponse"
                    }
                }
            }
        }
    }
}

```

```

        "400":{
            "description":"Error processing request",
            "schema":{
                "$ref":"#/definitions/Error"
            }
        }
    },
    "/inboxtransactionlist":{
        "get":{
            "description": "",
            "produces":[
                "application/json"
            ],
            "operationId":"getInboxTransactionList",
            "parameters":[
                {
                    "name":"startRow",
                    "in":"query",
                    "required":true,
                    "description":"\"type\": \"integer\"",
                    "type":"string"
                },
                {
                    "name":"maxRow",
                    "in":"query",
                    "required":true,
                    "description":"\"type\": \"integer\"",
                    "type":"string"
                },
                {
                    "name":"sort",
                    "in":"query",
                    "required":true,
                    "description":"\"type\": \"array\", \"items\": {\"$ref\": \"#/definitions/InboxTransactionSort\"}",
                    "type":"string"
                }
            ],
            {

```

```

        "name": "conditions",
        "in": "query",
        "required": true,
        "description": "\"type\": \"array\", \"items\": {\"$ref\": \"#/definitions/InboxTransactionCondition\"}\",",
        "type": "string"
    },
    {
        "name": "session",
        "in": "query",
        "required": true,
        "description": "\"type\": \"string\"",
        "type": "string"
    }
],
"responses": {
    "200": {
        "description": "Success",
        "schema": {
            "$ref": "#/definitions/getInboxTransactionListResponse"
        }
    },
    "400": {
        "description": "Error processing request",
        "schema": {
            "$ref": "#/definitions/Error"
        }
    }
}
},
"definitions": {
    "logoutResponse": {
        "properties": {
            "result": {
                "type": "string"
            }
        }
    }
}
},

```

```

    "getOperationsResponse":{
      "properties":{
        "result":{
          "type":"array",
          "items":{
            "$ref":"#/definitions/ProtocolOperation"
          }
        }
      }
    },
    "getHistoryTransactionSummaryResponse":{
      "properties":{
        "totalRows":{
          "type":"integer"
        },
        "rows":{
          "type":"array",
          "items":{
            "$ref":"#/definitions/HistoryTransactionSummary"
          }
        }
      }
    },
    "logonResponse":{
      "properties":{
        "session":{
          "type":"string"
        },
        "user":{
          "type":"object",
          "$ref":"#/definitions/User"
        }
      }
    },
    "generateLogonURLResponse":{
      "properties":{
        "url":{
          "type":"string"
        },

```

```

        "user":{
            "type":"object",
            "$ref":"#/definitions/User"
        }
    },
    "getHistoryTransactionListResponse":{
        "properties":{
            "conditions":{
                "type":"array",
                "items":{
                    "$ref":"#/definitions/HistoryTransactionCondition"
                }
            },
            "totalRows":{
                "type":"integer"
            },
            "rows":{
                "type":"array",
                "items":{
                    "$ref":"#/definitions/HistoryTransactionRow"
                }
            }
        }
    },
    "getDownloadUploadInfoResponse":{
        "properties":{
            "data":{
                "type":"array",
                "items":{
                    "$ref":"#/definitions/DownloadUploadInfo"
                }
            },
            "download":{
                "type":"object",
                "$ref":"#/definitions/DownloadInfo"
            },
            "upload":{
                "type":"object",
                "$ref":"#/definitions/UploadInfo"
            }
        }
    }
}

```

```

    }
  }
},
"getQueryListResponse":{
  "properties":{
    "result":{
      "type":"array",
      "items":{
        "$ref":"#/definitions/Query"
      }
    }
  }
},
"getInboxTransactionSummaryResponse":{
  "properties":{
    "totalRows":{
      "type":"integer"
    },
    "rows":{
      "type":"array",
      "items":{
        "$ref":"#/definitions/InboxTransactionSummary"
      }
    }
  }
},
"getInboxTransactionListResponse":{
  "properties":{
    "conditions":{
      "type":"array",
      "items":{
        "$ref":"#/definitions/InboxTransactionCondition"
      }
    },
    "totalRows":{
      "type":"integer"
    },
    "rows":{
      "type":"array",
      "items":{

```

```

        "$ref": "#/definitions/InboxTransactionRow"
    }
}
},
"uploadFileResponse": {
    "properties": {
        "result": {
            "type": "string"
        }
    }
},
"createQueryResponse": {
    "properties": {
        "oid": {
            "type": "string"
        }
    }
},
"changePasswordResponse": {
    "properties": {
        "result": {
            "type": "string"
        }
    }
},
"updateQueryResponse": {
    "properties": {
        "result": {
            "type": "string"
        }
    }
},
"deleteQueryResponse": {
    "properties": {
        "result": {
            "type": "string"
        }
    }
},
},

```



```

    "QueryCondition":{
      "properties":{
        "name":{
          "type":"string",
          "enum":[
            "PROTOCOL_NAME",
            "TPNAME",
            "USER_TRANS_ID",
            "GSUSER",
            "STREAM_SIZE",
            "OPERATION_ID",
            "STATUS",
            "TS"
          ]
        },
        "operator":{
          "type":"string",
          "enum":[
            "EQUALS",
            "NOT_EQUALS",
            "LIKE",
            "NOT_LIKE"
          ]
        },
        "value":{
          "type":"string"
        }
      }
    },
    "HistoryTransactionSort":{
      "properties":{
        "name":{
          "type":"string",
          "enum":[
            "PROTOCOL_NAME",
            "USER_TRANS_ID",
            "BC_TRANS_ID",
            "OPERATION_ID",
            "STATUS",
            "TS",

```

```

        "TPNAME"
    ]
},
"inc":{
    "type":"boolean"
}
}
},
"HistoryTransactionCondition":{
    "properties":{
        "name":{
            "type":"string",
            "enum":[
                "PROTOCOL_NAME",
                "USER_TRANS_ID",
                "BC_TRANS_ID",
                "OPERATION_ID",
                "STATUS",
                "TS",
                "TPNAME"
            ]
        },
        "operator":{
            "type":"string",
            "enum":[
                "EQUALS",
                "NOT_EQUALS",
                "LIKE",
                "NOT_LIKE"
            ]
        },
        "value":{
            "type":"string"
        }
    }
},
"DownloadUploadInfo":{
    "properties":{
        "id":{
            "type":"string"
        }
    }
}

```

```

    },
    "type":{
        "type":"string"
    },
    "name":{
        "type":"string"
    },
    "total":{
        "type":"number"
    },
    "current":{
        "type":"number"
    },
    "user_transaction_id":{
        "type":"string"
    },
    "protocol":{
        "type":"string"
    },
    "operation":{
        "type":"string"
    },
    "bc_transaction_id":{
        "type":"string"
    },
    "startTime":{
        "type":"string"
    }
}
},
"InboxTransactionSort":{
    "properties":{
        "name":{
            "type":"string",
            "enum":[
                "PROTOCOL_NAME",
                "USER_TRANS_ID",
                "OPERATION_ID",
                "STATUS",
                "TS",

```

```

        "TPNAME"
    ]
},
"inc":{
    "type":"boolean"
}
}
},
"ProtocolOperation":{
    "properties":{
        "protocol":{
            "type":"string"
        },
        "operations":{
            "type":"array",
            "items":{
                "$ref":"#/definitions/Operation"
            }
        }
    }
},
},
"InboxTransactionCondition":{
    "properties":{
        "name":{
            "type":"string",
            "enum":[
                "PROTOCOL_NAME",
                "USER_TRANS_ID",
                "OPERATION_ID",
                "STATUS",
                "TS",
                "TPNAME"
            ]
        },
        "operator":{
            "type":"string",
            "enum":[
                "EQUALS",
                "NOT_EQUALS",
                "LIKE",

```

```

        "NOT_LIKE"
    ]
},
"value":{
    "type":"string"
}
}
},
"Query":{
    "properties":{
        "conditions":{
            "type":"array",
            "items":{
                "$ref":"#/definitions/QueryCondition"
            }
        },
        "name":{
            "type":"string"
        },
        "type":{
            "type":"string",
            "enum":[
                "history",
                "inbox"
            ]
        },
        "editable":{
            "type":"boolean"
        },
        "oid":{
            "type":"string"
        },
        "description":{
            "type":"string"
        },
        "partner":{
            "type":"string"
        },
        "user":{
            "type":"string"
        }
    }
}

```

```

    }
  },
  "UploadInfo":{
    "properties":{
      "finish":{
        "type":"number"
      },
      "active":{
        "type":"number"
      }
    }
  },
  "User":{
    "properties":{
      "host":{
        "type":"string"
      },
      "readOnly":{
        "type":"boolean"
      },
      "name":{
        "type":"string"
      },
      "partner":{
        "type":"string"
      }
    }
  },
  "Operation":{
    "properties":{
      "operation":{
        "type":"string"
      },
      "type":{
        "type":"string",
        "enum":[
          "notify",
          "asyncRequestResponse"
        ]
      }
    }
  }
}

```

```

    }
  }
},
"HistoryTransactionSummary":{
  "properties":{
    "Operation":{
      "type":"string"
    },
    "Status":{
      "type":"string"
    },
    "Date":{
      "type":"string"
    },
    "ID":{
      "type":"string"
    },
    "Protocol":{
      "type":"string"
    }
  }
},
"DownloadInfo":{
  "properties":{
    "finish":{
      "type":"number"
    },
    "active":{
      "type":"number"
    }
  }
},
"HistoryTransactionRow":{
  "properties":{
    "PROTOCOL_NAME":{
      "type":"string"
    },
    "TPNAME":{
      "type":"string"
    },
  },

```

```

        "USER_TRANS_ID":{
            "type":"string"
        },
        "BC_TRANS_ID":{
            "type":"string"
        },
        "TS":{
            "type":"string"
        },
        "STATUS":{
            "type":"string"
        },
        "OPERATION_ID":{
            "type":"string"
        }
    }
},
"InboxTransactionSummary":{
    "properties":{
        "Host":{
            "type":"string"
        },
        "Status":{
            "type":"string"
        },
        "Operation":{
            "type":"string"
        },
        "Date":{
            "type":"string"
        },
        "ID":{
            "type":"string"
        },
        "Protocol":{
            "type":"string"
        },
        "Size":{
            "type":"integer"
        }
    }
}

```



```

    }
  },
  "InboxTransactionRow":{
    "properties":{
      "PROTOCOL_NAME":{
        "type":"string"
      },
      "PROCESSING":{
        "type":"boolean"
      },
      "USER_TRANS_ID":{
        "type":"string"
      },
      "BC_TRANS_ID":{
        "type":"string"
      },
      "STATUS":{
        "type":"string"
      },
      "SIZE":{
        "type":"integer"
      },
      "TS":{
        "type":"string"
      },
      "ACTION":{
        "type":"string"
      },
      "OPERATION_ID":{
        "type":"string"
      },
      "ATTACHMENT":{
        "type":"boolean"
      },
      "USER":{
        "type":"string"
      }
    }
  }
},
"Error":{

```

```
    "properties":{
      "error":{
        "type":"string"
      }
    }
  }
}
```


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 TIBCO Software Inc.

TIBCO, the TIBCO logo, the TIBCO O logo, TIBCO PartnerExpress, TIBCO Hawk, TIBCO Rendezvous, TIBCO Runtime Agent, TIBCO ActiveMatrix BusinessWorks, TIBCO Administrator, TIBCO Designer, TIBCO Enterprise Message Service, TIBCO BusinessConnect, TIBCO BusinessConnect Palette, TIBCO BusinessConnect Services Plug-in, and TIBCO BusinessConnect EDI Protocol powered by Instream are either registered trademarks or trademarks of TIBCO Software Inc. in the United States and/or other countries.

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

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

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

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

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

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

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

Copyright © 1999-2020. TIBCO Software Inc. All Rights Reserved.

